

UNIVERSIDADE DO ALGARVE

INTERFACES PESSOA-MÁQUINA EM AMBIENTES DE
PROGRAMAÇÃO LOGO

Stéphane Norte

Doutoramento em Engenharia Electrónica e Computação

Ciências da Computação

2010

UNIVERSIDADE DO ALGARVE

INTERFACES PESSOA-MÁQUINA EM AMBIENTES DE
PROGRAMAÇÃO LOGO

Stéphane Norte

Tese orientada por Professor Fernando Miguel Pais da Graça Lobo

Doutoramento em Engenharia Electrónica e Computação

Ciências da Computação

2010

“For me, my disability is a fact and not a problem. I’m not living the life of a disabled person. For sure, I have to handle some things differently from other people. But it’s not so different from the life of someone who is not disabled. In any case, who is really not disabled?”

Thomas Quasthoff, Cantor de Ópera

Resumo

Na nossa sociedade, inúmeras pessoas com deficiências físicas são muitas vezes excluídas do seu ambiente e impedidas de usufruir de condições de vida normais. Nos ambientes educativos, estes problemas acentuam-se e os alunos com limitações físicas começam a sentir frustração por não conseguirem realizar todas as tarefas impostas no meio escolar. Os ambientes virtuais surgem como uma ferramenta essencial para colmatar estas deficiências, possibilitando assim uma vida académica promissora. Com a evolução das novas tecnologias apresenta-se uma nova era da informática especializada em fornecer formas simples e eficientes de comunicação com os utilizadores.

Esta dissertação tem como objectivo principal auxiliar estudantes com limitações físicas a desenvolverem programas em ambientes de programação *Logo*. Para tal, foi desenvolvido um teclado virtual com características especiais denominado por *LogoKeyboard*. Este teclado apresenta a particularidade de ser composto por comandos de programação em *Logo*, havendo ainda a possibilidade de se poder utilizar um teclado alfanumérico. Uma pessoa que apresente características físicas suficientes para pressionar um simples botão, pode manipular e configurar a interface e assim programar na linguagem *Logo*. Esta tecnologia assistiva foi também implementada em ambientes lúdicos de aprendizagem, nomeadamente no jogo *Sudoku*.

Palavras-Chave: Deficiência física, Programação Logo, Teclado virtual, Construcionismo, Sudoku.

Abstract

In our society, many people with disabilities are often excluded from their environment and, consequently, prevented from enjoying normal living conditions. In educational environments there is an evident increment of these problems and the students with disabilities begin to feel frustrated for not being able to perform all the required school tasks. Virtual environments appear as an essential tool to address these disabilities, making possible a very promising academic life.

The blossoming of new technologies is promoting a new computing era that is specialized in providing simple and efficient manners of communication with the users. This dissertation's main goal is to help students with disabilities in the development of programs in *Logo* programming environments. In order to accomplish our goal we developed a virtual keyboard presenting special features. This keyboard was named *LogoKeyboard* and has the particularity of being composed by *Logo* programming commands and the possibility of being able to use an alphanumeric keyboard. A person with enough physical ability to press a single button can manipulate and configure the interface and programme with the *Logo* language. This technology was also implemented in an educational game called *Sudoku Access*.

Keywords: Physical impairment, Logo programming, Virtual keyboard, Constructionism, Sudoku.

Agradecimentos

Gostaria de agradecer a um determinado número de pessoas pelo apoio e suporte oferecido durante o longo e sinuoso, mas gratificante, caminho do desenvolvimento da minha dissertação.

Ao meu orientador Professor Fernando Lobo agradeço por me ter aceitado como seu aluno, pelo apoio e interesse demonstrado ao longo destes anos, pelo conhecimento partilhado e por todas as discussões científicas que ajudaram a semear as sementes do meu conhecimento. Durante a minha investigação tive a possibilidade de trocar ideias com pessoas muito interessantes e de apresentar o meu trabalho em conferências internacionais. Tudo isto só foi possível devido ao seu apoio incondicional.

Ao meu colega Dr. Paulo Condado expresso o meu sincero agradecimento pelas críticas e sugestões oferecidas, assim como pela sua amizade, boa disposição e pelos bons momentos que vivemos. Gostaria de agradecer ao meu colega António dos Anjos pelas suas ideias e pelo seu companheirismo. Aos meus colegas de trabalho da Escola Secundária de Loulé: Paulo Cavaco, Filipe de Jesus, Elisabete Faria Dias, Edite Machado, Dinis Dias Pereira, Carla Lourenço, Virgílio Ferreira e Fernando Castanheira e da Escola Secundária Dr.^a Laura Ayres: Paulo Viegas, Hélder Coelho, Maria Martins, Carla Dias e Victor Veiga, agradeço toda a amizade.

Um muito obrigado aos responsáveis e técnicos da Associação Portuguesa de Paralisia Cerebral (APPC), cujas indicações me proporcionaram uma precisão maior no desenvolvimento do meu trabalho. Aos estudantes que me ajudaram durante os testes com *Logo Keyboard* e *Sudoku Access*, nomeadamente: Ana Martins, Daniel Dixon, Nilton Delgado, Maria de Fátima Martins, Jorge Dionísio e Daniela Vieira dos Santos, assim como ao encarregados de educação que permitiram que as interfaces fossem testadas, nomeadamente, Cristiane Vieira e João Santos.

Agradeço à Professora Elsa Guerreiro pela sua ajuda e disponibilidade durante a realização dos testes com os alunos.

Um agradecimento especial à Professora Helena Pereira pela sua ajuda, apoio e amizade ao longo da minha dissertação.

Quero agradecer aos meus colegas de trabalho e principalmente amigos da direcção da Escola Secundária dr.^a Laura Ayres: A professora Maria do Nascimento pela sua simpatia e amizade, a professora Manuela Encarnação pela sua simpatia, carinho e boa disposição, a professora Dalila pela sua simpatia e amizade e um especial agradecimento a professora e directora Conceição Bernardes por toda a sua amizade, carinho, apoio e confiança que depositou em mim.

Ao meu amigo Nuno Castilho um sincero obrigado pela sua amizade, apoio e sugestões apresentadas ao longo destes anos.

Aos meus avós expresso um sincero obrigado pelo carinho e preocupação revelada ao longo de todos esses anos de trabalho académico e à minha querida e amorosa bisavó pela ajuda disponibilizada desde sempre.

Um obrigado muito sentido para os meus pais, Idalino e Nathalie pelo apoio, carinho e paciência demonstrado durante estes anos e por todas as conversas motivadoras e reconfortantes.

À minha mulher Carina, agradeço toda a sua ajuda, compreensão, paciência, apoio, carinho e por me ter influenciado positivamente, transmitindo-me confiança para acreditar que este projecto era possível de ser concretizado.

Finalmente quero agradecer à Fundação para a Ciência e a Tecnologia pelo financiamento de parte dos meus trabalhos de doutoramento, em especial as missões a centros de investigação internacionais e a participação em conferências de relevo, no âmbito do projecto POCI/CED/62497/2004.

Índice

1	Introdução	18
1.1	Objectivos	20
1.2	Contribuições	21
1.3	Organização	23
2	Problemas de Interação Pessoa-Máquina em Portadores de Deficiência Física	25
2.1	Introdução	25
2.2	A Deficiência Física	25
2.3	A Deficiência Física em Ambientes de Aprendizagem	30
2.3.1	A Tecnologia Assistiva	34
2.4	A Deficiência Física e o Computador como Ferramenta de Aprendizagem	38
2.5	Sumário	39
3	Linguagem de Programação <i>Logo</i>: Uma Filosofia de Aprendizagem	41
3.1	Introdução	41
3.1.1	O Papel do Construcionismo na Integração das Tecnologias na Educação	45
3.2	A Linguagem de Programação <i>Logo</i>	46
3.2.1	História da Linguagem <i>Logo</i>	47
3.2.2	Características da Linguagem <i>Logo</i>	50
3.2.3	O Ambiente da Programação <i>Logo</i>	52

3.3	Sumário	55
4	Evolução da Tecnologia Construcionista em Ambientes de Aprendizagem	56
4.1	Introdução	56
4.2	A Robótica Educativa	57
4.2.1	A Tartaruga <i>Logo</i> - o início da robótica educativa	57
4.2.2	O Sistema <i>LEGO/Logo</i>	58
4.2.3	Programmable Bricks	59
4.2.4	Logo Brick	60
4.2.5	Red Brick	61
4.2.6	LEGO Mindstorms	62
4.2.7	Cricket	64
4.3	Robótica Educacional de Baixo Custo: <i>GoGo Board</i>	65
4.3.1	Projectos Realizados com a <i>GoGo Board</i>	68
4.4	Sumário	73
5	Aprender <i>LOGO</i> no Ensino Secundário: Construcionismo versus Objectivismo	74
5.1	Introdução	74
5.2	Descrição do Estudo	75
5.3	População em Estudo	76
5.4	Ambiente Tradicional	77
5.5	Ambiente Construcionista	79
5.6	Feedback dos Alunos	83
5.7	Análise e Resultados	84
5.8	A Linguagem <i>Logo</i> para Crianças com Paralisia Cerebral	85
5.9	Sumário	89
6	LogoKeyboard	91
6.1	Introdução	91

6.2	Metodologias de Desenho da Interface	93
6.3	Organização do Teclado	95
6.4	Ferramentas da Interface	96
6.5	Métodos de <i>Input</i>	99
6.6	Motivação e Aprendizagem	102
6.7	Testes de Usabilidade	104
6.7.1	Introdução	104
6.7.2	Objectivos	105
6.7.3	Seleccção do Grupo de Utilizadores	105
6.7.4	Preparação dos Testes	108
6.7.5	Realização dos Testes	108
6.7.5.1	Caso de Estudo do Participante Daniel	112
6.7.5.2	Caso de Estudo do Participante Nilton	112
6.7.5.3	Caso de Estudo da Participante Ana	114
6.7.5.4	Caso de Estudo da Participante Beatriz	115
6.7.6	Análise e Resultados	117
6.8	Discussão	120
6.9	Sumário	122
7	Ambientes Lúdicos de Aprendizagem Computacional	124
7.1	Introdução	124
7.2	Utilidade dos Jogos na Aprendizagem	125
7.3	<i>Sudoku Access</i> : O Jogo do <i>Sudoku</i> para Pessoas com Deficiências Físicas	126
7.3.1	O Jogo do <i>Sudoku</i>	126
7.3.2	Metodologias de Desenho do Jogo	127
7.4	Testes de Usabilidade	133
7.4.1	Introdução	133
7.4.2	Objectivos	134
7.4.3	Seleccção do Grupo de Utilizadores	135

7.4.4	Preparação dos Testes	136
7.4.5	Realização dos Testes	136
7.4.5.1	Caso de Estudo da Participante Daniela	138
7.4.5.2	Caso de Estudo da Participante Fátima	140
7.4.5.3	Caso de Estudo do Participante Jorge	143
7.4.6	Análise e Resultados	145
7.5	Discussão	146
7.6	Sumário	148
8	Sumário, Trabalho Futuro e Conclusões	149
8.1	Sumário	149
8.2	Trabalho Futuro	151
8.3	Conclusões	152

Lista de Tabelas

6.1	Resumo dos dispositivos utilizados pelos alunos, número de teclas pressionadas e tempo despendido na construção do programa “ <i>Square</i> ”.	118
6.2	Teclado convencional utilizado pelos alunos para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: <i>Fd</i> , <i>Setsize</i> , <i>Setpenseize</i> e <i>OpenWorkSheet</i> (A Ana e a Beatriz não conseguiram interagir com o teclado).	118
6.3	Teclado virtual da <i>Microsoft</i> utilizado pelos alunos com o sistema de varrimento para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: <i>Fd</i> , <i>Setsize</i> , <i>Setpenseize</i> e <i>OpenWorkSheet</i> .	119
6.4	<i>LogoKeyboard</i> utilizado pelos alunos com o sistema de varrimento para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: <i>Fd</i> , <i>Setsize</i> , <i>Setpenseize</i> e <i>OpenWorkSheet</i> .	119
6.5	Resultado do número de teclas pressionadas por cada aluno utilizando um rato para manusear o teclado virtual da <i>Microsoft</i> e o <i>LogoKeyboard</i> .	119
6.6	Resultados do questionário relativamente a velocidade, facilidade de utilização e precisão quando utilizado o <i>LogoKeyboard</i> e um teclado físico. Os resultados são apresentados numa escala de 1 (Fraco) até 5 (Bom). (F- Físico, V- Virtual)	120
7.1	Resumo dos dispositivos utilizados pelos alunos, número de células preenchidas, número de teclas pressionadas, número de palavras pronunciadas e o tempo despendido para completar uma grelha 9x9 do <i>Sudoku</i>	146

Lista de Figuras

2.1	(a) São várias as dificuldades que os indivíduos com problemas motores podem enfrentar no mundo real, nomeadamente na realização de tarefas diárias e na integração na sociedade. (b) A tecnologia computacional surge como uma ferramenta essencial para abrir novas oportunidades para estas pessoas.	32
3.1	O procedimento “ <i>Square</i> ” permite a construção da figura geométrica quadrado.	54
4.1	A tartaruga robotizada controlada por crianças utilizando a linguagem <i>Logo</i> (Papert, 1993b)	57
4.2	Programmable Brick.	59
4.3	Logo Brick.	61
4.4	Red Brick.	62
4.5	Lego Mindstorms RCX.	63
4.6	Cricket.	65
4.7	Características principais da <i>GoGo Board</i>	67
4.8	Interface do jogo “Corrida de Cavalos” desenvolvido com o <i>software Microworlds</i> e a <i>GoGo Board</i>	69
4.9	Protótipo da “Casa de banho inteligente” desenvolvida no Brasil.	70
4.10	Escala de Calorias.	70
4.11	<i>Software</i> “Auxílio a Deficientes Motores” (ADM).	72
4.12	Maqueta do quarto de um possível utilizador	72
5.1	Primeira figura desenvolvida em <i>Logo</i> pelos alunos.	78

5.2	Casa implementada em <i>Logo</i> pelos estudantes.	80
5.3	Animações implementados pelos alunos com cavalos, bicicletas a motor, e plantas.	81
5.4	O procedimento do movimento do cavalo.	81
5.5	O jogo do labirinto implementado pelos alunos.	82
5.6	Procedimentos desenvolvidos pelos estudantes no jogo do labirinto. . .	83
5.7	Primeiro contacto no laboratório <i>Mindstorms</i> com as crianças da APPC. .	86
5.8	Primeira versão do teclado virtual para auxiliar os jovens na digitação dos comandos em <i>Logo</i>	87
5.9	Segunda versão do teclado virtual constituída não por palavras mas por imagens.	87
5.10	Terceira versão do teclado virtual constituída por 4 teclas de maiores dimensões.	88
5.11	Quarta versão do teclado virtual constituída por uma imagem central da tartaruga e rodeada por 4 setas direccionais.	89
6.1	a) O primeiro grupo de teclados é composto por vocabulários relacionados com o ambiente gráfico, b) O segundo com a gestão da tela e edição de texto, c) O terceiro com o manuseamento de listas/palavras e acesso ao disco.	96
6.2	d) O quarto grupo de teclados é composto por vocabulários relacionados com a utilização de objectos, controle e lógica, e) O quinto grupo com operações matemáticas e funções, f) O último grupo é constituído por comandos de entrada e saída de dados, tempos, sons e variáveis. . .	97
6.3	Constituição do teclado alfanumérico do <i>LogoKeyboard</i>	98
6.4	O utilizador visualiza o sistema de ajuda do <i>LogoKeyboard</i> apenas colocando o cursor do rato na tecla desejada. Neste exemplo o comando “Fd” foi seleccionado e à direita é gerado um ambiente para visualizar uma pequena animação referente ao comando desejado. . .	98

-
- 6.5 O sistema para aumentar o tamanho das teclas no teclado virtual pode ser manipulado de duas formas: a) colocando o cursor do rato na tecla desejada ou b) automaticamente através do sistema de varrimento. 99
- 6.6 a) Selecção do teclado de programação composto pelos comandos de programação em *Logo*; b) Selecção do teclado alfanumérico. 101
- 6.7 a) O utilizador ao pressionar o *switch* quando a área do teclado de interesse se encontra evidenciada com uma determinada cor no teclado desejado permite desta forma efectuar escolhas no *LogoKeyboard*. b) Após ter pressionado o botão de toque outro sistema de varrimento selecciona automaticamente cada grupo de teclas na horizontal. Neste exemplo o utilizador pressionou o botão de toque na linha onde se encontra o comando “SetPenSize”. c) Finalmente é activado outro sistema de selecção que focaliza com uma cor o interior de cada comando. Quando o utilizador visualiza a cor no comando “SetPenSize”, pressiona o botão de toque. d) Desta forma o “SetPenSize” é enviado para o editor de texto do interpretador de *Logo* executando a sua tarefa. 103
- 6.8 Projecto “Jogo do Labirinto” implementado pelo Nilton utilizando o teclado virtual *LogoKeyboard* 113
- 6.9 “Jogo do Labirinto” criado pela Ana utilizando o *LogoKeyboard*. 115
- 6.10 “Jogo do Labirinto” desenvolvido pela Beatriz utilizando o *LogoKeyboard*. 116
- 7.1 Exemplo de um puzzle *Sudoku* e a respectiva solução. O objectivo é preencher os espaços vazios com dígitos de 1 até 9 de modo que cada dígito ocorre apenas uma vez em cada linha, coluna e quadrado 3x3 assinalados com linhas a negrito. 127
- 7.2 *Sudoku Access*: o jogo do *Sudoku* para pessoas com limitações físicas. 129

- 7.3 Exemplo do funcionamento do sistema de selecção por varrimento no jogo *Sudoku Access*, onde os itens (ou grupos de itens) vão sendo seleccionados automática e sequencialmente, após um determinado tempo predefinido. Existem dois tipos principais de varrimento: a) varrimento para seleccionar a grelha do *Sudoku*, permitindo desta forma ao utilizador introduzir números; b) varrimento para escolher as opções de configuração do puzzle. 129
- 7.4 Sistema de selecção do jogo *Sudoku Access*. Este sistema de selecção permite ao utilizador preencher as células do puzzle com os números, utilizando um simples botão de toque ou pressionando a tecla “Espaço”. a) Selecção do primeiro grupo, b) selecção do primeiro subgrupo, c) selecção da primeira linha do primeiro subgrupo, d) selecção do primeiro item da primeira linha do primeiro subgrupo, e) um outro sistema de selecção é activado no teclado numérico (visualizado no lado direito do puzzle), sendo que nesta fase o próximo “click” corresponde à escolha do número desejado, f) o número desejado é enviado à célula. 131
- 7.5 O sistema de reconhecimento de voz como processo de selecção no jogo *Sudoku Access*. Utilizando a voz, o utilizador escolhe números entre 1 e 5 para escolher a tarefa desejada. Neste exemplo o utilizador escolheu o número “4” que corresponde à activação do sistema de treino do reconhecimento de voz da Microsoft. 132
- 7.6 Na primeira figura o utilizador escolhe a linha desejada (1-15) e a coluna (1-9) utilizando a própria voz. Na segunda figura, é exemplificado a escolha do número pretendido para preencher uma célula no *Sudoku Access* (linha 5, coluna 4). 133
- 7.7 A Fátima, uma rapariga de treze anos com *Spina Bífida*, está a manusear o seu computador utilizando a tecla “Espaço” para controlar o jogo do *Sudoku Access*. 142

-
- 7.8 O *Sudoku Access* pode ser configurado para jogar: a) apenas com números, b) cores e números, c) apenas com cores. 142
- 7.9 Resultados do questionário de satisfação realizado aos alunos relativamente à facilidade de utilização, facilidade de aprendizagem, velocidade, agradabilidade, conforto e contentamento com a interface *Sudoku Access*. 147

1

Introdução

A interacção entre pessoas e computadores é uma matéria multidisciplinar que relaciona ciência da computação, arte, design, ergonomia, psicologia, sociologia, semiótica, linguística e áreas afins. A interacção entre Homens e máquinas ocorre através da interface do utilizador, a qual é formada por *software* e *hardware*. Esta é utilizada para o desempenho de muitas acções, desde a manipulação de periféricos de computadores até ao controlo de grandes máquinas, como sendo os aviões. A performance humana no uso de computadores e de sistemas de informação é uma área de investigação e desenvolvimento que muito se expandiu nas últimas décadas. A microinformática tem-se desenvolvido com grande rapidez, transformando computadores de grande porte em pequenas estações de trabalho, cada vez mais potentes, e em *notebooks* portáteis. Os aplicativos têm evoluído da mesma forma.

Com o intuito de aprimorar a interacção Pessoa-Máquina, programadores e projectistas têm desenvolvido interfaces cada vez mais amigáveis, que substituem complicadas linhas de comando por um simples clicar de um ícone. Hoje em dia é complicado viver sem o auxílio de computadores. Seja na escola, no trabalho, em bancos, lojas e supermercados, a informática está cada vez mais empenhada no desenvolvimento de sistemas que facilitem a vida de inúmeras pessoas. Neste novo quadro que se apresenta para a era da informática, é essencial que os sistemas ou

as interfaces forneçam uma forma simples e fácil de comunicação com o utilizador (Norman, 2002).

O desenvolvimento de estudos no campo das acessibilidades utilizando computadores, proporciona enormes vantagens para suprimir limitações específicas de um determinado grupo de utilizadores, tornando mais simples e eficaz o acesso aos meios tecnológicos. Actualmente, nos estabelecimentos de ensino muitas são as dificuldades sentidas pelos professores devido à falta de tecnologia assistiva nas salas de aula. Esta realidade acompanha-me como professor numa escola secundária e vivo diariamente as lacunas que os jovens manifestam. Naturalmente, os alunos com deficiências físicas necessitam de uma interface de *hardware* e *software* adequada e que disponha de dispositivos de *input* tal como *switches*, sistemas de *face and eye tracking*, *joysticks*, *sip-and-puff*, sistemas de reconhecimento de voz, teclados virtuais, entre outros, que substituam os periféricos tradicionais. Assim, durante o desenvolvimento e a concepção de interfaces Pessoa-Máquina, é fundamental adotar um conjunto de testes de usabilidade com o objectivo de desenhar uma interface adequada às necessidades dos seus utilizadores.

Este estudo está essencialmente direccionado para a análise e o desenvolvimento de ferramentas educativas que auxiliem indivíduos com limitações ao nível da coordenação motora. A ênfase deste trabalho concentra-se no desenvolvimento de um sistema informático que possibilite que pessoas com limitações físicas possam ganhar uma maior independência na implementação dos seus próprios programas de computador. Este sistema, designado *LogoKeyboard*, combina tecnologias assistivas para auxiliar indivíduos com deficiências físicas a programarem numa linguagem de programação focalizada principalmente para crianças, mas que também é benéfica para jovens ou até mesmo adultos.

No decorrer deste projecto, diversos estudos foram realizados em escolas básicas e secundárias com crianças e jovens portadores de diversas deficiências físicas e cognitivas: limitações físicas ao nível dos membros superiores e inferiores, paralisia cerebral, doenças degenerativas caracterizadas por uma paralisia progressiva dos

músculos e deficiência congénita do tubo neural denominada por *Spina Bífida*. Todos estes jovens participaram na criação da interface *LogoKeyboard*, auxiliando no seu processo de desenvolvimento iterativo que permitiu adaptá-lo às necessidades dos utilizadores. Este processo de desenvolvimento iterativo, apoiado por um conjunto de testes de usabilidade realizados ao longo da construção da interface, permite que os principais problemas de usabilidade fossem detectados e resolvidos durante o período de concepção do produto.

1.1 Objectivos

Com a presente dissertação pretendo contribuir para o desenvolvimento de tecnologias assistivas e, conseqüentemente, para a melhoria da qualidade de vida de pessoas com deficiências físicas. É do nosso conhecimento que muitas tecnologias de acessibilidade já foram estudadas, analisadas e desenvolvidas no processo educativo de crianças com necessidades especiais, designadamente estudantes com problemas motores. No entanto, poucas são as tecnologias assistivas relacionadas com a programação de computadores que permitem a estas pessoas programarem os seus próprios programas. Deste modo, é fulcral a realização de uma análise directa que permita compreender que tipos de dificuldades são sentidas no manuseamento de dispositivos tecnológicos. É importante perceber que tipo de *software* pode ajudar indivíduos que apresentam limitações motoras. Sendo investigador e professor, todo o meu conhecimento científico e toda a minha experiência no convívio com jovens com deficiências físicas foram canalizados para o desenvolvimento de uma interface que possibilite às crianças com limitações motoras implementarem os seus próprios programas informáticos, utilizando para tal uma determinada linguagem de programação. O desfecho desta extensa e exaustiva mas gratificante investigação é a presente dissertação, cujos principais objectivos são:

- Estudar os tipos de dificuldades que os indivíduos com deficiências motoras revelam na interacção com o computador, de forma a compreender que tipo

de interfaces podem ser construídas para ajudar estas pessoas;

- Desenvolver tecnologias assistivas de baixo custo que permitam auxiliar pessoas com limitações motoras a adquirirem maior autonomia quando interagem com um computador;
- Implementar métodos que permitam que uma pessoa com deficiência motora possa desenvolver os seus próprios programas informáticos utilizando para este fim interfaces adequadas;
- Disponibilizar nas escolas um *software* gratuito para auxiliar alunos com deficiências motoras a desenvolverem autonomamente os seus próprios programas informáticos sem auxílio de um assistente.

1.2 Contribuições

Esta dissertação foca as interfaces Pessoa-Máquina para indivíduos com limitações motoras. O leitor poderá perguntar: *qual a importância destas interfaces em ambientes de aprendizagem?* A resposta é simples: interfaces que apresentem uma construção adequada podem ser o único elo de ligação entre o mundo real e o mundo virtual no meio educacional, permitindo que pessoas com determinadas limitações físicas possam usufruir dos mesmos direitos de aprendizagem que qualquer outro ser humano na nossa sociedade. As principais contribuições desta dissertação são:

- O estudo das diversas deficiências motoras permitiu compreender quais as maiores necessidades que um indivíduo com limitações físicas específicas apresenta quando interage com um computador. Com esta análise foi possível desenvolver um sistema que auxilia as pessoas com deficiências motoras a interagirem com o computador e, deste modo, implementarem os seus próprios programas informáticos. Este sistema, denominado *LogoKeyboard*, apresenta a particularidade de poder auxiliar, por exemplo, indivíduos que apresentam incapacidade de manusear um teclado convencional a utilizarem a linguagem de

programação *Logo*. Com o *LogoKeyboard* torna-se possível desenvolver programas com a linguagem *Logo* sem ser necessário digitar caracteres num teclado convencional (Norte & Lobo, 2007);

- O *LogoKeyboard* despertou nos alunos um grande interesse relativo à aprendizagem da linguagem de programação *Logo*. Este interesse foi crescendo ao longo das aulas assim como a capacidade criativa dos estudantes. A própria filosofia construcionista proporcionou aos alunos novos ambientes de aprendizagem em sala de aula que contribuíram para a exploração de diferentes projectos em *Logo* (Norte *et al.*, 2007);
- A linguagem de programação *Logo* combinada com a filosofia construcionista permitiu demonstrar que os alunos que aprendem num ambiente construcionista revelam maior interesse do que aqueles que aprendem num ambiente tradicional. Por esta razão reforçamos o despertar do interesse dos alunos do ensino básico e secundário com a utilização de novas técnicas educativas menos tradicionais (Norte *et al.*, 2007);
- A criação de um sistema com carácter lúdico, designado por *Sudoku Access*, vem auxiliar estudantes com deficiências físicas a permanecerem relaxados e a desenvolverem competências, uma vez que este tipo de puzzle ajuda a melhorar a concentração e o pensamento lógico. O *Sudoku* auxilia as crianças a raciocinar para a resolução de problemas e estas apenas necessitam de ter algumas informações em mente para definir as estratégias de jogo. O *Sudoku Access* apresenta ferramentas que permitem que indivíduos com deficiências motoras possam interagir com o jogo sem ser necessária a utilização de um teclado convencional ou de um rato (Norte & Lobo, 2008a,b).

O trabalho realizado no âmbito desta dissertação, originou seis artigos científicos, sendo que dois deles foram publicados na *ACM ITICSE* (Norte & Lobo, 2007, 2008b), a conferência de referência da *Association for Computer Machinery (ACM)* para a área de *Innovation and Technology in Computer Science Education*, e um

outro foi publicado na *ACM ASSETS* em 2008 (Norte & Lobo, 2008a), a conferência de referência da *ACM* para a área de *Computers and Accessibility*.

1.3 Organização

Esta dissertação encontra-se organizada do seguinte modo:

- O capítulo 1 evidencia os objectivos da dissertação, fundamentando a importância das interfaces Pessoa-Máquina em ambientes de aprendizagem. É descrita a organização dos capítulos da dissertação e a contribuição deste trabalho para a sociedade e para o estado da arte;
- No capítulo 2 estão enumerados os problemas motores mais frequentes que podem afectar a interacção Pessoa-Máquina em ambiente de aprendizagem, sendo realizada uma descrição pormenorizada dessas limitações físicas, bem como as estratégias e tecnologias que ajudam os estudantes a reduzir as suas restrições físicas;
- Uma revisão da linguagem de programação *Logo* é relatada no capítulo 3, assim como a integração das tecnologias na educação, revelando a importância dessa linguagem de programação no meio educativo e descrevendo as conjecturas de Seymour Papert. São ainda evidenciadas as principais características deste tipo de linguagem;
- No capítulo 4 é apresentada a evolução da tecnologia construcionista em ambientes de aprendizagem, relatando o progresso da robótica educativa de baixo custo e enfatizando a plataforma de *hardware* denominada de *GoGoBoard*. São ainda descritos alguns projectos nos quais a *GoGoBoard* foi implementada;
- O capítulo 5 descreve um estudo realizado numa escola secundária, utilizando duas abordagens pedagógicas distintas, onde são reveladas as potencialidades da programação em *Logo* no meio educativo. É narrado o *feedback* dos alunos,

assim como os resultados obtidos no decorrer deste estudo. É ainda demonstrado um estudo realizado com a Associação Portuguesa de Paralisia Cerebral utilizando a linguagem de programação *Logo*;

- No capítulo 6 surge o teclado virtual que proporciona uma técnica fácil e útil para auxiliar pessoas na programação com a linguagem *Logo*. É realizada uma descrição pormenorizada das técnicas usadas e das características desta nova ferramenta educativa. Neste capítulo também é apresentada uma revisão das técnicas de entrada de dados em aplicações informáticas, para auxiliar indivíduos com limitações físicas. Também neste capítulo encontra-se descrito um estudo com o *software LogoKeyboard* realizado com estudantes de escolas Básica e Secundária, que apresentam diferentes necessidades instrucionais;
- O capítulo 7 expõe a importância que os jogos educativos podem ter para os estudantes. É descrita a implementação de um novo jogo educativo utilizado nas escolas, *Sudoku Access*, que é adaptado com uma interface especial para que os estudantes com limitações físicas possam usufruir do jogo *Sudoku*. No final deste capítulo é evidenciado um estudo realizado numa escola básica com 3 estudantes, no qual o *Sudoku Access* foi utilizado;
- O capítulo 8 surge como o epílogo da dissertação, onde a síntese do trabalho é apresentada, bem como as futuras investigações e principais conclusões.

2

Problemas de Interação Pessoa-Máquina em Portadores de Deficiência Física

2.1 Introdução

As crianças de hoje fazem parte da primeira geração da “Era digital”. Estão a ser criadas numa sociedade que está em constante evolução, como resultado do influxo de novas tecnologias que providenciam uma ligação mais rápida e persuasiva ao comércio, comunicação e cultura. Cada vez mais, as escolas estão incumbidas de preparar todos os estudantes para utilizarem a tecnologia de uma forma que lhes permita competir num mercado de trabalho cada vez complexo do ponto de vista tecnológico. Muitas pessoas enaltecem a integração de tecnologias baseadas na computação nas salas de aula e reconhecem os muitos benefícios que estas podem oferecer a crianças com dificuldades motoras (Hasselbring & Glaser, 2000).

2.2 A Deficiência Física

“Dificuldades físicas que nos tornam dependentes dos outros. Mas dificuldades que não passam apenas por não mexer as pernas ou por não poder aceder a deter-

minados locais. Dificuldades que para além de nos invadirem o corpo nos invadem a alma...”.

Marta Reis, A Cegueira e a Deficiência Motora na Sociedade Actual, 2008.

As sociedades antepassadas não aceitavam a deficiência, levando à exclusão quase total destas pessoas. As famílias chegavam mesmo a escondê-las da sociedade, isolando-as do mundo. Felizmente, a evolução da sociedade levou a uma maior aceitação da deficiência devido ao aparecimento de novos pensamentos e mentalidades. Estas transformações aconteceram, em grande maioria, no final do século XIX e começo do século XX, durante a Revolução Industrial, com o aumento do interesse pela educação nos países desenvolvidos, o que levou ao início do atendimento aos deficientes, bem como ao aparecimento da educação especial destinada a um movimento de inclusão escolar e social. A sociedade aprendeu a ser mais inclusiva, compreensiva e solidária com a deficiência. Hoje em dia, as crianças com deficiência frequentam a escola, saem à rua, brincam, enfim, vivem como uma criança dita “normal”. No entanto, ainda há um longo caminho a percorrer para que todas as pessoas se sintam plenamente integradas.

A deficiência física motora refere-se ao comprometimento do aparelho locomotor, que compreende o sistema ósteo-articular, o sistema muscular e o sistema nervoso, afectando a capacidade de uma pessoa para se deslocar, manipular objectos e interagir com o mundo físico. As limitações físicas são inúmeras e podem ter origens diversas tais como serem resultantes de acidentes ou devidas a condições congénitas. Apesar de tanto as deficiências cognitivas como as físicas interferirem com a capacidade de interacção com aparelhos computacionais, neste capítulo focar-se-ão apenas as físicas, sendo que as mais relevantes para a presente discussão podem ser divididas em quatro categoriais (Jacko & Sears, 2003):

Alterações estruturais: Situações em que há uma alteração significativa (na posição ou dimensão) de uma parte do corpo ou perda parcial ou total de uma parte anatómica do mesmo (ausência de dedo, mão, braço);

Funções de mobilidade: Relacionadas com a capacidade de um indivíduo para mover as articulações ósseas, como por exemplo, dificuldade em mexer os dedos. Um exemplo é a Artrite, uma doença resultante da inflamação das articulações (Clough, 2006). Existem cerca de 200 tipos diferentes de Artrite, sendo as mais comuns a Osteoartrite, ou Artrose, e a Artrite Reumatóide. A Osteoartrite é uma perturbação crónica das articulações, sendo caracterizada pela degeneração da cartilagem e do osso adjacente, afectando principalmente os joelhos, os quadris, as mãos e a coluna vertebral. Além de provocar dores, sensações de rigidez e inchaço, também pode ocasionar limitações funcionais, tais como perda de movimentos, deformidades e incapacidade total do membro. Esta doença afecta principalmente pessoas de idade e diversas condições têm sido apontadas como responsáveis da Osteoartrite, particularmente as doenças metabólicas, factores genéticos, distúrbios anatómicos, alterações bioquímicas da cartilagem e membrana sinovial, sobrecarga mecânica, traumas e infecções (Arden *et al.*, 2008). A Artrite Reumatóide é uma doença auto-imune sistémica em que se inflamam simetricamente as articulações, incluindo habitualmente as das mãos e pés, originando inchaço, dor e muitas vezes levando à destruição definitiva do interior da articulação. Nesta doença, o sistema imunitário ataca o próprio tecido que reveste e protege as articulações, levando à deterioração da cartilagem, do osso e dos ligamentos da articulação, provocando a formação de cicatrizes dentro da articulação, que se deteriora a um ritmo muito variável. A Artrite Reumatóide apresenta-se em primeiro lugar em indivíduos entre os 25 e os 50 anos de idade, mas pode afectar pessoas de qualquer idade. Em alguns casos, a doença resolve-se de forma espontânea e o tratamento alivia sintomas em três de cada quatro pessoas. Contudo, pelo menos 1 em cada 10 pessoas fica incapacitada (Beers, 2006);

Funções musculares: Referentes à capacidade de um indivíduo para gerar força pela contracção de um músculo ou grupo de músculos, ocorrendo a disfunção ou interrupção dos movimentos de um ou mais membros: superiores, inferiores ou

ambos. Conforme o grau do comprometimento pode ter diferentes designações:

- Paresia: quando o movimento está apenas limitado ou enfraquecido e pode ser causado por várias condições. Um exemplo é a Paralesia Cerebral, ou Encefalopatia Crónica não progressiva, que é uma lesão de uma ou mais partes do cérebro, provocada pela falta de oxigenação das células cerebrais. Esta lesão ocorre durante a gestação, no momento do parto ou após o nascimento, ainda no processo de amadurecimento do cérebro da criança. É importante salientar que o portador possui inteligência normal (a não ser que a lesão tenha afectado áreas do cérebro responsáveis pelo pensamento e pela memória). No entanto se a visão ou a audição forem prejudicadas, a pessoa poderá ter dificuldades para entender as informações que lhe são transmitidas e se os músculos da fala forem atingidos, haverá dificuldade para comunicar os pensamentos ou necessidades. Quando tais factos são observados, o portador de paralisia cerebral pode ser erroneamente classificado como deficiente mental ou não inteligente. Para além das perturbações motoras, são frequentes nas pessoas com Paralisia Cerebral, o atraso cognitivo, perturbações visuais e auditivas, epilepsia, dificuldades de aprendizagem e défice de atenção;

- Paraplegia: quando ocorre perda total da capacidade para contrair os músculos e também pode ser derivada a inúmeras condições, como sendo as lesões irreversíveis da espinal medula. Como exemplos podem-se salientar a Paraplegia, Quadriplegia e a Distrofia Muscular. A Paraplegia pode classificar-se de completa ou incompleta, dependendo do facto de existir ou não controlo e sensibilidade periféricos abaixo do nível da lesão da pessoa em questão. Numa pessoa paraplégica normalmente isto traduz-se na perda de controlo e sensibilidade nos membros inferiores, impossibilitando a marcha e dificultando a posição de sentado e é possível que os membros afectados deixem de receber permanentemente qualquer tipo de estímulos, tornando os músculos flácidos, o que se traduz numa acentuada diminuição de massa muscular. Em determinados casos ocorre um fenómeno denominado de espasticidade, que mantém

os músculos activos através de movimentos involuntários, que podem tornar-se bastante incómodos e em determinadas situações limitar a vida activa ou até mesmo impossibilitá-la. Quando a paralisia afecta todos os quatro membros secundários, superiores e inferiores, é designada de Quadriplegia ou Tetraplegia (Ida, 2006). O doente tetraplégico é uma pessoa desprovida de auto-suficiência, que terá necessidade de assistência contínua durante toda a vida, necessitando da intervenção de outras pessoas não só para lhe assegurar uma certa mobilidade, mas também para realizar os actos quotidianos necessários à sobrevivência, como comer ou praticar a higiene pessoal. A Distrofia Muscular progressiva é uma doença de carácter hereditário, tendo como principal característica a degeneração da membrana que envolve a célula muscular, causando a morte. Dependendo do tipo de distrofia, esta afecta diferentes grupos de músculos e apresenta velocidade de degeneração variável. À medida que a doença evolui e a musculatura que proporciona sustentação para a coluna vertebral enfraquece, é comum ocorrerem escolioses (desvio tridimensional da coluna, o que significa que a coluna para além de desviar para um dos lados também faz rotação e inclinação) de gravidade variável. Estes fenómenos devem ser cuidadosamente acompanhados e em alguns casos corrigidos cirurgicamente devido ao seu potencial de restringir a capacidade respiratória. Numa fase mais adiantada, já no período da adolescência, a fraqueza muscular das pernas impedirá o jovem de caminhar. É nesta fase que se inicia o comprometimento cardíaco e respiratório, este último pelo progressivo acometimento do diafragma, dos músculos intercostais e da musculatura abdominal, essenciais na inspiração e no processo de tosse (Emery, 2000).

Funções de movimento: Relativas à habilidade que um indivíduo tem para controlar os movimentos voluntários e involuntários. A Disdiadocinesia, doença relacionada com a dificuldade para realizar movimentos rápidos alternadamente, enquadra-se na categoria dos movimentos voluntários. Por outro lado, o tremor não controlado das mãos, como os tremores relacionados com a

doença de Parkinson, é considerado como movimento involuntário. A doença de Parkinson é uma perturbação degenerativa e lentamente progressiva das células nervosas dos gânglios basais, e conseqüentemente do sistema nervoso, que provoca uma redução da produção de dopamina (o principal neurotransmissor dos gânglios basais), resultando na diminuição das conexões com outras células nervosas e músculos. Apresenta várias características particulares: tremor em repouso, lentidão na iniciação de movimentos e rigidez muscular e afecta maioritariamente indivíduos com mais de 50 anos (Duvoisin & Sage, 2001).

2.3 A Deficiência Física em Ambientes de Aprendizagem

“You only need to enter in a clothing store to see that the human being can have different shapes and sizes. It is only necessary to spend some time in a classroom to see that students could also have all the instructional shapes and forms....”

Marty Bray, Technology and Diverse Learner: A Guide to Classroom Practice, 2004.

Para pessoas com problemas motores, a necessidade de controlar os movimentos é um grande obstáculo que tem de ser ultrapassado para que possam interagir com o meio que as rodeia, sendo que a interação em ambiente de aprendizagem não é excepção. A tecnologia pode ser bastante eficaz na redução e supressão de restrições que impedem a realização de actividades quotidianas (ver Figura 2.1). De acordo com Valente, o computador surge como um poderoso aliado no processo educativo de alunos com necessidades educacionais especiais, uma vez que dispõe de recursos como animação, som, imagem, efeitos especiais, que superam as possibilidades didácticas e metodológicas tradicionais, tornando o processo de aprendizagem mais interessante e atractivo para os alunos (Valente, 1983). O computador também possibilita a adaptação às necessidades e capacidades do aluno, sendo possível a individualização do processo de ensino e aprendizagem. O processamento de texto, por exemplo,

surge como um recurso que pode simplificar muitas tarefas de escrita, como no caso da correção ortográfica. Para muitos estudantes com problemas de aprendizagem, os processadores de texto foram uma grande ajuda, uma vez que lhes permitiu uma nova liberdade para escreverem os seus pensamentos, sem terem de recorrer ao papel e à caneta ou à máquina de escrever. Actualmente, os processadores de texto encontram-se em quase todos os locais e a ferramenta de correção ortográfica está integrada virtualmente em todos os processadores de texto, e até mesmo em alguns *Web browsers*, permitindo ao indivíduo proceder a uma correção ortográfica antes de submeter o conteúdo de um formulário. Muitas outras tecnologias seguem o mesmo caminho, incluindo o discurso computadorizado, que pode ser utilizado por estudantes com problemas visuais para lerem os conteúdos de páginas da Internet (Asakawa & Itoh, 1998; Paciello, 2000).

Para além de auxiliar no processo educativo, o computador também pode ser um meio de acesso a toda uma série de acções/actividades, como sejam a comunicação, os serviços de informação, de documentação, trabalho, entre outros, sendo desta forma um importante veículo de inclusão social. No entanto, há que salientar que inicialmente os computadores não foram desenvolvidos para serem utilizados por pessoas com necessidades especiais. O computador não foi pensado para ser usado por um utilizador cego, surdo, com dificuldades locomotoras, etc., mas foi sim idealizado para um utilizador que possua movimentos precisos e meios sensoriais e cognitivos perfeitos. Visto desta forma, o computador nunca seria utilizado por um deficiente. No entanto, graças a estratégias e tecnologias de acessibilidade é possível tornar o computador uma ferramenta mais abrangente e de inserção social. Porém, há que adaptar esta ferramenta às características de cada utilizador. Desta forma, também nos ambientes de aprendizagem, é importante que o professor esteja atento às necessidades e características de cada estudante, observando as suas potencialidades, de forma a desenvolver adaptações e utilizar os recursos de acessibilidade já existentes.

As estratégias ou tecnologias que ajudam os estudantes a reduzir ou suprimir as

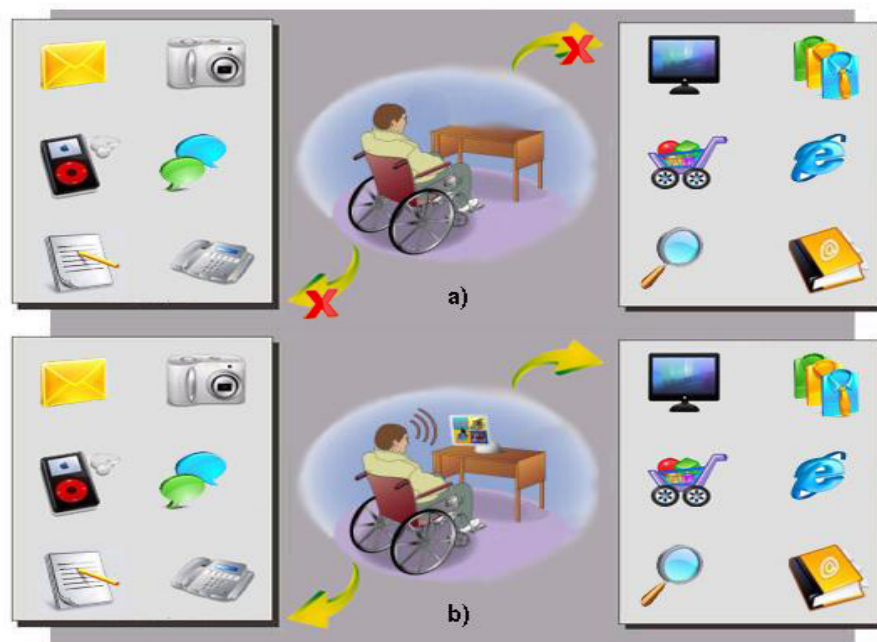


Figura 2.1: (a) São várias as dificuldades que os indivíduos com problemas motores podem enfrentar no mundo real, nomeadamente na realização de tarefas diárias e na integração na sociedade. (b) A tecnologia computacional surge como uma ferramenta essencial para abrir novas oportunidades para estas pessoas.

suas restrições físicas podem ser divididas em três tipos: **assistivas/adaptativas**, **correctivas** e **compensatórias**. No entanto, uma tecnologia pode ser enquadrada em diferentes grupos, dependendo do contexto em que está a ser utilizada (Bray & Green, 2004):

As **estratégias** e **tecnologias assistivas/adaptativas** ajudam os estudantes na realização de tarefas que de outra forma não conseguiriam realizar. Estas tecnologias podem ser simples ou complicadas. Uma palhinha que ajuda uma pessoa paralisada a beber de um copo é um exemplo de uma tecnologia assistiva simples. Uma cadeira de rodas computadorizada que pode ser controlada pelo sopro para um bocal, é um exemplo de uma tecnologia assistiva mais complexa. Tecnologias assistivas/adaptativas não são utilizadas para ensinar um conceito, mas sim para ajudar o indivíduo a aceder à informação necessária para aprender esse conceito. Por exemplo, os óculos graduados não ensinam

uma criança a ler; eles ajudam a criança a ver para que esta aprenda a ler.

As **estratégias e tecnologias correctivas** ensinam ou corrigem. Estas técnicas utilizam a repetição e simplificação de conceitos complexos para “corrigir” o estudante. Exemplos incluem *software* desenvolvido para ajudar estudantes a praticar habilidades específicas como ortografia, leitura e matemática. Este tipo de tecnologia é fácil de ser implementada na sala de aula, uma vez que requer pouca contribuição por parte do professor. No entanto requer que o professor entenda as necessidades educativas dos estudantes e as capacidades do *software*, para que possa fazer a ponte entre os dois. Os professores devem estar atentos para observarem quando já não há necessidade do uso do *software* e quando começam a ser necessárias actividades que levem ao desenvolvimento de um nível superior de raciocínio. Isto porque a ausência de compreensão dos problemas de um indivíduo com limitações físicas tem conduzido as pessoas a adoptarem determinados comportamentos que têm um impacto negativo no desenvolvimento da aprendizagem cognitiva destes indivíduos. Por exemplo, é comum encontrarmos pais e professores realizando diversas actividades com as crianças, como por exemplo, vestindo-as, escrevendo por elas, etc. Este tipo de postura leva a criança a ser mais “deficiente” à medida que a torna mais dependente e não lhe proporciona a possibilidade de desenvolver actividades que ela tem condições de realizar.

As **estratégias e tecnologias compensatórias** ajudam os estudantes na realização de uma tarefa académica de um modo mais eficaz e activo que se a realizassem sozinhos. Os exemplos incluem o processamento de texto (em vez da escrita à mão), *software* de verificação ortográfica e calculadoras. Estes instrumentos não só ajudam o estudante a realizar uma tarefa, mas também auxiliam na compreensão dos conceitos. A utilização de uma calculadora gráfica para ensinar conceitos estatísticos é um bom exemplo, uma vez que o estudante realiza operações simples, enquanto aprende o modo como operações mais complexas funcionam.

O sistema informático desenvolvido nesta dissertação, direccionado para o auxílio de pessoas com problemas motores a programarem facilmente numa linguagem de programação educativa, está enquadrado nas estratégias e tecnologias assistivas/adaptativas.

2.3.1 A Tecnologia Assistiva

Durante as décadas de 80 e 90, os direitos civis e educativos dos indivíduos com incapacidades foram fortalecidos. As tecnologias assistivas têm o potencial de providenciar oportunidades para aumentar a independência e a participação destas pessoas nas actividades do dia-a-dia. O governo norte americano, por exemplo, reconhece esta capacidade na seguinte declaração:

“...for all individuals, technology can provide important tools for making the performance of tasks quicker and easier, but for some individuals with disabilities, assistive technology is a necessity that enables them to engage in or perform many tasks.”

Office of Special Education and Rehabilitative Services (OSERS), 1989.

O termo Tecnologia Assistiva foi criado em 1988, tornando-se um importante elemento jurídico dentro da legislação norte-americana, conhecida como *Public Law 100-407*, tendo sido renovada em 1998 como *Assistive Technology Act* de 1998. Juntamente com outras leis, compõe o *ADA-American with Disabilities Act*, que regula os direitos dos cidadãos com deficiência nos EUA (Hawking, 2004). A Tecnologia Assistiva inclui ferramentas e acessórios, *hardware* e *software*, que permitem aos indivíduos com incapacidades físicas realizarem acções que de outra forma seriam difíceis, ou até mesmo impossíveis, de executar (Hlauschek *et al.*, 2009; Fairweather *et al.*, 2002; Liffick, 2003). São diversos os tipos de tecnologias assistivas, desde uma simples colher adaptada, até um complexo sistema de controlo de um computador através do reconhecimento ocular, passando pela tão familiar cadeira de rodas

(Hawking, 2004).

Em termos computacionais, a Tecnologia Assistiva auxilia os indivíduos em dois processos fundamentais, *input* e *output* (Robbins, 2006):

Dispositivos de *Input*

A Tecnologia Assistiva para fins de *input* providencia o mesmo tipo de funcionalidade que um teclado ou um rato. Alguns exemplos deste tipo de tecnologia são:

- Teclados alternativos: Providenciam um *layout* de teclas mais funcional, podendo apresentar características confortáveis para o utilizador como o código de cores, para ser usado por portadores de dificuldades cognitivas, e incluir teclas de dimensões variadas. Também podem apresentar um guia que facilite a selecção da tecla correcta, ou até mesmo serem desenvolvidos para poderem ser manipulados apenas com uma mão. Apresenta as desvantagens de ser um dispositivo bastante caro e de a utilização da combinação de teclas (*Shift + <Tecla>*) poder afigurar-se como um problema para utilizadores que manifestam tremores;
- Teclados Virtuais: Tipo de teclado que é exibido no ecrã para auxiliar pessoas que têm dificuldades em digitar, mas que conseguem usar eficazmente um rato ou um outro qualquer dispositivo de apontar. Estes teclados apresentam como vantagens o facto de poderem ter um *layout* dinâmico e de apresentarem uma série de opções como a possibilidade de aumentar o tamanho das teclas, alterar as suas cores, associar-lhes sons, entre outras. A grande desvantagem deste tipo de dispositivos é o facto do tamanho do teclado ficar restringido ao tamanho do ecrã do computador, o que por vezes pode condicionar a área de trabalho do utilizador;
- Software de reconhecimento da voz: É um sistema bastante útil para indivíduos que não conseguem utilizar um teclado e que não apresentam limitações

físicas na fala. Este *software* utiliza as capacidades do computador para detectar a voz de uma pessoa com dois propósitos principais: transcrever voz para texto e ouvir comandos operacionais. No entanto, é necessário perder algum tempo para treinar a voz no sistema, de forma a ser mais fácil o seu reconhecimento por parte deste. É conveniente que se utilize o microfone utilizado durante o treino da voz. É um sistema que não pode ser utilizado em qualquer lado: numa sala de aula, por exemplo, torna-se muito complicado utilizá-lo devido ao barulho do meio envolvente. É comum este tipo de ferramenta estar disponível ao nível de sistemas operativos.

- Varinhas para cabeça e boca: São utilizadas para digitar em teclados normais ou modificados. Estes instrumentos de *input* são regularmente utilizados com uma característica comum dos sistemas operativos conhecida como *sticky keys*, que permite ao utilizador pressionar e libertar uma tecla, como o *Ctrl*, e então pressionar outra tecla, tratando a sequência de teclas como se estivessem a ser pressionadas em simultâneo. Estas varinhas são dispositivos que podem ser adquiridos a preços razoáveis mas que apresentam a desvantagem de, quando manuseadas durante muito tempo, a sua utilização se tornar num processo cansativo para o utilizador;
- Face and eye tracking: Esta tecnologia geralmente utiliza *software* para seguir o olhar ou a face de uma pessoa que apresenta mobilidade limitada e está incapacitada de falar de forma suficientemente clara para utilizar o *software* de reconhecimento da voz. À medida que o utilizador move os olhos, o cursor do rato segue o olhar. Através deste sistema, muitas acções podem ser realizadas como o clique ou duplo clique. Uma desvantagem deste sistema é o facto de apresentar um custo muito elevado. É possível encontrar algum software com custo mais baixo mas que apresenta menor precisão. Outra desvantagem é o facto deste sistema não poder ser utilizado por uma população que apresente tremores incontroláveis;
- Switches adaptados: São mecanismos altamente especializados que, entre outras

funcionalidades, podem servir como ratos de botão único ou até mesmo serem utilizados como um conjunto de botões para os pés. Estes mecanismos podem ser utilizados em conjunto com *software* especializado para permitir que as pessoas tenham controlo total de todas as funções dos seus computadores. Um ponto negativo deste sistema é que por vezes pode-se tornar um pouco lento durante as tarefas.

Dispositivos de *Output*

As fontes normais de *output* para os computadores do dia-a-dia são os monitores e os altifalantes. Em alguns casos o uso de legendas ou transcrições pode ser bastante útil. No entanto existem certas situações onde se tem de recorrer a outros tipos de dispositivos ou *software* que facilite o *output*:

- *Screen readers*: São úteis para indivíduos que apresentam deficiências visuais, uma vez que estes programas interpretam o texto sintetizando a fala humana. A utilização deste sistema pode inicialmente ser complicada para o utilizador, sendo necessária alguma prática até se obter uma boa prestação. Em algumas aplicações, como sendo o *Flash*, este *software* pode não funcionar correctamente. Alguns exemplos deste tipo de ferramenta são *Window Eyes*, *GW Micro*, *HAL*, *Automated Living*, e *SuperNova*;
- *Screen magnifiers*: Utilizado por pessoas com problemas visuais, fornece uma imagem ampliada do contexto do ecrã. Como principal lacuna deste *software* pode-se salientar o facto de não ser aconselhado para ser utilizado em ecrãs de pequenas dimensões (Blenkhorn *et al.*, 2003). Exemplos deste sistema são o *Zoom-Text*, *SuperNova* e *MAGic (Freedom Scientific)*;
- *Aural browsers*: Esta ferramenta é similar aos *screen readers*, mas é especializada para utilização na Internet e providencia menos funcionalidades que os anteriores. Alguns exemplos são o *Home Page Reader* e *Browsealoud*;
- *Exibição Braille*: Converte o *output* do computador para *Braille*, exibindo as

letras através de *pins* amovíveis que representam a actual linha de exibição. Esta ferramenta é usualmente utilizada em conjunto com *screen readers*. Para além das vantagens óbvias para deficientes visuais, este tipo de *software* também manifesta algumas desvantagens: pode apresentar dificuldades de interpretação em imagens, tabelas ou mapas, e até mesmo dificuldades em abrir determinados tipos de ficheiros que não são compatíveis com o interpretador de *Braille*, o seu custo é elevado e a assistência técnica torna-se por vezes complicada.

2.4 A Deficiência Física e o Computador como Ferramenta de Aprendizagem

O computador no contexto educativo pode ser interpretado como uma ferramenta por meio da qual um indivíduo idealiza e desenvolve um conhecimento, quer seja reproduzindo um saber ou construindo uma aprendizagem. Deste modo, o processo de aprendizagem é estimulado pelo facto do aluno executar uma tarefa com o recurso ao computador. Este processo de aprendizagem é designado por “aprendizagem computacional”.

À semelhança de qualquer outro ambiente de aprendizagem, também nos ambientes de aprendizagem computacional os estudantes com problemas motores deparam-se com dificuldades, nomeadamente na interação com os dispositivos computacionais. O acesso ao computador pode tornar-se extremamente complicado, excluindo, muitas vezes, o aluno das tarefas que deveria efectuar, acabando este por ficar desmotivado. Como referido anteriormente, os estudantes com dificuldades motoras podem utilizar a Tecnologia Assistiva para compensarem as suas limitações no meio educacional, suprimindo barreiras do seu dia-a-dia de forma a controlarem o seu ambiente, a tornarem-se mais autónomos e a desenvolverem actividades como qualquer outro estudante. Através deste tipo de tecnologia, o aluno pode escrever num processador de texto, mover um *dispositivo de apontar* para poder interagir com o computador (criar/organizar pastas e documentos, jogar, ouvir música), navegar na

Internet e realizar tantas outras actividades. Ou seja, um ambiente computacional oferece a possibilidade de controlar e operar no mundo real aliviando parcialmente as limitações físicas.

A aprendizagem computacional abrange diversas actividades que podem ser benéficas no desenvolvimento cognitivo do indivíduo (Valente, 1983). Inserida na aprendizagem computacional encontra-se a programação, actividade importante que abrange um grande número de estudantes na nossa sociedade. As linguagens de programação são proveitosas na criação de actividades científicas, linguísticas, musicais, sistemas robotizados, telecomunicações e operações matemáticas (Resnick, 1997). O processo de aprendizagem de linguagens de programação e as dificuldades encontradas pelos estudantes constituem temas recorrentes nas investigações da área de educação e do ensino de computação (Govender & Grayson, 2008). A grande necessidade de auxiliar os alunos na aprendizagem de linguagens de programação tem motivado as pesquisas que visam o desenvolvimento de ambientes de aprendizagem informatizados (Papert, 1993a; Resnick *et al.*, 2009; Maloney *et al.*, 2008; Klopfer *et al.*, 2004).

Actualmente, na nossa sociedade poucos estudos têm sido realizados na aprendizagem de linguagens de programação para auxiliar estudantes com problemas motores. Devido à falta de meios tecnologicamente adaptados, muitos indivíduos permanecem frustrados pela falta de acessibilidade na área da programação de computadores.

2.5 Sumário

Para pessoas com problemas motores a necessidade de controlar os movimentos surge como um grande obstáculo para interagirem com o meio que as rodeia, sendo que a interacção em ambiente de aprendizagem não é excepção.

Neste capítulo foram descritos os problemas motores mais frequentes que podem afectar a interacção Pessoa-Máquina. A paraplegia, quadriplegia, paralisia cerebral

e distrofia muscular foram alguns dos exemplos referidos, sendo todas lesões de uma ou mais partes do corpo humano que afectam e perturbam a coordenação motora. Foram salientadas as estratégias ou tecnologias que ajudam os estudantes a reduzir ou suprimir as suas restrições físicas, sendo estas divididas em três tipos: assistivas/adaptativas, correctivas e compensatórias.

3

Linguagem de Programação *Logo*: Uma Filosofia de Aprendizagem

3.1 Introdução

“Logo is the name for a philosophy of education and a continually evolving family of programming languages that aid in its realization”

Harold Abelson, Apple Logo, 1982.

As linguagens de programação constituem uma actividade importante na área da informática, abrangendo um vasto número de estudantes. O processo de aprendizagem de linguagens de programação e as dificuldades sentidas pelos estudantes têm motivado as pesquisas que visam o desenvolvimento de ambientes de aprendizagem. Ao longo das últimas décadas, diversos investigadores colaboraram no desenvolvimento de diversas linguagens de programação (Hudak, 1989; Malik, 1998). A linguagem *Logo* destaca-se de todas as outras pelo facto de ter sido desenvolvida especificamente para crianças. O grande precursor desta filosofia de aprendizagem foi sem dúvida Seymour Papert.

Seymour Papert, colaborador de Jean Piaget na década de 60, desenvolveu uma filosofia de educação baseada no trabalho do seu mentor, a qual designou de *Cons-*

trucionismo. Esta difere da teoria *Construtivista* criada por Piaget, uma vez que aprofunda a ideia de “construção mental”, estudando com grande interesse a natureza do pensamento e a forma como as crianças se tornam pensadores (Papert, 1993a). O Construtivismo de Piaget, abre uma janela para o que as crianças estão realmente interessadas, e no que conseguem atingir, nos diferentes estágios do seu desenvolvimento. O Construcionismo de Papert, por outro lado, incide mais sobre a arte de aprender, ou “aprender a aprender”, e na importância de “fazer coisas” durante o processo de aprendizagem (Ackermann, 2001).

A partir da reflexão simultânea de como as crianças pensam e de que forma os computadores poderiam “pensar”, Papert idealizou uma linguagem de programação apropriada para crianças. No entanto, esta linguagem deveria possuir a mesma capacidade de outras linguagens de programação e ao mesmo tempo ser facilmente compreendida por principiantes sem domínio da matemática. Esta linguagem de programação foi designada de *Logo*. Muitos outros ambientes de aprendizagem foram posteriormente desenvolvidos, mas diversos estudos demonstraram as dificuldades com que as crianças se deparavam na sua utilização. Entre os ambientes revistos encontram-se *Boxer* (DiSessa & Abelson, 1986), *Stagecast Creator* (Smith *et al.*, 1994), *ToonTalk* (Kahn, 1996), *AgentSheets* (Repenning, 1993), *NetLogo* (Uri, 1999), *Squeak* (Ingalls *et al.*, 1997) e *Hands* (Pane *et al.*, 2002). A linguagem *Logo* certamente se destaca de todas as outras, tendo sido uma das primeiras linguagens a ser desenvolvida especificamente para crianças e tornando-se a mais influente (Sheehan, 2005).

Nos últimos trinta e cinco anos têm sido desenvolvidos vários ambientes de programação em *Logo* baseados no modelo Construcionista de Papert, com a intenção de proporcionar uma aprendizagem que siga os princípios desta corrente. Neste capítulo evidencia-se a criação da linguagem de programação *Logo*, focando principalmente as teorias construcionistas do seu criador. Para além disso, descreve-se um pouco da sua evolução, bem como as transformações sofridas e as suas aplicações ao longo dos tempos.

“Watch a baby explore the world. Everything within sight is seen, everything within touch is felt by hand, by foot and by mouth. The full range of available sound is emitted. The baby is exploring a small world, but is exploring it very thoroughly. It is obvious that a great deal is being learned. Educators talking about the way the learning happens would use words like: self-directed, experimental, nonverbal. I prefer to say “home-style learning”.

Seymour Papert, The Connected Family: Bridging the Digital Generation Gap, 1996.

Seymour Papert é um matemático e um dos pioneiros da Inteligência Artificial, sendo reconhecido internacionalmente como um pensador seminal acerca como os computadores podem mudar a aprendizagem. Nascido e educado na África do Sul, onde participou activamente no movimento *anti-apartheid*, Papert prosseguiu investigação na área da Matemática na Universidade de Cambridge desde 1954 até 1958. Entre os anos de 1958 e 1963, Papert trabalhou com Jean Piaget na Universidade de Geneva, onde começou a considerar o uso da matemática para a compreensão do modo como as crianças aprendem e pensam. Foi no início dos anos 60 que Papert se afiliou ao MIT onde, juntamente com Marvin Minsky, fundou o *MIT Artificial Intelligence Laboratory*. Em 1967, Papert desenvolveu a linguagem de programação *Logo* e nos anos 80 definiu a sua teoria construcionista, baseada no trabalho de Jean Piaget, John Dewey e Maria Montessori. Em 1985 ajudou a fundar o programa *Media Arts and Sciences* no *MIT Media Laboratory* e em 1988 foi nomeado *LEGO Professor of Learning Research*. Actualmente, Papert é considerado o maior especialista do mundo relativamente ao modo como a tecnologia pode proporcionar novas formas de aprendizagem (Papert, 1993a,b, 1996b). Foi no seu laboratório onde as crianças tiveram pela primeira vez a oportunidade de usar o computador para escrever e fazer gráficos. A sua actuação tem passado pela realização de projectos educativos em todos os continentes, alguns deles em aldeias remotas e em países subdesenvolvidos.

O trabalho de Papert demonstra a abordagem do construcionismo, uma filosofia de aprendizagem que expande o conceito de construtivismo de Piaget. De acordo com a corrente construtivista da psicologia do desenvolvimento, o sujeito aparece como um construtor activo do seu próprio conhecimento. O construtivismo argumenta que os professores devem compreender a essência activa do processo de aprendizagem, no qual os estudantes já estão empenhados, de modo a estarem aptos a poderem facilitar e enriquecer esse processo, ao invés de tentarem impor-lhes experiências que não fazem sentido. O pensamento construcionista acrescenta algo ao ponto de vista construtivista. Onde o construtivismo indica o sujeito como construtor activo e argumenta contra modelos passivos de aprendizagem e de desenvolvimento, o construcionismo dá particular ênfase a construções próprias do indivíduo, que são externas e partilhadas (Harel & Papert, 1991), (Ackermann, 2001). De acordo com Resnick “o construcionismo adiciona a ideia de que as pessoas constroem novo conhecimento de forma mais eficiente quando estão comprometidas a construir objectos com valor pessoal” e que “esta visão coloca o construcionismo (e não a informação) no centro da análise” (Resnick, 1996). Assim sendo, a visão que Papert tem do Homem e do mundo situa-se numa perspectiva interaccionista entre os dois, sendo o conhecimento o produto desta interacção que está centrada no modo como o mundo cultural actua e influencia o sujeito em interacção com o objecto.

O trabalho que Papert tem desenvolvido tem-se focado essencialmente no processo de aprendizagem infantil (Papert, 1972). Para Papert, o processo educativo tem como pressuposto que a criança não aprende apenas pelo ensino formal e imposto. A criança é uma aprendiz inata, que mesmo antes de chegar à escola apresenta conhecimentos adquiridos por meio de uma aprendizagem natural, espontânea e intuitiva, que se dá através da exploração, da busca e da investigação, a qual pode ser caracterizada como uma auto-aprendizagem. Aquilo que a criança aprendeu porque fez, após ter explorado, investigado e descoberto por si própria, além de contribuir para o desenvolvimento das suas estruturas cognitivas, reveste-se de um significado especial que a ajudará a reter e a transferir, de um modo muito mais fácil, aquilo

que foi aprendido (Papert, 1996a).

3.1.1 O Papel do Construcionismo na Integração das Tecnologias na Educação

Para Papert, o computador é defendido como uma poderosa ferramenta educativa, onde o aluno actua e participa activamente no seu processo de construção de conhecimentos, interagindo com um versátil instrumento de aprendizagem, fácil de operar e rico em possibilidades (Papert, 1993b). Os computadores têm o potencial de auxiliar no processo de aprendizagem de diversas formas: podem ser utilizados para reforçar o ensino nas áreas curriculares tradicionais, podem ser usados como ferramentas para aumentar a produtividade do aluno ou até mesmo serem empregues na exploração de novas abordagens de aprendizagem que não poderiam ser concebidas em ambientes sem computadores (Watt, 1988). Foi com base nesta convicção de informática educativa, onde a criança, aprendiz inata, constrói as suas estruturas de pensamento a partir da exploração e da realização de novos processos de pensar, que Papert desenvolveu a linguagem de programação *Logo* para auxiliar os alunos a construir, a estruturar e a reestruturar o seu pensamento (Papert, 1976). Na filosofia do *Logo*, está implícita a ideia construcionista de que a aquisição do conhecimento é realizada pela interacção das pessoas com o meio que as rodeia. Durante o processo de programação em *Logo*, é o indivíduo que controla a sua aprendizagem, tomando várias decisões, sendo que o resultado destas decisões irá influenciar a escolha de outras. Outro dos aspectos do *Logo* que demonstra a presença da teoria construcionista de Papert é o facto do “erro” ser utilizado como forma de aprendizagem: a criança é incentivada a procurar, a investigar, a explorar e a decidir sozinha qual a melhor forma de ultrapassar um “obstáculo”. Através do *Logo*, o aluno aprende princípios, técnicas e habilidades que o ajudam no processo de aprendizagem e na resolução de problemas. Com esta ferramenta, é possível criar simulações, animações, apresentações, jogos gráficos, textos, controlar dispositivos externos (robótica), com a vantagem de proporcionar a integração curricular (Papert, 1993b).

Inicialmente, Papert encontrou muita resistência nas escolas ao uso da tecnologia. Actualmente, já não há tanta resistência uma vez que a tecnologia já é mais barata e já se encontra mais acessível às escolas (Papert, 1996a). Inclusive, em muitos lugares já existe o ensino à distância, onde as crianças nem sequer se reúnem todas numa sala de aula. Para Papert isto não é o fim da escola em si, mas o início de uma nova escola. O currículo escolar divide os conteúdos a aprender por idades, o que para Papert não faz sentido, e será decerto substituído por um sistema onde o conhecimento vai sendo aprendido à medida que o aluno está preparado para o obter. Deve ser dado às crianças o poder de controlar a sua aprendizagem. Se isso acontece em casa e é tão natural, porque não nas escolas? Existem computadores na maioria dos lares e as crianças já exploram e aprendem por conta própria fora das escolas, porque não levar isso para as salas de aula, já que é tão natural? (Papert, 1972). Papert compara a introdução dos computadores e da Internet nas salas de aulas com o uso de um lápis: pode ser usado para fazer “asneiras”, mas pode também ser usado para escrever e desenhar, ou seja criar; tal como os computadores que quando bem canalizados, fomentam a aprendizagem. Defende ainda que os governos de todo o mundo devem fazer esforços para proporcionar às crianças materiais de primeira qualidade para a aprendizagem (referindo-se às novas tecnologias claro) e, para que isso seja possível, os países ricos devem apoiar os países pobres. Isto porque, fomentando o conhecimento e combatendo a ignorância é possível criar um mundo mais pacífico ou com mais investigação a nível de, por exemplo, investigar a cura para determinadas doenças.

3.2 A Linguagem de Programação *Logo*

“In most contemporary educational situations where children come into contact with computers the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the Logo environment the relationship is reversed: The child, even at preschool ages, is in control: The child

programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think. The experience can be heady: Thinking about thinking turns the child into an epistemologist, an experience not even shared by most adults.”

Seymour Papert, Mindstorms: Children, Computers, and Powerful Ideas, 1993

3.2.1 História da Linguagem Logo

Fundamentada nas teorias construcionistas de Seymour Papert, a linguagem de programação Logo foi desenvolvida em 1967 na *Bolt Beranek and Newman, Inc.*, por Wallace Feurzeig, Daniel Bobrow e Seymour Papert, com o objectivo de expandir e estruturar o conhecimento de crianças e adolescentes. A denominação Logo foi utilizada como referência ao termo grego Lógos (pensamento, raciocínio, discurso) para designar tanto a linguagem de programação, quanto a abordagem educativa que sustenta a sua utilização (Papert, 1976; Papert, 1993b). Grande parte deste estudo teve a colaboração de grandes investigadores do *Massachusetts Institute of Technology (MIT)* como Marvin Minsky e John McCarthy.

Falar da história do Logo é como falar da história de uma vida, cheia de encruzilhadas, progressos, recuos e paixões. Muitos investigadores apareceram como verdadeiros “apaixonados”, evitando a todo o custo o desaparecimento do “espírito Logo”. A *Logo Foundation* apresenta a verdadeira história do Logo. A mim cabe-me a tarefa de salientar os seus aspectos mais relevantes.

Durante os anos 70, o Logo esteve confinado ao MIT e a poucos outros centros de investigação como o *Departamento de Inteligência Artificial* da Universidade de Edimburgo e o *Instituto de Educação da Universidade de Londres*, tendo sido desenvolvidas apenas pequenas pesquisas em escolas locais. Naquela época, ainda não havia microcomputadores e as únicas implementações de Logo existentes eram destinadas a equipamentos de maior porte. Isso significava que, para que uma criança pudesse utilizar o Logo, teria que se deslocar a uma das poucas universidades in-

teressadas em investigar o impacto do computador no processo de aprendizagem infantil. O ambiente de programação *Logo* mais popular envolve uma tartaruga virtual e, durante este período inicial, esta apresentava-se como um dispositivo robótico que se sentava no chão e que era comandada através de comandos seleccionados no computador.

No final dos anos 70, com o aparecimento dos computadores pessoais, o uso do *Logo* expandiu-se para fora dos laboratórios. Diversos projectos pioneiros envolvendo populações estudantis começaram a ser desenvolvidos. A partir dos anos 80, novas versões do *Logo* começaram a ser implementadas em mais de uma dúzia de idiomas e em diversas máquinas, muitas delas com grafismos de jogos de computador e capacidades sonoras. O *Logo* para computadores MSX, conhecidos computadores domésticos dos anos 80, tornou-se popular na Europa, na América do Sul e no Japão, enquanto que *Atari Logo* e *Commodore Logo* popularizaram-se na América do Norte. Em meados dos anos 80, os computadores com capacidade para videojogos começaram a aparecer no mercado, levando consigo as versões do *Logo*. Por esta altura as máquinas *MSDOS* começaram a dominar o mundo da educação computacional, excepto nos Estados Unidos onde a *Apple* era a favorita das escolas. Assim sendo, os criadores do *Logo* voltaram as suas atenções para este tipo de máquinas. Apesar das novas implementações adicionarem novas funcionalidades e tirarem vantagem da maior velocidade de processamento e superior capacidade de memória dos novos computadores, as versões mais populares do *Logo* em 1985 eram bastante similares às de 1980. Por volta desta altura também começou a surgir o interesse em utilizar o *Logo* como uma linguagem de programação “séria”, principalmente para os novos computadores *Macintosh*. O *MacLogo* da *Logo Computer Systems, Inc* adicionou novas funcionalidades ao ambiente *Logo*. O *Coral Software* desenvolveu uma versão *Logo* orientada a objectos denominada *Object Logo*, que incluía um compilador que possibilitava aos programas serem executados a alta velocidade. No entanto o *Logo* não se tornou popular entre os programadores.

Em 1985 a *Logo Computer Systems, Inc.* introduziu o *Logo Writer* que apresen-

tava diversas novidades. Primeiro, possuía a capacidade de processamento de texto. Segundo, a interface com o utilizador apresentava-se mais simplificada. O *Logo Writer* também incluía múltiplas tartarugas que podiam incluir diferentes formas. Esta versão foi implementada em vários idiomas, tornando-se popular em todo o mundo. Outra inovação dos meados dos anos 80 foi o *LEGO Logo*. Foi desenvolvido um sistema para interacção do *Logo* com motores, luzes e sensores que estavam incorporados em máquinas construídas de peças *LEGO* e de outros elementos. Posteriormente, a *Harvard Associates* desenvolveu o *PC Logo* para *DOS* e mais tarde para *Windows*.

Ao longo dos últimos anos tem surgido uma nova onda de desenvolvimento do *Logo* acompanhada de entusiasmo e de uma renovada consciencialização pública. Para além dos desenvolvimentos na Europa e América Latina, o aumento do interesse nos Estados Unidos e no Canadá tem sido elucidado pela introdução do *MicroWorlds*. Lançado em 1993 pela *Logo Computer Systems, Inc.*, este apresenta grandes alterações tanto no ambiente como na própria linguagem, incluindo muitas características extras ao *Logo* — ferramentas de desenho, editor de formas, capacidade para inserir gráficos e sons, etc — que permitem, juntamente com o *Logo*, a criação de projectos multimédia, jogos e até mesmo simulações. De todas as alterações apresentadas pelo *MicroWorlds*, talvez a mais significativa seja a capacidade de multi-tarefa, ou processamento paralelo, onde vários processos podem ser lançados independentemente, o que é essencial aquando da criação de animações com mais de um “actor”.

Control Lab e *Control System* são novos produtos *LEGO Logo* cujo *software* multi-tarefa está construído sobre o mesmo núcleo que o *MicroWorlds*. Outra inovação *LEGO Logo* é o *Programmable brick*, um projecto de investigação liderado por Fred Martin. Ao contrário dos produtos *LEGO Logo* convencionais onde o computador recebe instruções através de cabos conectados ao computador, o *Programmable brick* possui um computador no seu interior. Outra versão paralela do *Logo* é o *StarLogo* que foi desenvolvido pela equipa liderada por Mitchel Resnick do MIT.

Neste caso, milhares de tartarugas podem desempenhar processos independentes e podem interagir entre elas. Recentemente, têm sido desenvolvidas inúmeras versões comerciais do *Logo*, incluindo o *Logo Grafico* na Argentina e o *Mach Turtles Logo* no Canadá. *UCBLogo* foi desenvolvido por Brian Harvey e trata-se de uma versão de domínio público para *Macintosh*, *MSDOS* e *Unix*. George Mills utilizou o núcleo do *UCBLogo* como base para o seu *MSWLogo* que é processado em Windows. O *Logo* também tem vindo a ser incorporado no *HyperStudio*, o popular programa multimédia para *Macintosh* e *Windows* da *Roger Wagner productions*.

Após três décadas de crescimento, o *Logo* sofreu alterações dramáticas devido ao desenvolvimento rápido da tecnologia dos computadores, sendo que a família dos ambientes *Logo* nunca foi tão diversa como agora. No entanto, o *Logo* permanece como um movimento mundial de pessoas que partilham um compromisso com a filosofia educativa do construcionismo.

3.2.2 Características da Linguagem *Logo*

O *Logo* é uma linguagem de programação simples, flexível e poderosa, sendo também um ambiente de aprendizagem interactiva e exploratório, direccionado para a resolução de problemas matemáticos. Trata-se de uma linguagem de programação ideal para ilustrar três formas importantes do modo como os computadores podem ser utilizados para melhorar o processo de aprendizagem. Em primeiro lugar, apresenta-se como um domínio através do qual as ideias fundamentais da ciência da computação elementar podem ser facilmente introduzidas. Em segundo lugar, a “Geometria da Tartaruga”, o ambiente matemático de aprendizagem construído em *Logo*, ilustra algumas das formas através das quais microambientes computacionais podem promover a aprendizagem exploratória de ideias matemáticas. Em terceiro lugar, pode ser utilizado pelos professores para o desenvolvimento de actividades exploratórias de aprendizagem relacionadas com objectivos curriculares específicos (Watt, 1988; Papert, 1996b). As aplicações do *Logo* encontram-se em constante desenvolvimento e, embora dedicado à educação, *Logo* não é uma linguagem apenas

para crianças, pois possui excelentes recursos que podem ser usados em trabalhos nos mais diversos campos científicos como sendo controlo do meio, robótica e inteligência artificial (Sipitakiat & Blikstein, 2010; Kumar, 2004).

Existem muitas características marcantes da linguagem *Logo* mas podem ser destacadas aquelas que a tornaram numa linguagem de fácil assimilação:

De acordo com (Papert, 1976) as principais características da linguagem *Logo* são:

- **Amigabilidade e simplicidade:** Apresenta-se como uma linguagem de programação de fácil aprendizagem e uso;
- **Modularidade e extensibilidade:** Permite a especificação e criação de novos comandos utilizando a própria linguagem *Logo*;
- **Capacidade e Flexibilidade:** É uma linguagem de programação poderosa, apresentando ferramentas essenciais para criar programas com diferentes graus de sofisticação, sendo uma linguagem útil tanto para utilizadores pré-escolares como para estudantes de graus de ensino mais elevados;
- **Interactividade:** Oferece uma resposta imediata e mensagens informativas sobre o comando aplicado;
- **Constatação do erro:** O utilizador aprende com os seus erros, sendo que a correcção de erros não aparece como uma tarefa desagradável;
- **Linguagem de procedimentos:** A criação de novos termos ou procedimentos é extremamente fácil, contribuindo para o desenvolvimento dos princípios de criação e invenção das crianças.
- **Grafismo:** Os recursos gráficos tornaram-se a parte mais conhecida da linguagem *Logo*.

3.2.3 O Ambiente da Programação *Logo*

Considerando o *software*, *Logo* é uma linguagem muito inspirada em *Lisp*, muito usada em inteligência artificial. O *Logo* caracteriza-se pela presença de um cursor gráfico representado pela figura de uma tartaruga, que pode ser deslocada no espaço do ecrã do computador através de comandos primitivos, permitindo ao sujeito lidar com o espaço bidimensional. Nos primeiros tempos do seu desenvolvimento, esta linguagem era utilizada para transmitir comandos que possibilitavam o movimento de um robot em forma de tartaruga. Este robot possuía uma caneta e quando colocado em cima de uma folha de papel permitia o desenho de figuras. O formato de tartaruga foi adoptado para que os que contactam pela primeira vez com esta linguagem de programação se identifiquem e se coloquem no lugar da tartaruga (Papert, 1993b).

Em *Logo* pode-se trabalhar de dois modos: directo e programa. No modo directo são utilizados comandos primitivos, que já vêm implementados com a linguagem, e a tartaruga executa imediatamente cada instrução digitada. No modo programa são desenvolvidos procedimentos (novos programas que podem fazer parte dos comandos primitivos da linguagem) que envolvem diversos vocabulários da tartaruga que poderão ser reaproveitados ao longo da construção de projectos em *Logo*. Os procedimentos são executados quando chamados na área de trabalho, tal como se de comandos primitivos se tratassem. Tanto os comandos primitivos como os procedimentos podem ser chamados dentro de outros procedimentos. O *Logo* também possui comandos que permitem manipular palavras e listas, permitindo comandar a tartaruga para produzir texto, o que integrado com a parte gráfica permite fazer animações (Watt, 1988). A digitação dos comandos pode ser feita interactivamente ou através de um editor de texto (Sheehan, 2005).

Durante a edição do procedimento, são digitados os comandos que serão executados através do nome que lhe foi atribuído. Se é solicitado à tartaruga que execute um TRIÂNGULO e nenhum procedimento com esta designação foi construído, surgirá no ecrã uma mensagem de erro. No caso do procedimento TRIÂNGULO existir

e se este não apresentar erros de sintaxe então o procedimento é executado. A acção de construir um procedimento é designada por “*ensinar a tartaruga*”. Como o *Logo* é uma linguagem de procedimentos, novos procedimentos e termos podem ser criados e armazenados para serem usados posteriormente. Por exemplo, pode-se criar um quadrado ordenando à tartaruga para ir em frente e virar 90 graus à direita, quatro vezes de seguida. Depois essas instruções podem ser colocadas dentro de um único grupo (um procedimento) e assim quando se quiser fazer um quadrado basta chamar o procedimento.

A maioria das pessoas tem uma visão distorcida da linguagem *Logo* pela mística de ser mais conhecida como uma linguagem infantil e limitada por possuir um vocabulário de fácil acesso. Este é um ponto de vista erróneo, uma vez que o *Logo* permite a criação de procedimentos, aumentando a capacidade e a facilidade de programação (Papert, 1993b). Uma parte do vocabulário da linguagem *Logo* está destinado a facultar meios simples para o tratamento de problemas espaciais. Para isso emprega recursos que, no seu conjunto, são designados de *Geometria da Tartaruga* (Abelson & Sessa, 1981).

A *Geometria da Tartaruga* é um estilo computacional de geometria. Para estudar este tipo de geometria é necessário a apresentação da já referida *tartaruga virtual*. A tartaruga virtual é uma criatura simples que apenas sabe a sua posição, direcção e a sua escala. Obedece a comandos simples para alterar a sua localização, orientação e noção de escala (R. Goldman & Ju, 2004). Considerando esta tartaruga, a sua localização num plano pode ser descrita por um ponto P , representado por um par de coordenadas $(p1, p2)$; de forma semelhante, o vector direcção pode ser descrito pelo vector w , representado por outro par de coordenadas $(w1, w2)$. A escala da tartaruga é simplesmente o comprimento do vector w . O par (P, w) é designado como o “estado da tartaruga”.

A tartaruga responde a seis comandos básicos: FORWARD, BACK, LEFT, RIGHT, PENDOWN e PENUP . Cada comando apresenta uma função específica:

- **FORWARD (FD):** A tartaruga move-se para a frente (para onde se encontra direccionada) e pode desenhar uma linha desde a sua posição inicial até à sua posição final;
- **BACK (BK):** A tartaruga move-se para a trás e pode desenhar uma linha desde a sua posição inicial até à sua posição final;
- **LEFT (LT):** A tartaruga altera a sua direcção, rodando no plano no sentido anti horário;
- **RIGHT (RT):** A tartaruga altera a sua direcção, rodando no plano no sentido horário;
- **PENDOWN (PD):** A tartaruga disponibiliza uma caneta para desenhar a sua trajectória;
- **PENUP (PU):** A tartaruga remove a caneta e deixa de desenhar a sua trajectória.

Através destes comandos principais é possível desenhar formas interessantes. Os programas mais simples apenas utilizam várias combinações de FD, LT e RT para desenharem diferentes figuras. Por exemplo, utilizando os comandos FD e RT ou LT, é possível criar um quadrado. O comando “REPEAT” permite repetir varias vezes uma instrução reduzindo assim a escrita dos comandos (ver Figura 3.1).

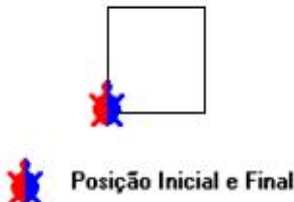
<i>PROCEDIMENTO</i>	<i>RESULTADO</i>
<pre>to Square repeat 4 [forward 50 right 90] end</pre>	 <p>Posição Inicial e Final</p>

Figura 3.1: O procedimento “*Square*” permite a construção da figura geométrica quadrado.

3.3 Sumário

Neste capítulo focou-se essencialmente a forma como Seymour Papert foi educado e os fundamentos das suas teorias, nomeadamente o Construcionismo, que assenta basicamente numa aprendizagem por conta própria, através da exploração e pesquisa, encontrando erros que sendo solucionados também ajudam a fortalecer o desenvolvimento cognitivo. Uma vez que um conhecimento adquirido desta forma é mais natural e permite que a sua divulgação posterior seja melhor fundamentada, Papert defende uma profunda mudança no sistema de ensino, onde a integração de novas tecnologias é um dado inevitável. Seguindo estes princípios, Papert criou a linguagem de programação *Logo*, com vista a desenvolver nas escolas estruturas e meios que permitissem aos alunos aprender por descoberta e não como se fossem uns contentores prontos para receber o que o professor ensina. O *Logo* é uma linguagem que assenta basicamente na introdução de comandos não muito complicados, que permitem a movimentação de um ícone — uma tartaruga. Através dos comandos dados às tartarugas é possível aos alunos entrarem nos campos da exploração espacial, numérica e geométrica e, trabalhando com os sucessos e os erros obtidos, desenvolvem capacidades nestes campos. Esta linguagem permite fazer animações, e manipular listas e palavras, o que, para além dos trabalhos com crianças, possibilita o desenvolvimento de projectos em muitas áreas. Papert afirma que quando a criança está a interagir com o *Logo* está a criar o seu próprio micro mundo, onde interage e visualiza mentalmente os objectos que constroi.

4

Evolução da Tecnologia Construcionista em Ambientes de Aprendizagem

4.1 Introdução

Actualmente existem diversas formas de utilização dos computadores no processo de aprendizagem. Neste contexto, uma tecnologia que tem vindo a ser bastante utilizada é a robótica educativa que, devido às suas características, abrange condições didácticas e pedagógicas extremamente atraentes e motivadoras. A robótica educativa tem como principal objectivo impulsionar o estudo de conceitos multidisciplinares como a mecânica, física, matemática, informática, inteligência artificial, entre outros. Assim, através da utilização das ferramentas essenciais para o desenvolvimento de dispositivos robóticos, os alunos e os professores trabalham os conceitos das diversas disciplinas, demonstrando na prática a aplicabilidade dos conceitos teóricos. Na robótica educativa o aluno trabalha geralmente com *kits* de montagem compostos por diversas peças, como motores e sensores, que são controladas por um *software* que permite programar os protótipos montados, proporcionando ao aluno a oportunidade de desenvolver e explorar a sua criatividade (O'Hara & Kay, 2003; Martin *et al.*, 2000a). Pretende-se desta forma que o aluno questione,

pense e procure soluções, saindo da teoria para a prática através dos conhecimentos adquiridos na sala de aula ou na vivência quotidiana.

Neste capítulo, está exposta uma breve síntese dos principais dispositivos de robótica educativa que têm surgido ao longo dos anos, dando especial ênfase à *GoGo Board* que foi a principal fonte de inspiração do trabalho desenvolvido nesta tese.

4.2 A Robótica Educativa

4.2.1 A Tartaruga *Logo* - o início da robótica educativa

A ideia de utilizar a robótica no meio educativo surgiu com Seymour Papert no início dos anos 70, através do desenvolvimento da linguagem *Logo* que, como já referido anteriormente, nos seus primórdios apresentava uma tartaruga robotizada (ver Figura 4.1). Esta tartaruga surgiu como uma tentativa de dar uma “expressão corporal” à programação *Logo*, facultando um nível de comunicação com o computador através do movimento. Assim, o computador passou a ter “pernas”, “braços” e até alguns “sentidos” como sendo o tacto, através de sensores de toque, e a “visão”, através de sensores de luminosidade (Papert, 1993b). Com a evolução dos computadores pessoais, a comunidade *Logo* voltou a sua atenção para as “tartarugas de ecrã”. As crianças continuavam a usar comandos como *forward* e *back*, mas agora estes comandos serviam para controlar pequenas imagens gráficas no ecrã do computador e não robôs mecânicos.

Posteriormente, o conceito de *Logo* evoluiu para o ambiente de robótica *LEGO/Logo*, resultante do trabalho conjunto de um grupo de pesquisadores do *Epistemology and Learning Group* do *MIT* e da empresa di-

namarquesa *LEGO*, que comporta uma variedade de actividades de design (Resnick



Figura 4.1: A tartaruga robotizada controlada por crianças utilizando a linguagem *Logo* (Papert, 1993b)

et al., 1988). Surgia assim uma nova era direccionada para uma visão educativa mais interactiva.

4.2.2 O Sistema *LEGO/Logo*

O sistema *LEGO/Logo* é uma plataforma que faz a ponte entre o mundo da construção em *LEGO* e o mundo da programação em *Logo*, permitindo a construção de dispositivos programáveis. Utilizando este sistema, as crianças começam por construir engenhos feitos com peças *LEGO*, utilizando não apenas as tradicionais peças de construção *LEGO*, mas também peças alternativas como pneus, motores, engrenagens, roldanas, lâmpadas e até mesmo sensores. Após a construção dos protótipos, estes são conectados a um computador através de uma interface e são controlados através de programas escritos numa versão modificada da linguagem *Logo*. Os novos comandos são novas primitivas associadas às primitivas já existentes, incluindo palavras como “*on*” e “*off*” para controlar os motores e luzes *LEGO*, e palavras como *sensors* para retirar informação dos sensores *LEGO* (Resnick & Ocko, 1991). A interface computador/dispositivo estabelece um canal de comunicação que permite a troca de informação nos dois sentidos: informação que parte do computador e chega ao dispositivo e vice-versa. O sistema *LEGO/Logo* ficou comercialmente disponível no final dos anos 80, começando a ser vendido para as escolas com o nome de *Legote Logo*.

Apesar de ter surgido como uma fantástica inovação, este sistema apresentava algumas limitações. Uma delas é o facto dos protótipos desenvolvidos pelas crianças terem de estar obrigatoriamente conectados a um computador através de fios. No caso do modelo construído possuir apenas um motor, esta limitação não era muito inconveniente. No entanto, se este possuísse um par de motores ou sensores, cada um conectado ao computador através do seu próprio cabo, poder-se-ia tornar numa confusão de cabos, o que obviamente limita a capacidade móvel dos protótipos (Martin *et al.*, 2000a).

A necessidade de recursos informáticos com características específicas, tais como

baixo consumo energético, portabilidade e dimensão reduzida, impulsionou a evolução da robótica educativa para o conceito de *Programmable Brick* (“Tijolo Programável”).

4.2.3 Programmable Bricks

Ao mesmo tempo que o *LEGO/Logo* estava a entrar nas escolas, o grupo do *MIT Media Laboratory* começava a pensar no seu sucessor. Obviamente que o próximo passo passaria por colocar electrónica nos próprios blocos *LEGO*. A questão que se colocava era se conseguiriam inserir um computador dentro de um bloco tão pequeno que pudesse ser transportado por um modelo *LEGO*. A resposta a esta questão tornou-se afirmativa em 1987, quando Fred Martin, Randy Sargent e Brian Silverman, sobre a supervisão de Seymour Papert e Mitchel Resnick, e com a colaboração da empresa *LEGO*, construíram os primeiros protótipos dos *Programmable Bricks*. Os *Programmable Bricks* são microcomputadores portáteis inseridos em peças *LEGO* do tamanho de um baralho de cartas (ver Figura 4.2).

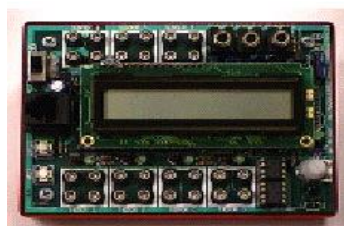


Figura 4.2: Programmable Brick.

Estes podem ser programados para interagirem com o mundo exterior através de sensores de temperatura, toque, som e luminosidade ou para controlarem dispositivos como motores e actuadores (Resnick *et al.*, 1996). Desta forma torna-se possível a sua utilização em projectos que necessitem de pequenos computadores, que operam independentemente, nos campos da robótica, controlo ambiental, acessibilidades, etc. Uma das vantagens deste tipo de sistema é que, devido ao facto de serem utilizadas peças *LEGO*, as crianças podem criar o objecto que pretendem controlar, não estando confinadas a um que tenha uma forma pré-definida.

São várias as características associadas a este tipo de plataforma, de entre as

quais podem-se salientar:

Versatilidade: Podem ser utilizados nas mais diversas áreas;

Capacidade de ligação a múltiplos dispositivos: A ligação aos mais variados mecanismos, como sensores e actuadores, permite uma maior adaptação à finalidade pretendida;

Capacidade de processamento paralelo: Permite controlar simultaneamente diversos dispositivos;

Capacidade de interagir com outros *Programmable Bricks*: Permite a interligação de diversos dispositivos de *hardware*.

Baseado nos princípios especificados, diversos tipos de *Programmable Bricks* têm sido desenvolvidos e as mais diversas aplicações têm-lhes sido inculcidas. Desde o controlo ambiental, onde objectos interagem e “reagem” à acção das pessoas, passando pela robótica, onde crianças constroem animais e brinquedos com as peças de *LEGO* programadas em *Logo*, até projectos onde jovens criam robots inteligentes que respondem a estímulos como luzes ou circulam livremente controlando os seus próprios movimentos. Os *Programmable Bricks* também se encontram ligados à biometria, onde podem regular e informar parâmetros relacionados com os seres vivos, nomeadamente na medição das pulsações (Resnick *et al.*, 1996).

Desde a sua invenção, muitas variantes do conceito de *Programmable Brick* têm sido implementadas com diferentes objectivos funcionais, como sendo o *Red Brick*, o *LEGO Mindstorms* e o *Cricket* (Martin *et al.*, 2000b).

4.2.4 Logo Brick

Os primeiros *Programmable Bricks* surgiram na versão de *Logo Bricks*. Nesta versão, o protótipo construído era conectado a um computador para ser programado na linguagem *Logo* e posteriormente era desconectado para ser utilizado autono-

mente, permitindo o controlo de até quatro motores e a recepção de informação de vários sensores (ver Figura 4.3).

Um conjunto de experiências realizado com o *Logo Brick* em estudantes está descrito em “*Children, cybernetics, and programmable turtles*” (Martin, 1988). Neste estudo, as crianças construíram projectos associando o *Programmable Brick* com uma tartaruga *LEGO*, e escreveram programas para transmitirem diferentes comportamentos à tartaruga, como por exemplo o desvio de obstáculos. Prestando atenção à relação das crianças com a tecnologia, verificaram que algumas crianças tratavam a tartaruga como um animal de estimação, e gostavam quando este exibía um comportamento imprevisível. Outras crianças estavam mais interessadas em que a tartaruga executasse um conjunto específico de acções e foram desafiadas pela dificuldade em consegui-lo.

Esta primeira geração de *Programmable Bricks* foi um sucesso na medida em que permitiu estudar a interacção das crianças com este tipo de dispositivo. No entanto foi necessário esperar pelo desenvolvimento de outros *Programmable Bricks*, para que estes pudessem efectivamente serem utilizados nas salas de aula.

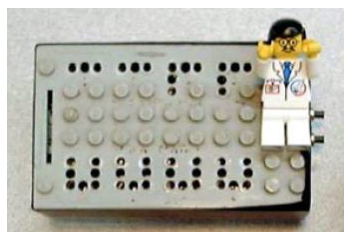


Figura 4.3: Logo Brick.

4.2.5 Red Brick

Entre 1994 e 1996, o mesmo grupo do *MIT Media Laboratory* desenvolveu uma segunda geração de *Programmable Bricks*, a qual ficou conhecida como *Red Brick* devido a possuir um invólucro de plástico vermelho (ver Figura 4.4). Este dispositivo foi desenvolvido especificamente para ser robusto e fácil de construir. Mais de 100 cópias foram fabricadas para uso em escolas e centros comunitários (Martin *et al.*,

2000a).

O *Red Brick* não era muito diferente do *Logo Brick* original. No conceito básico da sua utilização, as crianças construía um protótipo em *LEGO* e depois conectavam-no ao computador para a sua programação. As crianças podiam transcrever directamente os comandos *Logo* para o *brick*, ou desenvolver procedimentos *Logo* e fazer o *upload* destes. Durante este período, enquanto o *Red Brick* estava a ser usado extensivamente, o grupo do *MIT Media Laboratory* começou a estudar as características do dispositivo que poderiam ser melhoradas ou alteradas no desenvolvimento de um novo tipo de *Programmable Brick*. Os investigadores chegaram à conclusão de que as maiores alterações seriam ao nível do tamanho e do peso. O próximo dispositivo deveria ser mais pequeno e mais leve. Relativamente ao número de *inputs* e *outputs* que o dispositivo deveria possuir, concluíram que a configuração do *Red Brick* era essencialmente a correcta, assim como a presença de um ecrã LCD. Surgiu desta forma um novo produto denominado de *LEGO Mindstorms* que, ao contrário dos outros que estavam direccionados para escolas, pretendeu levar esta tecnologia directamente para as crianças nas suas casas.



Figura 4.4: Red Brick.

4.2.6 LEGO Mindstorms

O *LEGO Mindstorms* foi disponibilizado comercialmente em 1998 através da versão 1.0 dos *kits Robotics Invention System (RIS)*, tornando-se imediatamente num sucesso global e atraindo tanto crianças como adultos. Esta versão original era constituída por 717 componentes de entre eles um *Programmable Brick* conhecido por *RCX Brick* (um microprocessador de 8 bits), dois motores, dois sensores de

toque e um sensor de luz (Patterson-McNeill & Binker, 2001).

O *Robotics Invention System (RIS)* evoluiu para as versões 1.5 e 2.0 que, apesar das melhorias ao nível do *software*, se baseavam essencialmente no mesmo RCX e nos mesmos tipos de motores e sensores (ver Figura 4.5).

Para além do bloco RCX, outros *bricks* foram sendo gradualmente desenvolvidos até se chegar à versão actual NXT, sendo que em 2006 foi lançado o *LEGO Mindstorms NXT* e em 2009 foi disponibilizada a versão mais recente conhecida por *LEGO Mindstorms NXT 2.0*. Com o lançamento deste NXT, algumas mudanças significativas foram feitas, sendo a maior delas o próprio micro controlador, um processador de 32 bits. Para além disso, esta versão vem ainda equipada com três novos motores, um sensor de toque, um sensor de luz (que reage à luminosidade e à cor), um sensor ultrasónico e um sensor de distância. Novas tecnologias, como a transferência de dados por *Bluetooth*, e capacidades sensoriais aperfeiçoadas elevam o nível de sofisticação destes sistemas e desafiam a capacidade inventiva dos melhores e mais criativos criadores de robôs.

No entanto, o tamanho do *Red Brick*, que deu origem ao *LEGO Mindstorms*, mostrou-se inadequado em certas aplicações, levando ao desenvolvimento de um tipo de *Programmable Brick* muito mais pequeno e económico. Este novo sistema ficou conhecido como *Cricket*.



Figura 4.5: Lego Mindstorms RCX.

4.2.7 Cricket

O *Cricket* é uma das plataformas mais utilizadas por investigadores que pretendem estender o uso de *Programmable Bricks* a novos domínios da aprendizagem (ver Figura 4.6). Cada *Cricket* apresenta o tamanho aproximado de uma pilha de 9 volts e pode controlar autonomamente dois motores e receber informação de dois sensores, sendo que o seu *software Logo Cricket* foi desenvolvido especificamente para este sistema. Possui ainda a capacidade de comunicação através de infra-vermelhos que permite a comunicação com um computador e a comunicação entre *Cricket*s. A presença de uma porta de expansão periférica amplia a capacidade de comunicação do *Cricket* com outros dispositivos electrónicos. Três *LEDs* de estado transmitem a indicação de ligado/desligado, programa a decorrer e transmissão de infra-vermelhos em progresso (Martin *et al.*, 2000b).

Display Cricket, *Tag Cricket* e *The bit Ball* surgiram como novas versões do *Sensor/Motor Cricket* original, apresentando algumas alterações a nível de *hardware*.

A incrível flexibilidade deste tipo de sistema para assumir quase todas as formas físicas e adoptar uma grande variedade de comportamentos programados, combinada com a crescente variedade de sensores e actuadores tornam o sistema num instrumento de robótica rápido e versátil.

Todas estas variantes da robótica educacional facultam um ambiente caracterizado pela tecnologia e criatividade que estimula a aprendizagem construcionista de conceitos. Os *kits* comerciais, como sendo os da linha *LEGO Mindstorms*, disponibilizam uma elevada quantidade de peças e acessórios que auxiliam a montagem dos dispositivos robóticos, levando a criatividade ao limite. No entanto, o seu elevado custo surge como factor responsável pela baixa receptibilidade para a introdução desta tecnologia na maior parte das escolas.

Com o objectivo de tornar a robótica educacional uma ferramenta economicamente ao alcance de todos e com o intuito de possibilitar a sua utilização por pessoas

com poucos conhecimentos de electrónica e programação, surgiu a *GoGo Board*, uma plataforma de *hardware* desenvolvida por dois estudantes do *MIT Media Laboratory*, Arnan Sipitakiat e Paulo Blikstein (Sipitakiat *et al.*, 2002).

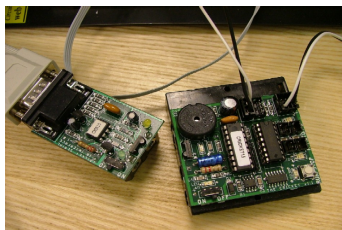


Figura 4.6: Cricket.

4.3 Robótica Educacional de Baixo Custo: *GoGo Board*

A ideia de desenvolver um dispositivo de robótica educacional relativamente barato mas que se regesse pelos mesmos princípios dos *Programmable Bricks*, surgiu quando Sipitakiat estava a fazer a sua tese de mestrado na Tailândia. Durante o desenvolvimento do seu trabalho, Sipitakiat verificou que os *Programmable Bricks*, que eram de grande utilidade para as escolas, apresentavam o inconveniente de serem muito dispendiosos para os orçamentos bastante limitados dos estabelecimentos de ensino. Antes de considerar a compra de computadores, assim como de qualquer outro material tecnológico, o conselho de gestão das escolas tem de atender a problemas mais básicos como a falta de livros e outro material escolar, fazer a manutenção e melhoria das infra-estruturas escolares, pagar os salários dos funcionários, entre outros (Sipitakiat *et al.*, 2002). Recorrendo aos seus conhecimentos em electrónica, Sipitakiat desenvolveu a *GoGo Board*, um *Programmable Brick* de baixo custo que apresenta um conjunto interessante de características que torna acessível a todos as actividades de aprendizagem da robótica (Sipitakiat & Blikstein, 2010):

Versatilidade: O dispositivo é versátil de forma a possibilitar uma elevada diversidade de utilizações, como sendo a construção de robôs, a medição e o armazenamento de dados ambientais ou a criação de instalações interactivas;

Open-source: O formato da placa é do tipo *open-source* ou “código aberto”, significando que qualquer pessoa pode recriar ou modificar o dispositivo, de acordo com o fim pretendido, mas não o pode comercializar. Assim como o seu *hardware*, também o *software* da placa *GoGo Board* é *Open-Source* sob a licença GPL (*General Public Licence*).

Fácil construção: Esta placa está desenhada de forma a proporcionar uma montagem fácil e acessível para qualquer pessoa, até mesmo para indivíduos com poucos conhecimentos ao nível da electrónica, possibilitando ao utilizador participar activamente no processo de construção. Por exemplo, o circuito impresso na placa (*Printed Circuit Board (PCB)*) foi especialmente concebido para tornar simples o processo de soldadura dos componentes. O *design* da placa também foi bastante ponderado, excluindo tecnologia de ponta que tornaria as peças difíceis de encontrar, para que esta pudesse ser fabricada localmente em vários países. No entanto, há que salientar o facto de que, infelizmente, ao contrário do que é anunciado pelos seus criadores, nem todos os componentes electrónicos são de tão fácil acesso, sendo que alguns não são comercializados na Europa;

Baixo custo: A *GoGo Board* surge como um alternativa aos dispendiosos *kits* comerciais, como sendo o *RCX Brick* da *LEGO Mindstorms*. Os componentes electrónicos escolhidos para a construção deste dispositivo são de baixo custo, estando assim economicamente mais acessíveis a todos. Para além do custo dos componentes, também, durante a sua criação, os seus autores tentaram minimizar o número de componentes obrigatórios na placa, de forma a reduzir ainda mais o seu custo. Desta forma, vários componentes são opcionais, podendo ser adicionados apenas quando necessários.

Assim como outros dispositivos pertencentes à família dos *Programmable Bricks*, também a *GoGo Board* permite a interacção entre o mundo digital e o mundo físico. Neste contexto, o utilizador pode conectar à placa vários sensores e actuadores

e pode escrever programas para lerem os dados do sensor e para controlarem o comportamento de vários objectos físicos utilizando motores, pequenas lâmpadas, LEDs e relés (Sipitakiat *et al.*, 2004).

O *layout* básico de uma *GoGo Board* está representado na Figura 4.7 e é composto essencialmente por:

- 1 – Memória EEPROM32KB
- 2 – Porta série
- 3 – Oito entradas
- 4 – Quatro saídas
- 5 – Beeper programável
- 6 – LED programável
- 7 – BUS de extensão

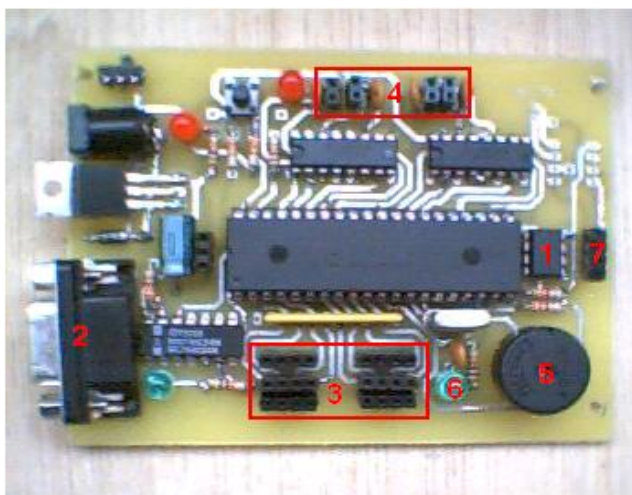


Figura 4.7: Características principais da *GoGo Board*.

Em termos de linguagem de programação, a linguagem *Logo* tem sido a mais utilizada para interagir com a *GoGo Board*. Os criadores desta plataforma disponibilizam aos utilizadores bibliotecas *Logo*, para *LCSI Microworlds* e *Image Logo*, para que estes possam controlar as suas placas. Outras bibliotecas também estão disponíveis para outros ambientes comerciais de programação incluindo *Squeak*, *Visual Studio* (*Visual Basic*, *Visual C++*) e *Microsoft Office* (*Excel*, *Access*, *Power Point*, *Word*). Ainda há a possibilidade de outras linguagens de programação comunicarem com a *GoGo Board*, mas o programador tem de estudar o protocolo da placa. Esta plataforma também já foi utilizada com *Macromedia Director* e *Java*.

Relativamente ao modo de operação, a *GoGo Board* pode funcionar de duas formas (Sipitakiat *et al.*, 2002, 2004):

Modo ligado ao computador, em que a placa permanece sempre conectada ao computador, o que providencia elevadas capacidades de sensibilidade e controlo

para qualquer linguagem de programação que suporte comunicação através da porta serie. Assim sendo, os estudantes podem desenvolver projectos científicos ou sistemas interactivos sofisticados que se baseiem no maior poder de processamento do computador;

Modo autónomo, onde o utilizador cria o programa e faz o *upload* destes procedimentos para a memória da placa, que pode assim funcionar desconectada do computador. Este modo é muito vantajoso para, por exemplo, a criação de robôs autónomos e outros objectos/ambientes inteligentes.

Independentemente do modo de operação da *GoGo Board*, vários têm sido os projectos, mais ou menos complexos, que têm vindo a ser desenvolvidos em todo o mundo. Segue-se uma síntese de alguns projectos realizados nos Estados Unidos da América, Brasil e Portugal, utilizando a versátil e pouco dispendiosa *GoGo Board*.

4.3.1 Projectos Realizados com a *GoGo Board*

Corrida de Cavalos

Quando se pretende referir exemplos da utilização da *GoGo Board* não se pode omitir uma das suas principais aplicações: o *design* e a interface de jogos de computador. Neste caso, os utilizadores criam vários controlos físicos para interagir com os jogos desenvolvidos no computador. Um destes exemplos é o projecto denominado de “Corrida de Cavalos”, projecto este que foi desenvolvido nos Estados Unidos da América e que assenta num jogo onde um cavalo e um homem competem entre si numa corrida (ver Figura 4.8). O cavalo corre com uma velocidade aleatória, enquanto a velocidade do homem é controlada por um sensor que é um acelerador meramente construído com folha de alumínio e pauzinhos de gelado (Sipitakiat *et al.*, 2002).

Dance-Dance-Revolution

O projecto Dance-Dance-Revolution é uma réplica do famoso jogo musical que consiste em pisar uma série de sensores colocados no chão de acordo com

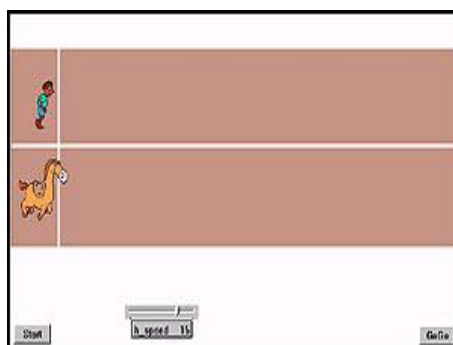


Figura 4.8: Interface do jogo “Corrida de Cavalos” desenvolvido com o *software Microworlds* e a *GoGo Board*.

instruções exibidas no ecrã. Neste caso os sensores são feitos com papel grosso e folha de alumínio (Sipitakiat *et al.*, 2002). À semelhança da “Corrida de Cavalos”, também o “Dance-Dance-Revolution” foi desenvolvido nos Estados Unidos da América.

Casa de Banho Inteligente

A utilização da *GoGo Board* para o desenvolvimento de instalações interactivas tem vindo a prosperar ao longo dos anos. Um exemplo bastante interessante deste tipo de aplicação foi desenvolvido no Brasil, por um grupo de professores que pretendeu ensinar noções de higiene aos seus alunos através do desenvolvimento de uma casa de banho inteligente (ver Figura 4.9). Tudo começou quando os professores notaram que alguns alunos, após utilizarem a casa de banho, se esqueciam de puxar o autoclismo. Assim, criaram um modelo de uma casa de banho que possuía um sensor de infra-vermelhos na porta, para detectar a entrada de alguém, e um sensor de toque no autoclismo, que detectava quando alguém o descarregava. Desta forma, quando a criança saía da casa de banho sem descarregar o autoclismo era gerada uma mensagem auditiva a lembrá-la da sua responsabilidade (Sipitakiat *et al.*, 2002).

Escala de Calorias

Com o objectivo de melhorar a consciência das pessoas relativamente ao número de calorias presente nos alimentos, um grupo desenvolveu um projecto denom-

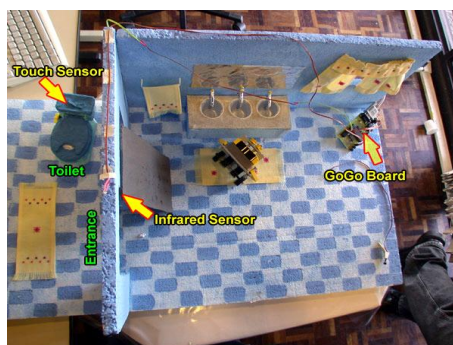


Figura 4.9: Protótipo da “Casa de banho inteligente” desenvolvida no Brasil.

inado de “Escala de Calorias” (ver Figura 4.10). A ideia subjacente a este projecto era criar uma escala de medidas do peso do alimento e posteriormente converter o peso em calorias. Os sensores foram construídos com folha de alumínio, sendo que um sensor indicava qual o tipo de alimento a ser pesado, e outro indicava o peso do alimento em estudo. Um conjunto de botões foi também acoplado ao sistema para permitir ao utilizador seleccionar o tipo de comida que estava a pesar. O valor de calorias é impresso no ecrã do computador (Sipitakiat *et al.*, 2002).

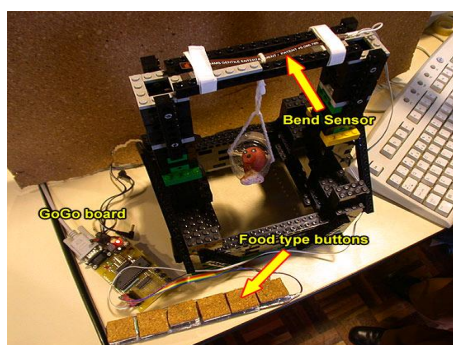


Figura 4.10: Escala de Calorias.

Auxílio a Deficientes Motores (ADM)

Com o intuito de utilizar a *GoGo Board* e a programação em *Logo* para auxiliar indivíduos com problemas motores na realização de diversas tarefas do dia-a-dia, foi desenvolvido pelo estudante Stéphane Norte, na Universidade do Algarve, Portugal, o *software* “ADM”, abreviatura de “Auxílio a Deficientes

Motores” (ADM) (Norte *et al.*, 2005).

Como em todas as situações onde o desenvolvimento de um produto é direccionado, também no decorrer deste projecto houve a necessidade de perceber as reais necessidades dos utilizadores finais deste *software*. Este primeiro objectivo foi alcançado através da visita às instalações da Associação Portuguesa de Paralisia Cerebral (APPC), em Faro, onde se pôde observar em concreto o *software* e os recursos já existentes que apresentam a mesma finalidade do nosso projecto: auxiliar os deficientes motores nas suas actividades diárias. No entanto, apesar de ser inegável que o material existente nesta associação é eficiente, não se pode deixar de salientar que tanto o *software* como os dispositivos de *hardware* existentes são poucos e extremamente caros. Por exemplo, um simples botão de toque chega a rondar os 500 euros, o que se deve claramente à sua especificidade e à pouca oferta de mercado.

O objectivo do programa ADM é o desenvolvimento de uma plataforma de baixo custo que permita dar autonomia a pessoas com problemas motores, especialmente indivíduos que apresentem dificuldades em manusear um simples rato ou teclado convencional, para que estas pessoas possam controlar o meio que as rodeia. Assim sendo, o grupo optou por implementar um sistema automático, onde a escolha das operações pretendidas é feita através de um único toque num botão de pressão. Resumidamente, o ADM apresenta um *layout* bastante simples composto por imagens de possíveis acções que o utilizador possa querer realizar, como por exemplo abrir uma porta, ir à casa de banho ou ligar o ar condicionado. O sistema automático do ADM vai seleccionando individual e sequencialmente cada uma das acções através de uma cor, até ao momento em que o utilizador pressiona o botão de toque para realizar essa tarefa. Este método é designado por varrimento (Majaranta & Rähkä, 2002; Norte *et al.*, 2005). Por exemplo, se o utilizador pretende abrir uma porta deve esperar que a cor se encontre na opção “Abrir Porta” e então pressionar o botão de toque para activar o mecanismo (ver Figura 4.11).



Figura 4.11: *Software* “Auxílio a Deficientes Motores” (ADM).

Para melhor ilustrar o potencial do *software* desenvolveu-se uma maquete do quarto de um possível utilizador (ver Figura 4.12).



Figura 4.12: Maquete do quarto de um possível utilizador

Todo este projecto tornou-se fundamental para o desenvolvimento e implementação do trabalho apresentado nesta dissertação, uma vez que foi nesta fase que emergiu uma pergunta pertinente: será que uma pessoa com deficiência física também consegue programar em *Logo*? Para poder responder a esta questão e antes de poder estruturar e implementar uma ferramenta que pudesse auxiliar indivíduos a escrever os seus próprios programas, foi importante entender as potencialidades desta linguagem tanto ao nível educacional como ao nível do vocabulário utilizado no *Logo*. Surgiu assim a necessidade de realizar um estudo com a linguagem *Logo* que envolvesse crianças.

4.4 Sumário

No decorrer das últimas décadas, as ideias subjacentes à teoria construcionista de Seymour Papert têm despoletado o desenvolvimento de ambientes robóticos educativos, que pretendem estimular a aprendizagem de conceitos multidisciplinares através da utilização de ferramentas essenciais para o desenvolvimento de dispositivos robóticos, demonstrando na prática a aplicabilidade dos conceitos teóricos. O aluno trabalha com *kits* de montagem compostos por diversas peças, como motores e sensores, que são controladas por um *software* que permite programar os protótipos montados, proporcionando ao aluno a oportunidade de desenvolver e explorar a sua criatividade.

Neste capítulo, foram demonstradas várias opções de robótica educativa que têm surgido mundialmente, sendo que tudo começou com a famosa Tartaruga *Logo* que deu, pela primeira vez, uma “expressão corporal” à programação *Logo*, facultando um nível de comunicação com o computador através do movimento. Posteriormente o conceito de *Logo* evoluiu para a plataforma LEGO/*Logo* que proporciona uma acção educativa mais interactiva através da construção de dispositivos programáveis, utilizando para tal *kits* de robótica constituídos por peças LEGO e peças alternativas. A necessidade de recursos informáticos com características específicas, tais como baixo consumo energético, portabilidade e dimensão reduzida, impulsionou a evolução da robótica educativa para o conceito de *Programmable Bricks*, microcomputadores portáteis inseridos em peças *LEGO* do tamanho de um baralho de cartas. Ficou também evidenciado que vários *programmable bricks* têm surgido ao longo do tempo desde o simples *Logo Brick* até ao mais complexo e autónomo sistema LEGO Mindstorms, passando pelas plataformas Red Brick e Cricket. Para finalizar foi apresentada a *GoGo Board*, que surgiu recentemente como uma alternativa de baixo custo aos *Programmable Bricks*, tendo sido salientados diversos projectos desenvolvidos com esta placa.

5

Aprender *LOGO* no Ensino Secundário: Construcionismo versus Objectivismo

5.1 Introdução

Actualmente, a aprendizagem e o ensino das ciências da computação nas escolas secundárias estão geralmente baseadas em duas filosofias de ensino: o objectivismo e o construcionismo. O objectivismo tem dominado o campo da educação ao longo de vários anos. Um educador objectivista acredita que a aprendizagem ocorre geralmente quando os estudantes ouvem a explicação do professor (Fosnot, 1996; Skinner, 1953). Este paradigma tradicional manifesta-se, muitas vezes, num professor centrado, portador do poder total nas salas de aula. Sob esta abordagem pedagógica, o instrutor serve como um detentor do saber que é a principal fonte de conhecimento e a aprendizagem é vista como um procedimento de informação na qual o instrutor transmite os seus conhecimentos aos estudantes. Por outro lado, como anteriormente demonstrado, o construcionismo concentra-se mais nas experiências do aluno. O pressuposto básico e fundamental do construcionismo é que o conhecimento não existe independente do aluno, o conhecimento é construído pela exploração experimental, ou seja, fazendo coisas (Kafai & Resnick, 1996; Papert, 1993b, 1996b).

Com o aparecimento das novas tecnologias nas escolas, os professores podem seguir um modelo de ensino menos tradicional, empregando uma abordagem mais construcionista (Gorp & Grissom, 2001). No entanto, algumas escolas ainda não estão preparadas tecnológica e pedagogicamente para incentivar os professores a utilizarem este tipo de tecnologias numa base frequente e sustentada de forma a melhorar a aprendizagem dos estudantes (Watt, 1988). Não se trata apenas da utilização do *PowerPoint*, folha de cálculo ou processador de texto, mas sim direccionar-se para estratégias de ensino e aprendizagem que marquem a diferença num domínio prático das actividades.

Utilizando a linguagem de programação *Logo* na sala de aula, foi-nos permitido exercer uma excelente introdução do mundo do ensino das ciências da computação e entender as possíveis dificuldades sentidas pelos alunos com este tipo de linguagem (Alimisis, 2007; Jochemczyk *et al.*, 2007). De forma a estudar a interacção dos alunos com esta linguagem, optou-se por utilizar a abordagem objectivista, ou tradicional, numa turma e a abordagem construcionista noutra turma. Assim sendo, neste capítulo está apresentado o modo como foi conduzido este estudo, bem como os resultados obtidos utilizando as duas abordagens pedagógicas distintas, revelando assim as potencialidades da programação em *Logo* no meio educacional. É ainda demonstrado um estudo realizado com a Associação Portuguesa de Paralisia Cerebral onde se desenvolveu uma interface que possibilita às crianças com paralisia cerebral utilizarem a linguagem de programação *Logo*.

5.2 Descrição do Estudo

Este estudo baseou-se essencialmente em determinar quais as maiores dificuldades e facilidades sentidas pelos jovens programadores na aprendizagem da linguagem de programação *Logo*. Pelo facto da linguagem *Logo* ser uma filosofia baseada na abordagem construcionista e por ter tido o relato positivo de vários investigadores nessa área, resolvi aplicar o estudo do *Logo* em dois ambientes distintos, o construcionista e o objectivista (Beynon & Roe, 2004; Stager, 2007; Meerbaum-

Salant & Hazzan, 2010; Apiola *et al.*, 2010; Valente, 2003). O estudo foi conduzido com duas turmas. No ambiente objectivista as aulas foram praticamente apenas expositivas e conduzidas pelo professor. Todas as actividades foram planeadas e impostas aos alunos passo a passo. No ambiente construcionista as aulas foram essencialmente práticas. No início foram apenas revelados alguns trabalhos em *Logo* e demonstrado o mecanismo da linguagem. Todas as actividades foram desenhadas pelos alunos sendo o professor, na maior parte do tempo, um mero espectador. O carácter deste estudo não incidiu fundamentalmente na preocupação de recolher dados quantitativos, podendo até mesmo afirmar que este foi o objectivo menos importante. Este estudo teve como principal objectivo recolher informação prática baseada sobretudo na observação e colecta de opiniões dos alunos para esmiuçar a interacção da linguagem *Logo* com os utilizadores. A realização de um exame final permitiu recolher dados quantitativos que serviram apenas para classificar os alunos para o currículo da disciplina.

5.3 População em Estudo

A população deste estudo resume-se a duas turmas do 10^o ano da escola secundária de Loulé onde leccionava aulas de informática. As turmas em questão foram-me atribuídas porque na planificação anual do curso teriam de aprender uma linguagem de programação. A população em estudo não foi escolhida por mim apenas me foi atribuída. Os alunos de ambas as turmas nunca tinham aprendido uma linguagem de programação e apresentavam a particularidade de manifestarem grandes dificuldades de aprendizagem, falta de concentração e mau comportamento dentro da sala de aula. Os dois grupos são referenciados como turma O e C e pertencem ao curso de multimédia com um currículo especialmente adaptado. A turma O é composta por 19 estudantes de idades compreendidas entre os 16 e 20 anos, sendo 7 raparigas e 12 rapazes. A turma C é constituída por 21 estudantes com idades compreendidas entre os 16 e 23 anos, sendo 6 raparigas e 15 rapazes. É importante sublinhar que 65% dos estudantes apresentam uma idade avançada para

frequentar o 10º ano de escolaridade. Em circunstâncias normais estudantes do 10º ano deveriam apresentar idades compreendidas entre os 15 e 16 anos. O facto de alguns deles serem muito mais velhos do que o normal é precisamente devido às suas dificuldades de aprendizagem e problemas comportamentais. Uma grande fracção dos estudantes reprovou no final do ano escolar e desta forma, vai ter de repetir a disciplina no ano seguinte. Cerca de 60% dos alunos já reprovaram pelo menos 2 vezes. Cada turma teve aulas de programação em *Logo* utilizando diferentes paradigmas. Na turma O optou-se por utilizar uma abordagem educativa tradicional e na turma C recorreu-se a uma abordagem construcionista. O estudo foi realizado ao longo de 5 meses com 180 minutos de aulas semanais.

5.4 Ambiente Tradicional

A turma O iniciou o contacto com a programação *Logo* ouvindo as explicações do professor e respondendo a motivações externas. Foram utilizados pequenos vídeos e apresentações em *PowerPoint* para explicar a introdução dos movimentos da tartaruga no programa *Logo*. O interpretador de *Logo* utilizado no início do curso foi a versão gratuita do *MicroWorlds*. Foi explicado aos estudantes como usar a *interface* do *MicroWorlds* e diversas aulas foram utilizadas para descrever as funcionalidades e ferramentas deste *software*. Após a explicação, cada aluno experimentou o ambiente do *MicroWorlds* com o auxílio de um professor. Uma vez aprendido o essencial do *software*, começou-se a introduzir os vocabulários da programação *Logo*. Com um projector de vídeo foi apresentado algum vocabulário e procedimentos simples para desenhar formas geométricas. Os estudantes deram instruções simples à tartaruga, sendo uma delas definida por nós, utilizando a tartaruga com simples vocabulários da linguagem *Logo* (ver Figura 5.1).

Após o desenvolvimento destas formas, os alunos foram estimulados a realizarem algumas formas comuns, tais como o quadrado, triângulo, pentágono, hexágono, octógono e heptágono. Os estudantes revelaram muita dificuldade na criação do hexágono, heptágono e octógono, sendo que apenas dois estudantes foram capazes

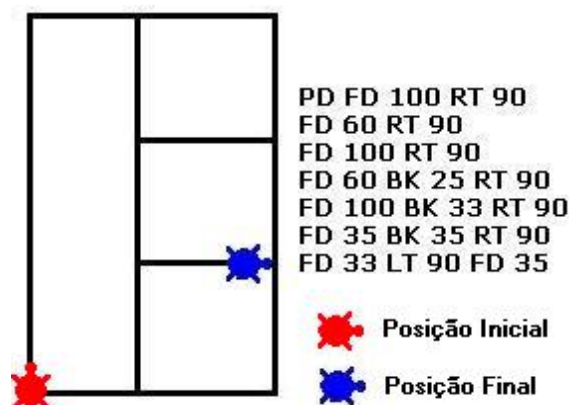


Figura 5.1: Primeira figura desenvolvida em *Logo* pelos alunos.

de executar as formas geométricas sem qualquer tipo de ajuda.

Após a criação destas formas geométricas, foi introduzida a ideia básica dos procedimentos em *Logo*. Esta actividade permitiu que os estudantes entendessem algumas ideias essenciais de geometria e vissem o potencial dos procedimentos na construção das figuras. No decorrer das aulas e das suas actividades, observou-se que os alunos apenas se concentravam em realizar exactamente o que era pedido sem apresentar qualquer curiosidade para explorar um pouco mais da linguagem de programação *Logo*. As perguntas que iam sendo realizadas eram extremamente básicas, tendo-se verificado que os estudantes memorizavam os comandos e procedimentos em *Logo* em vez de compreenderem a própria linguagem de programação.

O passo seguinte consistiu na introdução de estruturas de repetição. Para tal, foram realizadas algumas lições recorrendo a exemplos do comando “*repeat*” e foi solicitado aos alunos que o utilizassem na criação de formas geométricas. Após a explicação observou-se que os alunos apresentavam grandes dificuldades de compreensão e utilização do comando. Decidiu-se modificar o funcionamento das aulas criando um maior número de actividades práticas. Ao longo das aulas e após uma série de exercícios, os alunos aprenderam o conceito do comando “*repeat*” e foram observadas melhorias no seu desempenho.

A actividade subsequentemente desenvolvida pelos alunos foi a criação de uma casa usando procedimentos, estruturas de repetição, e outros comandos aprendi-

dos ao longo das aulas. Os conceitos matemáticos revelaram ser um dos grandes obstáculos sentidos pelos alunos ao longo das diversas actividades. Enquanto desenhavam a casa, os estudantes sentiram grandes dificuldades na colocação do telhado (figura geométrica triangular) na posição correcta relativamente ao resto da casa. Para ajudar os alunos a ultrapassarem este problema foi explicado o teorema de Pitágoras. Com esta actividade pretendeu-se observar a capacidade e a destreza dos alunos na utilização dos conceitos aprendidos na sala de aula, tendo-se concluído que estes apenas utilizaram comandos simples e nunca recorreram ao comando “repeat” (ver Figura 5.2). No final do projecto os alunos utilizaram uma ferramenta do *MicroWorlds* para preencher com cores as formas geométricas da casa. Foi perguntado aos alunos o porquê de não utilizarem o comando “repeat” na construção da casa e a resposta foi: “O professor não disse que era preciso usar o comando “repeat” nesta actividade”.

5.5 Ambiente Construcionista

Na nossa sociedade, as escolas concentram-se mais na pedagogia tradicional em detrimento da utilização de diferentes estratégias de aprendizagem. De forma a contrariar esta situação, decidiu-se desenvolver aulas centradas numa pedagogia de aprendizagem empírica através da experimentação. O único material disponibilizado na sala de aula aos alunos foram computadores com o *software MicroWorlds* e no início das aulas o único apoio fornecido foi a explicação do sistema de ajuda integrado no sistema *MicroWorlds*. De forma a motivar os estudantes foram revelados alguns trabalhos realizados por outros alunos e foram explicadas as potencialidades da programação em *Logo*. Nas duas primeiras aulas explicou-se o funcionamento do interpretador *MicroWorlds* e demonstrou-se alguns exemplos para construir formas geométricas. No decorrer das aulas foi possível verificar que os estudantes manipulavam e aprendiam diversos vocabulários de *Logo* de uma forma mais rápida que a turma O, abrangidos pela abordagem educativa tradicional, sendo surpreendente que estes até utilizaram vocabulário mais avançado tal como: *setshape*, *touching?*,

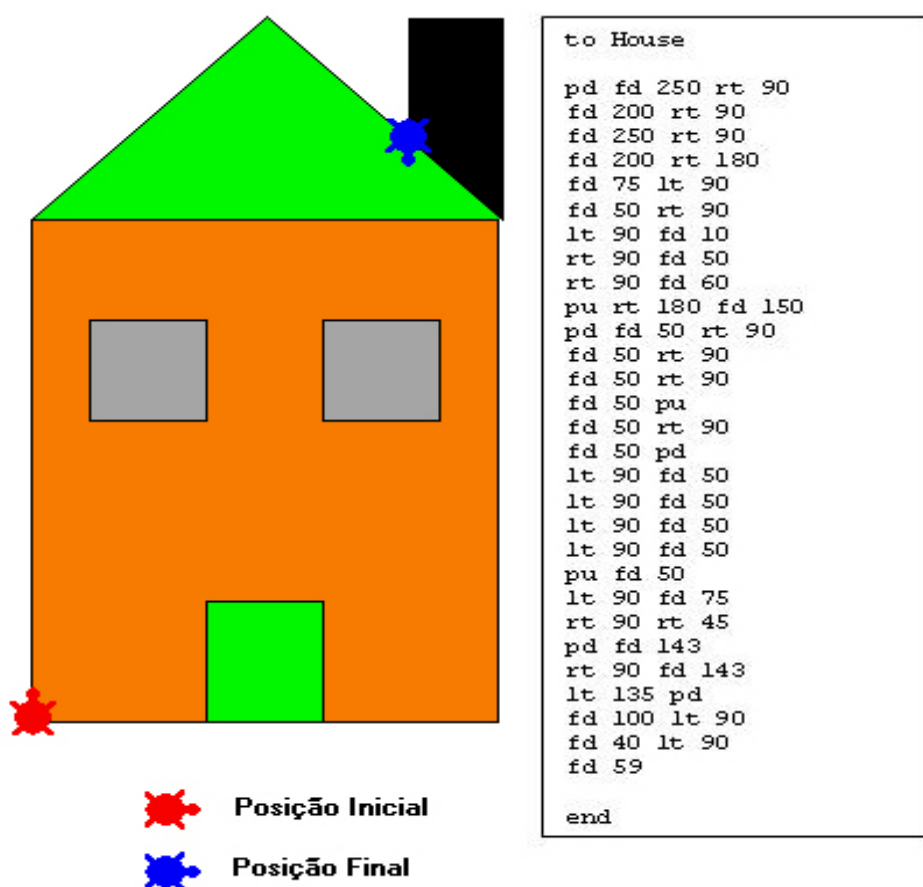


Figura 5.2: Casa implementada em Logo pelos estudantes.

forever, waituntil e o *repeat*.

Os estudantes aprenderam o movimento da tartaruga e a interface do *MicroWorlds* rapidamente e foram desafiados a desenvolverem formas geométricas com o mínimo de comandos de programação. Grande parte dos alunos utilizou o comando “*repeat*” para realizar as formas geométricas. O desafio seguinte centrou-se na construção da casa, sendo que alguns estudantes não quiseram realizar esta actividade, uma vez que a consideraram como sendo demasiado trivial, tendo preferido desenvolver animações em *Logo*. Enquanto alguns estudantes realizaram o desenho da casa utilizando comandos simples e vocabulário em *Logo* mais avançado, outros desenvolveram animações com cavalos, bicicletas a motor e plantas (ver Figura 5.3 e 5.4).



Figura 5.3: Animações implementados pelos alunos com cavalos, bicicletas a motor, e plantas.

```

to MoveHorse
  Repeat 500 [fd 5 wait 1
  setshape 22 fd 5 wait 1
  setshape 23 fd 5 wait 1
  setshape 24]
end
    
```

Figura 5.4: O procedimento do movimento do cavalo.

Durante uma aula, um aluno perguntou se seria possível desenvolver um jogo utilizando a programação em *Logo*. A nossa resposta foi afirmativa e toda a turma começou a trocar ideias, tendo os alunos decidido desenvolver o jogo do labirinto com uma tartaruga no interior. A ideia dos estudantes foi a de criar 3 níveis: um fácil, um médio e um difícil. No interior do labirinto algumas regras foram criadas e a tartaruga nunca poderia colidir contra as paredes do labirinto. No caso de isto acontecer a tartaruga recomençaria o labirinto. Quando uma tartaruga conseguisse chegar ao fim do labirinto esta seria transportada automaticamente para outro nível. Para movimentar a tartaruga seria necessário utilizar botões pré-programados. Quando a tartaruga conseguisse terminar o último nível, uma imagem deveria surgir para finalizar o jogo. Ao longo das aulas os alunos revelaram uma grande motivação e um grande empenho no desenvolvimento deste projecto. A figura 5.5 apresenta dois labirintos desenvolvidos pelos estudantes.

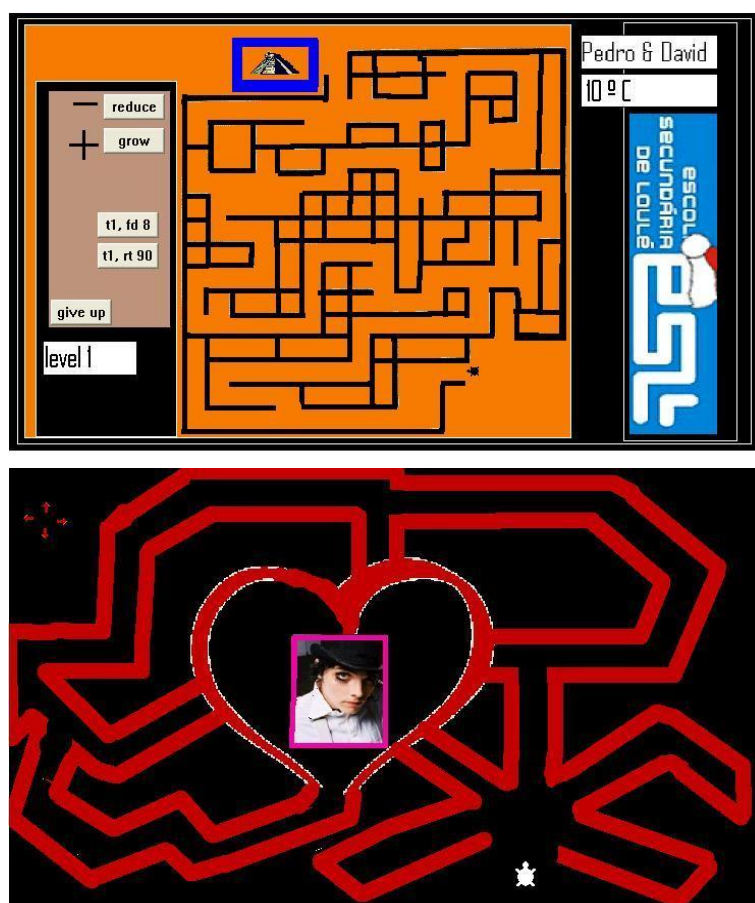


Figura 5.5: O jogo do labirinto implementado pelos alunos.

Os principais procedimentos realizados na criação do labirinto foram as paredes (*Walls*) e a transição para um nível seguinte. Alguns grupos criaram procedimentos diferentes para a mesma acção (ver Figura 5.6). Um grupo utilizou o comando “*touching?*” para mudar de nível enquanto outros grupos utilizaram o comando “*colorunder*”. Ambos os grupos utilizaram o comando “*colorunder*” para não permitir à tartaruga atravessar as paredes do labirinto.

<pre> To Walls Turtle1, forever [if colorunder = 15 [setpos[-23 -104]]] end </pre>	<pre> To ChangeLevels t1, forever [if touching? "t1 "End1 [setpos[-340 84] level2]] end </pre>	<pre> To Levels turt1, forever [if colorunder = 105 [setpos[-340 184] level2]] end </pre>
---	--	---

Figura 5.6: Procedimentos desenvolvidos pelos estudantes no jogo do labirinto.

O objectivo dos estudantes foi o desenvolvimento de um labirinto complexo, que posteriormente trocaram entre os grupos para verificar quem teria capacidade de alcançar o fim do jogo. No final da criação do labirinto os estudantes desenvolveram um *template* em *HTML* para poderem jogar o jogo na Internet. Todo este trabalho foi desenvolvido somente pelos alunos, com apenas algumas sugestões dos professores.

5.6 Feedback dos Alunos

No início das aulas, os estudantes da turma C tiveram algumas dificuldades em compreender a metodologia utilizada na sala de aula e várias questões foram colocadas, tais como: “*Mas como é que vamos aprender a linguagem de programação em Logo por nós próprios e sem ajuda?*”, “*Isto é como um grande projecto!*”. Todas as perguntas apresentavam um carácter tradicional. Porém, na segunda aula, o positivismo aumentou na turma. Os estudantes interagiram uns com os outros e construíram um ambiente mais positivo na sala de aula, sendo que, enquanto alguns estudantes discutiam o movimento da tartaruga, outros conversavam acerca da *in-*

terface do *Microworlds*. No decorrer das aulas a motivação dos alunos foi crescendo, tornando-se eles próprios instrutores na sala de aula, incentivando e ajudando os colegas. O feedback dos alunos no ambiente construcionista foi bastante positivo e estes entenderam o conceito de “aprender fazendo” (*Learning by doing*). Com o passar do tempo verificou-se uma mudança de mentalidades na sala de aula. O comportamento da turma O contrasta fortemente com o observado na turma C. Os estudantes fizeram apenas algumas questões e a interacção da turma foi bastante limitada. As actividades apenas foram centralizadas em efectuar aquilo que era pedido e revelaram pouca motivação e entusiasmo.

5.7 Análise e Resultados

Ambas as turmas terminaram as aulas com um exame final. Apesar da realização de um exame na turma C infringir as regras do espírito construcionista, este teve de ser realizado, uma vez que é obrigatório de acordo com regulamento interno da escola. A classificação final teve de ser dada a cada estudante de acordo com este exame e não com base numa avaliação subjectiva. O exame consistiu em três perguntas limitadas a 35 minutos e foi disponibilizado às duas turmas ao mesmo tempo. O objectivo do exame consistiu na criação de algumas formas geométricas, utilizando os vocabulários da linguagem *Logo* aprendido nas aulas, e na construção de simples animações. Durante a realização do exame, os estudantes da turma O demonstraram algumas dificuldades na programação das formas e revelaram inquietação relativamente ao tempo disponibilizado para a realização do exame, o que lhes induziu uma reacção de stress. A capacidade de programar em *Logo* apresentava um carácter essencialmente teórico e menos focalizado no processo de descoberta, uma vez que nesta turma os estudantes não foram estimulados a aprenderem sozinhos a explorar o vocabulário da programação em *Logo*. Observou-se uma reacção negativa ao longo do exame, uma vez que quando um estudante não apresentava um resultado positivo o primeiro impulso era o de desistir. Por outro lado, a turma C revelou um comportamento mais descontraído. Os estudantes revelaram uma postura mais

calma para resolver o exame e não houve qualquer reacção negativa.

O resultado das notas confirmou o esperado, tendo a turma C apresentado melhores notas. A média das notas da turma C foi de 12,4 e na turma O foi de apenas 10,1. É de salientar o facto de que não haveria necessidade de realizar um exame final a fim de concluir que a turma C tinha uma melhor experiência de aprendizagem e de desempenho. Observando as actividades realizadas nas duas turmas ao longo de 5 meses, tornou-se claro que a aprendizagem da turma C foi superior, tendo os respectivos alunos desenvolvido procedimentos de maior complexidade, utilizando para tal vocabulário mais avançado. A turma O, por sua vez, empregou vocabulário mais simples e muito repetitivo.

No final das aulas, os alunos da turma C apreciaram a programação em *Logo* e os da turma O não revelaram qualquer interesse na mesma.

5.8 A Linguagem *Logo* para Crianças com Paralisia Cerebral

Após os estudos realizados com o *Logo* no ensino secundário foi possível entender as potencialidades pedagógicas desta linguagem de programação no desenvolvimento criativo das crianças. Mas *será que uma criança com deficiências físicas também consegue programar em Logo e desenvolver projectos de forma autónoma?* Com o objectivo de responder a esta questão nasceu o laboratório *Mindstorms*. Este projecto foi realizado em parceria com a Associação Portuguesa de Paralisia Cerebral (APPC), e visa a participação de alunos com paralisia cerebral num programa de investigação em que o objectivo principal é adaptar e adequar a linguagem de programação *Logo* às necessidades e potencialidades destes alunos. As instalações foram criadas num laboratório da Universidade do Algarve e a primeira implementação de uma interface para auxiliar indivíduos com deficiências físicas na linguagem *Logo* estava prestes a ser construído (ver Figura 5.7).

O primeiro contacto com os jovens da APPC revelou-se extremamente duro



Figura 5.7: Primeiro contacto no laboratório *Mindstorms* com as crianças da APPC.

devido ao elevado grau de deficiência revelado por eles. Apesar de possuirmos o apoio de técnicos especializados da APPC durante as sessões, foi fundamental compreender quais as características específicas de cada jovem. Todos eles, de idades compreendidas entre os 12 e os 17 anos, apresentavam graves deficiências físicas e de comunicação, sendo-lhes totalmente impossível a utilização de um teclado convencional ou de um simples dispositivo de apontar, tal como um rato, para manipularem o computador e utilizarem a linguagem *Logo*.

Tendo estes fundamentos em consideração, foi desenvolvida a primeira versão de um teclado virtual (ver Figura 5.8), onde os comandos em *Logo* não são digitados através de um teclado convencional. Utilizando o sistema de varrimentos da interface os jovens apenas tinham de pressionar um botão de toque (*switch*) com a cabeça ou com o braço a fim de introduzir o comando em *Logo* no editor de texto. Na interface existem dois grupos principais de varrimentos. O primeiro é utilizado para seleccionar o teclado com os comandos em *Logo* e o segundo para seleccionar o teclado numérico. Quando um dos grupos é escolhido o varrimento é dividido em pequenos grupos de itens e estes são, posteriormente, seleccionados individualmente. A hierarquia dos subgrupos é realçada com uma determinada cor até atingir a selecção individualizada de cada item. Com este processo torna possível indivíduos com limitações físicas introduzirem informação num computador apenas pressionando.

ando uma única tecla ou um botão de toque. No entanto, esta primeira versão não se apresentou totalmente funcional uma vez que a população em estudo também apresentava dificuldades de leitura.



Figura 5.8: Primeira versão do teclado virtual para auxiliar os jovens na digitação dos comandos em *Logo*.

Foi então elaborada uma segunda versão constituída não por palavras mas por imagens de modo a facilitar a utilização da interface (ver Figura 5.9). A segunda versão revelou-se mais eficiente, os alunos começaram a experimentar a interface e a enviar alguns comandos para o editor de texto.

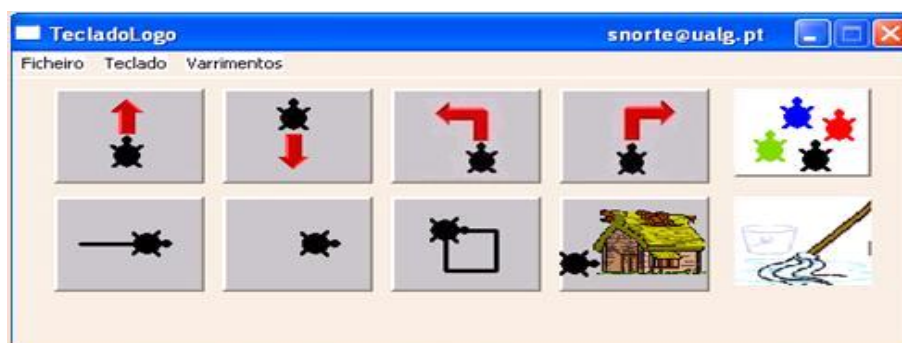


Figura 5.9: Segunda versão do teclado virtual constituída não por palavras mas por imagens.

Após algumas experiências pôde-se constatar que a interface não apresentava um carácter de fácil compreensão por parte das crianças, uma vez que para além das limitações físicas, os alunos também revelavam graves dificuldades cognitivas. Foi assim necessária uma nova alteração da interface, tendo-se eliminando algumas teclas, aumentando o tamanho das mesmas e criando alguns sons. No sistema automático de selecção dos comandos foi adicionada ainda a opção de configuração para poder

aumentar ou reduzir a velocidade de selecção dos comandos. Nascia assim a terceira versão do teclado *Logo* (ver Figura 5.10). Esta terceira versão permitia apenas aos alunos movimentarem a tartaruga para a frente, para trás, virar 90° à esquerda ou à direita.

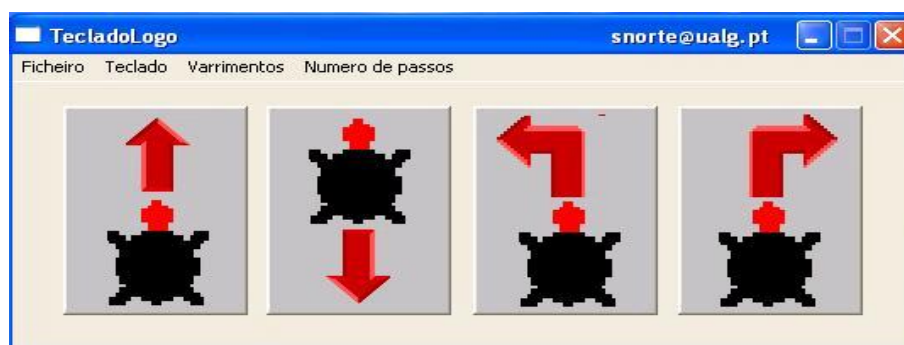


Figura 5.10: Terceira versão do teclado virtual constituída por 4 teclas de maiores dimensões.

De modo a facilitar a compreensão dos movimentos da tartaruga, foi determinado que de cada vez que um aluno realizasse uma tarefa no teclado virtual, ele próprio efectuaria o mesmo movimento da tartaruga com a sua cadeira de rodas. Este exercício com a interface permitiu auxiliar os alunos na exploração do seu próprio espaço, fortalecendo a sua orientação espacial. Apesar das melhorias, e embora esta terceira versão do teclado virtual se tivesse tornado bastante clara para um dos alunos, os restantes estudantes continuaram a apresentar algumas dificuldades de compreensão do seu manuseamento. Com o intuito de ultrapassar esta barreira, foi criada uma última versão bastante similar à anterior mas onde o objectivo principal centraliza-se apenas no ambiente virtual da tartaruga. Foram removidas da terceira versão as várias imagens da tartaruga e foi colocada apenas uma imagem central da mesma rodeada por diversas setas direccionais. Conforme a seta direccional é escolhida pelo aluno, a imagem da tartaruga modifica-se e nas costas da tartaruga surge a imagem da seta seleccionada (ver Figura 5.11). Esta última versão foi bastante benéfica, facilitando a compreensão da interface e tornando desde modo possível o movimento da tartaruga no espaço virtual.

Os alunos conseguiram controlar os movimentos da tartaruga mas necessitaram

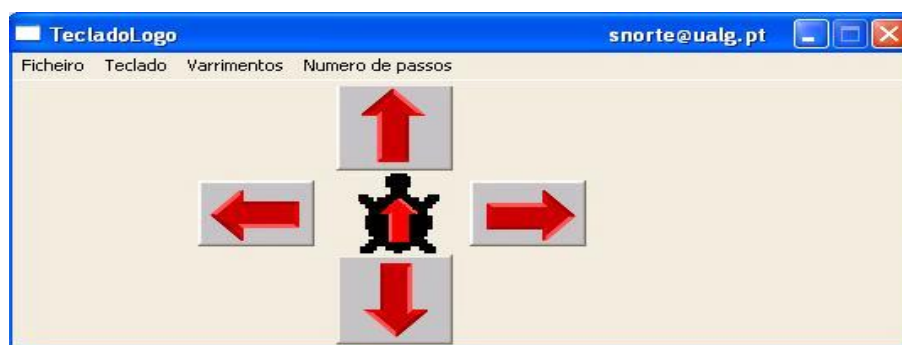


Figura 5.11: Quarta versão do teclado virtual constituída por uma imagem central da tartaruga e rodeada por 4 setas direccionais.

quase sempre da nossa ajuda.

A população em estudo revelou graves dificuldades cognitivas o que dificultou o nosso desempenho nesse projecto. Os próprios auxiliares do apoio educativo da APPC confirmaram que esta população apresentava características complicadas para este projecto devido ao elevado nível de deficiência cognitiva. No entanto, este projecto revelou-se extremamente importante para a compreensão das dificuldades motoras que um individuo apresenta na interacção com a interface, e surgiu como o impulsionador para a construção de uma nova interface de programação em *Logo* no meio educacional.

5.9 Sumário

Neste capítulo foram apresentados os resultados de um estudo efectuado para comparação da utilização de dois paradigmas pedagógicos, objectivismo e construcionismo, no ensino de linguagens de programação em duas turmas do ensino secundário. Na turma onde a abordagem utilizada foi o objectivismo, a linguagem de programação *Logo* foi apresentada através de pequenos vídeos e apresentações em *PowerPoint*, explicando aos estudantes todas as funcionalidades e ferramentas do *software*. Na turma onde a vertente educacional utilizada foi o construcionismo, o único material disponibilizado na sala de aula foram computadores com o *software MicroWorlds* e no início das aulas a única ajuda fornecida foi a explicação do sis-

tema de ajuda integrado no *software*. Os alunos foram assim obrigados a explorar os conceitos de programação.

Os alunos das duas turmas evoluíram de forma distinta, construindo saberes e conhecimentos diferentes. A turma focada numa abordagem tradicional, mais objectiva, revelou competências e desempenho muito inferiores ao da turma centralizada no paradigma construcionista. Com este estudo também foi possível observar as potencialidades que a programação em *Logo* pode proporcionar no meio educacional.

Este capítulo fica ainda marcado pelo importante projecto, desenvolvido em parceria com a Associação Portuguesa de Paralisia Cerebral (APPC), onde se desenvolveu uma interface que possibilita às crianças da instituição utilizarem comandos básicos da linguagem de programação *Logo*. Apesar de este estudo ter revelado algumas dificuldades de concretização devido às graves limitações cognitivas dos alunos, foi possível interiorizar quais os principais problemas sentidos por este tipo de população. Os alunos conseguiram satisfatoriamente controlar e movimentar a tartaruga mas sempre acompanhados pela presença de um auxiliar.

6

LogoKeyboard

6.1 Introdução

A história do Homem está marcada pela progressão dos media utilizados para transmitir experiências e ideias. A História começa com pessoas a pintarem paredes das cavernas para expressarem acontecimentos de caçadas ou até mesmo para partilharem crónicas da história da comunidade ou tribo. Foi a importância desta comunicação que elevou a posição de narrador de histórias a uma posição de elevada estima na sociedade. Estas pinturas rupestres foram um meio primitivo de comunicar ideias, factos úteis e eventos através das pessoas. Ao longo dos anos, as pessoas têm explorado formas para utilizarem cada novo meio de comunicação que melhor expresse as suas ideias. Talvez o mais recente passo na progressão das tecnologias da comunicação seja o desenvolvimento de interfaces (Sharp *et al.*, 2007). Estas estão a ser desenvolvidas para utilização em várias áreas, como sejam animação e multimédia, diagnóstico médico e terapia, para estudos de património e arqueológicos, marketing, educação, entre outros. De acordo com Zhai e colaboradores, as ferramentas virtuais podem facilitar a manipulação de dispositivos computacionais, mas antes de um utilizador manipular um dispositivo virtual, é necessário que este utilize primeiramente um dispositivo físico. Diversos dispositivos podem ser utilizados, nomeadamente, botões de toque mais conhecidos por *switches*, dispositivos ou

ponteiros bocais, *joysticks*, teclados físicos, ratos, *trackballs*, microfones e *webcams* (Zhai *et al.*, 2000).

Em 2002, alguns investigadores realizaram experiências utilizando um teclado virtual incorporado num *palmtop*, ou *PDA*, com indivíduos que apresentavam problemas a nível motor. Através de uma caneta o utilizador interagia com o *palmtop*, manipulando um *touchpad* ou teclado virtual (Myers *et al.*, 2002). Ao longo de diversos estudos realizados com indivíduos portadores de problemas motores, Condado e colaboradores verificaram que este tipo de dispositivos demonstram uma grande lacuna quando utilizados por indivíduos portadores de graves limitações físicas, especialmente em indivíduos portadores de paralisia cerebral (Condado *et al.*, 2004). Assim sendo, os dispositivos físicos devem ser adequados de modo a facilitar o manuseamento de interfaces tais como interagir com um teclado virtual. Um teclado virtual não é mais do que um utilitário de acessibilidade que exhibe um teclado no ecrã do computador, permitindo que as pessoas com problemas de movimentos digitem os dados utilizando um dispositivo ou *joystick* de apontar e esta tecnologia deve proporcionar ao utilizador um ambiente agradável, composto por uma interface bem construída, de modo a facilitar a sua utilização. Para desenhar uma interface com boa qualidade e acessibilidade é essencial compreender as capacidades e comportamentos dos utilizadores (Nielsen, 1995).

Existe uma série de teclados virtuais destinada a utilizadores que estão, por qualquer motivo, impossibilitados de utilizar um teclado físico. Os teclados virtuais podem ser manuseados com um simples rato ou através de dispositivos alternativos de entrada de dados que simulam acções do rato. Diversos teclados virtuais já foram testados com indivíduos portadores de limitações físicas (Darragh & Witten, 1992; Edwards, 1995; Jones, 1998; Trewin, 2004; Wilson & Agrawala, 2006; Norte *et al.*, 2005; Norte & Lobo, 2007). A própria *Microsoft* disponibiliza um teclado virtual integrado no sistema operativo *Windows*, mas este apresenta algumas limitações, uma vez que não permite aumentar o tamanho das teclas ou modificar as cores no teclado e os sons do sistema estão limitados. Este teclado proporciona um nível

mínimo de funcionalidade para utilizadores com deficiências físicas. Diversos outros teclados virtuais com características especiais têm sido implementados para auxiliar indivíduos com limitações físicas, tais como, WiViK e HandsOFF! (Shein *et al.*, 1991).

6.2 Metodologias de Desenho da Interface

A ideia de se desenvolver uma plataforma que permita a integração de indivíduos com deficiências motoras em ambientes de programação e que lhes possibilite o desenvolvimento dos seus próprios programas, surgiu durante a construção do *software* ADM.

Alguns estudos já demonstraram que pessoas com determinadas limitações físicas ou cognitivas também podem desenvolver programas informáticos (Valente, 1983). No entanto, muitas crianças e adolescentes continuam sem conseguir desempenhar muitas actividades, pelo facto de não possuírem os dispositivos de acessibilidade adequados. Através do desenvolvimento da interface denominada *LogoKeyboard*, tornamos possível a utilização da linguagem *Logo* por parte de utilizadores com deficiências motoras (Norte & Lobo, 2007).

O *LogoKeyboard* pode ser utilizado com um interpretador de *Logo*, como sendo o *MicroWorlds*, o *UCBLogo*, ou o *MSWLogo*. A interacção do utilizador com este teclado virtual é iniciada através da utilização de um dispositivo físico tal como um rato, teclado ou *switch* que permitirá a selecção dos comandos. A mensagem é enviada para o teclado virtual *LogoKeyboard* e o comando *Logo* escolhido é enviado para a linha de comandos — o *output stream* onde é decifrado pelo interpretador de *Logo*.

No decorrer do processo de desenvolvimento do *LogoKeyboard* procurou-se encontrar e eliminar qualquer erro de concepção do *software* que dificultasse a sua utilização, recorrendo para tal a experiências, observações e testes de usabilidade. A realização de testes de usabilidade é um factor crucial no aperfeiçoamento de

qualquer tipo de interface. É considerada uma boa interface aquela que revela simplicidade e facilidade de utilização. Na área das acessibilidades, na qual se enquadra a plataforma *LogoKeyboard*, a realização de experiências, observações e testes de usabilidade torna-se ainda mais preponderante devido às dificuldades de interacção apresentadas pelos utilizadores com deficiências físicas.

A interface do *software LogoKeyboard* foi desenvolvida na linguagem de programação C++, recorrendo à biblioteca multi-plataforma *wxWidgets* que forneceu os elementos básicos para a sua construção. Desta forma os componentes principais do *LogoKeyboard* podem ser facilmente portado para outros sistemas operativos.

A sua construção não foi trivial, tendo sido necessária a realização de diversas iterações para se alcançar o produto final adequado. As primeiras iterações da interface surgiram com o projecto “*MindStorms*” com as crianças da APPC. Inicialmente o teclado era constituído por imagens que permitiam apenas o controlo da tartaruga. Com este projecto verificou-se que se tornava bastante complexo e confuso desenvolver uma interface constituída apenas por imagens devido à grande quantidade de vocabulário da linguagem *Logo*. Surgiu assim a necessidade de incorporar os comandos do *Logo*, de uma forma simples e atractiva, no próprio desenho da interface. Para atingir este objectivo, foi fundamental primeiramente entender o mecanismo de interacção do utilizador com a linguagem *Logo*, tendo sido realizados, ao longo de vários meses, vários estudos com alunos a interagir com a programação *Logo* (ver capítulo 5). Esta análise permitiu-me compreender, com base em observações qualitativas, que os vocabulários do *Logo* deveriam ser organizados por grupos comuns. Por exemplo, vocabulário relacionado com o movimento da tartaruga deveria ser agrupado num só conjunto e não misturado com vocabulário de operações matemáticas. Desta forma surgiu a ideia de construir diversos teclados constituídos por teclas com vocabulário da linguagem *Logo*.

6.3 Organização do Teclado

Quando se trabalha com indivíduos que revelam deficiências motoras graves torna-se fundamental exigir o mínimo de esforço possível na execução de uma determinada tarefa. Tendo este critério em consideração, o teclado foi desenvolvido com uma estrutura que visa facilitar a utilização das teclas durante a construção de programas em *Logo*. Para poder organizar os vocabulários do *Logo* foi necessário colocar cada vocabulário num determinado grupo, permitindo, desta forma, a criação de um teclado extremamente organizado (ver Figura 6.1 e 6.2). No total existem seis grupos de vocabulário: “*Graphics*”; “*Screen Management e Text Editing*”; “*Words/Lists e Disk Access*”; “*Object e Control/Logic*”; “*Math e Assigning*”; e “*Input/Output, Time, Sound, Variables*”. Depois de organizar por grupos os diversos vocabulários, foi necessário estudar qual a ordem das teclas a aplicar. O critério utilizado para a organização sequencial das teclas em cada teclado focou-se nos comandos mais utilizados pelos alunos no desenvolvimento dos projectos em *Logo*. Outro aspecto tido em consideração na organização do teclado foi minimizar o tempo de espera na escolha dos comandos quando os utilizadores com deficiências motoras utilizam o sistema de varrimento. Para um utilizador que necessite de utilizar o sistema de varrimento torna-se mais cómodo conter no mesmo teclado vocabulário da mesma família. Por exemplo, se este pretender utilizar vocabulário matemático, será mais vantajoso se as teclas relativas a este vocabulário estiverem todas localizadas no mesmo teclado. De outro modo, tornar-se-á mais cansativo para o utilizador uma vez que este terá de procurar em que teclado se encontra o comando desejado. Globalmente o teclado apresenta 312 teclas e pode ser descarregado gratuitamente no endereço: <http://w3.ualg.pt/~snorte/LogoKeyboard.htm>. Além dos comandos de programação em *Logo*, há ainda a possibilidade de se utilizar um teclado alfanumérico (ver Figura 6.3).

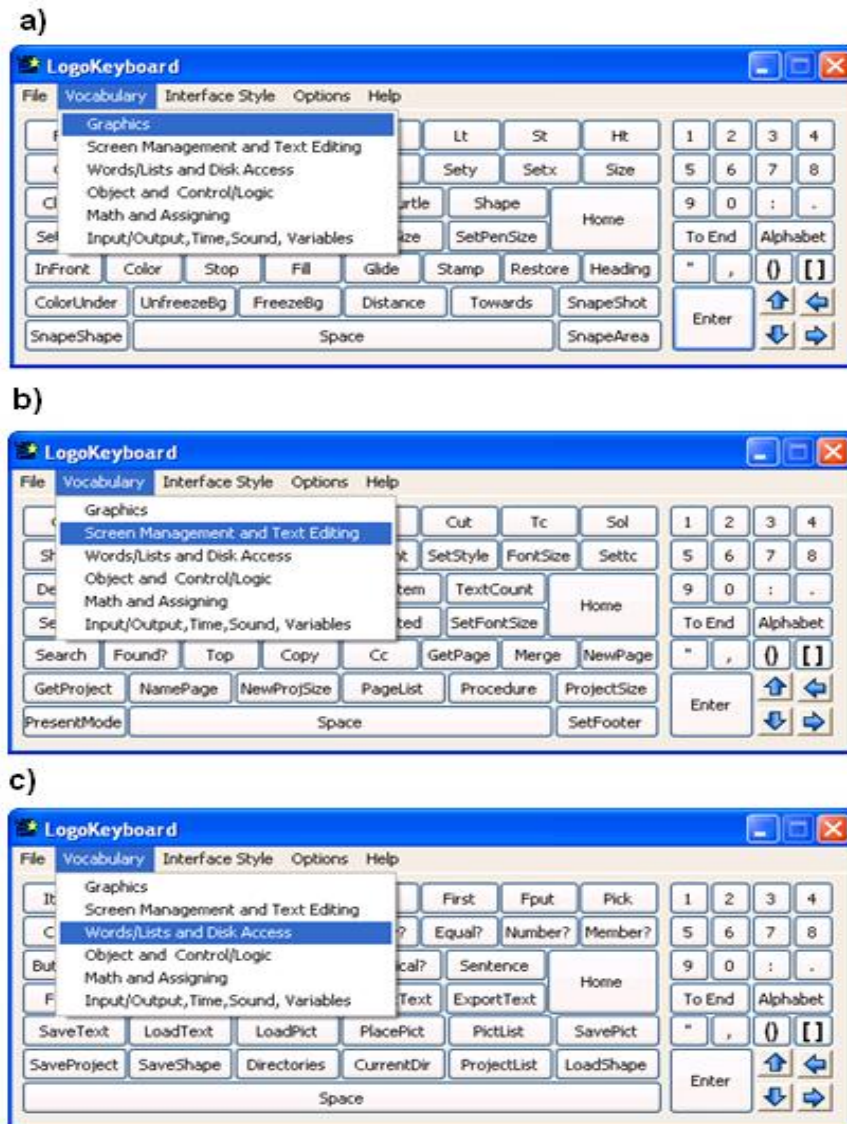


Figura 6.1: a) O primeiro grupo de teclados é composto por vocabulários relacionados com o ambiente gráfico, b) O segundo com a gestão da tela e edição de texto, c) O terceiro com o manuseamento de listas/palavras e acesso ao disco.

6.4 Ferramentas da Interface

O *LogoKeyboard* possui um sistema que permite ao utilizador modificar a transparência da interface dependendo da necessidade. O menu de opções contém ferramentas úteis, incluindo um sistema integrado de ajuda que disponibiliza informação de cada comando através de uma breve descrição e de um exemplo animado para a sua utilização (ver Figura 6.4).

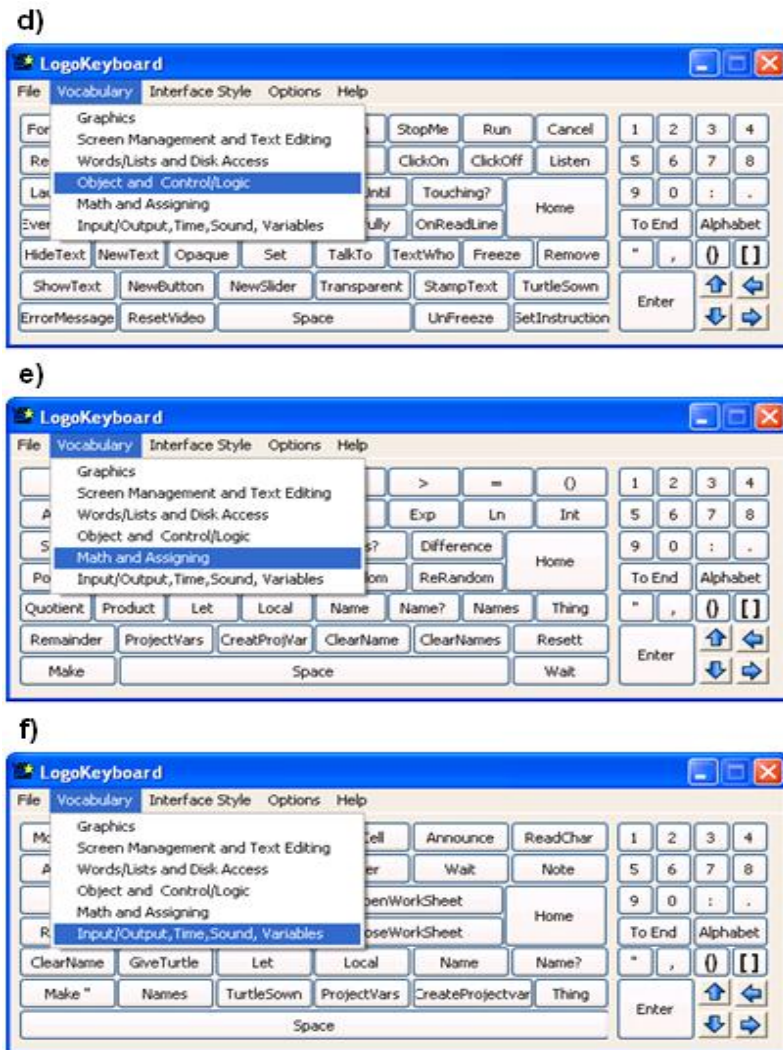


Figura 6.2: d) O quarto grupo de teclados é composto por vocabulários relacionados com a utilização de objectos, controle e lógica, e) O quinto grupo com operações matemáticas e funções, f) O último grupo é constituído por comandos de entrada e saída de dados, tempos, sons e variáveis.

A opção Zoom permite o aumento do tamanho das teclas, sendo extremamente útil para pessoas que apresentem problemas visuais. Esta opção pode ser utilizada colocando o cursor do rato no topo da tecla desejada ou automaticamente através de um sistema de varrimento (ver Figura 6.5).

O utilizador pode ainda configurar diversas opções no teclado, tais como: a velocidade do sistema de selecção automática, o número de repetições do ciclo do sistema de selecção, o som, a cor e o tamanho do teclado. O utilizador tem ainda

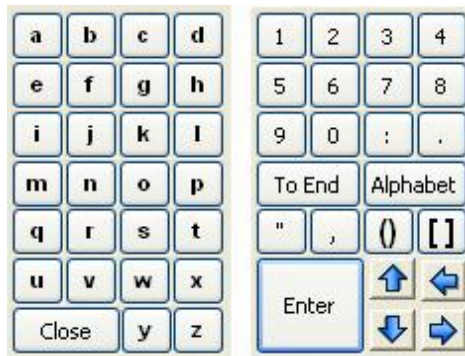


Figura 6.3: Constituição do teclado alfanumérico do *LogoKeyboard*.

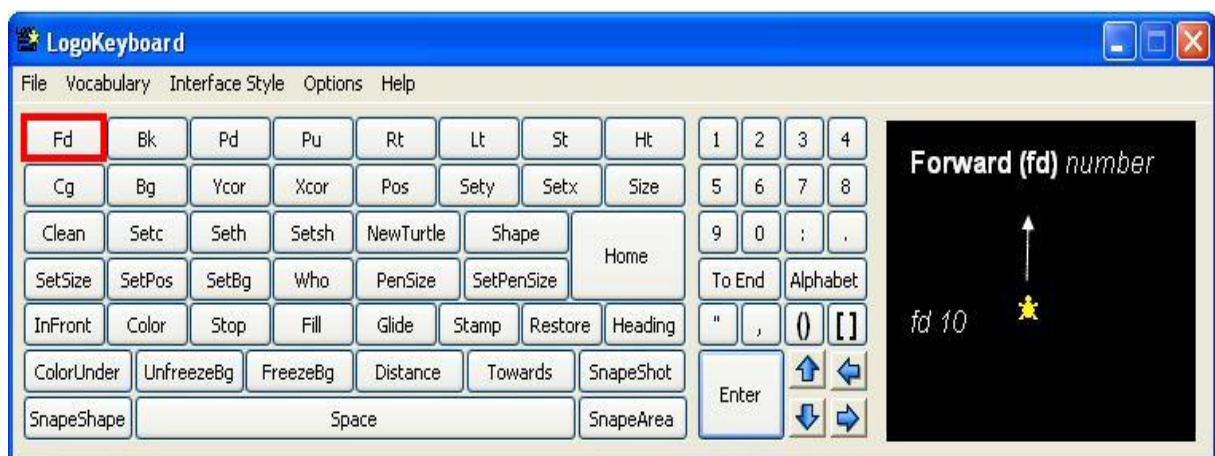


Figura 6.4: O utilizador visualiza o sistema de ajuda do *LogoKeyboard* apenas colocando o cursor do rato na tecla desejada. Neste exemplo o comando “Fd” foi seleccionado e à direita é gerado um ambiente para visualizar uma pequena animação referente ao comando desejado.

a possibilidade de identificar os comandos em *Logo* através de um sistema de voz quando o rato se movimenta por cima de cada tecla. Um aspecto que foi cuidadosamente pensado foi a possibilidade de desenvolver uma interface configurável e de fácil acesso. Uma pessoa com limitações físicas pode pressionar um simples botão para configurar e manipular a interface sem qualquer ajuda de um assistente. A utilização de métodos de varrimentos, para simular o funcionamento de um dispositivo de *input* como um rato, vai permitir que um utilizador possa construir um ambiente construcionista com a linguagem *Logo*.



Figura 6.5: O sistema para aumentar o tamanho das teclas no teclado virtual pode ser manipulado de duas formas: a) colocando o cursor do rato na tecla desejada ou b) automaticamente através do sistema de varrimento.

6.5 Métodos de *Input*

Para indivíduos com limitações físicas, os métodos utilizados para a introdução de dados numa aplicação são fundamentais. Várias técnicas têm sido implementadas utilizando *joysticks*, teclados virtuais (Wilson & Agrawala, 2006), *trackballs*, métodos gestuais (Wobbrock & Myers, 2006) e sistemas de reconhecimento de voz para controlar o cursor do rato (Dai *et al.*, 2004; Mihara *et al.*, 2005; Arnold *et al.*, 2000).

Os sistemas de varrimento não são uma novidade na área das acessibilidades e têm sido utilizados, principalmente, em interfaces para manipulação de teclados convencionais ou para substituir o uso do rato na realização de operações no ambiente

de trabalho de um computador (Steriadis & Constantinou, 2003; Lin *et al.*, 2008; Ghedira *et al.*, 2009; MacKenzie, 2009). No *LogoKeyboard* os sistemas de varrimento tornaram-se fundamentais para manipular teclados que apresentam características únicas essenciais para programar na linguagem de programação *Logo*. Além da utilização dos sistemas de varrimento para seleccionar comandos no teclado virtual, desenvolveu-se ainda a possibilidade de um utilizador poder empregar os sistemas de varrimento para configurar sozinho a interface, de modo a adequar-se às suas necessidades. Pessoas com capacidade para utilizar um rato podem facilmente usar o *LogoKeyboard* para o desenvolvimento de programas em *Logo*, pressionando apenas o botão do rato na tecla desejada. O *LogoKeyboard* é suposto ser utilizado conjuntamente com um interpretador de *Logo*, mas também pode ser utilizado apenas para escrever texto com o teclado alfanumérico. Em ambos os casos é necessário utilizar um editor de texto pertencente a um interpretador de *Logo* ou a um editor de texto regular, que fique activo à espera de receber informação do teclado virtual. Pessoas que não podem utilizar um rato convenientemente podem usufruir de um sistema de selecção automático designado por varrimento, que se encontra disponível no *LogoKeyboard*. Aliás, uma das técnicas mais utilizadas para auxiliar pessoas com problemas de locomoção a interagir com teclados virtuais é o método de varrimento ou selecção automática (Kate *et al.*, 1979; Frietman *et al.*, 1984; Shein, 1997; Kahn *et al.*, 1999; Majaranta & Rähkä, 2002). A técnica de varrimento permite ao utilizador usar apenas uma simples tecla, ou um *switch* para efectuar escolhas num teclado virtual. A ideia principal consiste em alterar a área de selecção (que se encontra evidenciada com uma determinada cor) de um item para outro, após completado um determinado tempo predefinido. A única operação que o utilizador necessita de realizar é pressionar apenas uma tecla, ou um *switch*, quando o item desejado se encontrar seleccionado (evidenciado com uma determinada cor).

Existem dois grupos principais de varrimentos no *LogoKeyboard*: o primeiro é utilizado para seleccionar teclas no teclado de programação em *Logo* e o segundo é usado para seleccionar itens no teclado alfanumérico (ver Figura 6.6). Após o

utilizador ter escolhido um dos grupos, a área seleccionada é reconhecida e o sistema de varrimento procede para um sistema de subgrupos derivados do grupo original. Após a selecção do subgrupo, o sistema de varrimento movimenta-se linha por linha até o utilizador escolher uma delas. Finalmente, o sistema focaliza uma tecla de cada vez. Nesta fase, a próxima escolha do utilizador será para eleger a tecla pretendida. Para utilizar o sistema de varrimento, o utilizador necessita de pressionar a tecla “Espaço” ou utilizar um *switch* ou um simples rato.

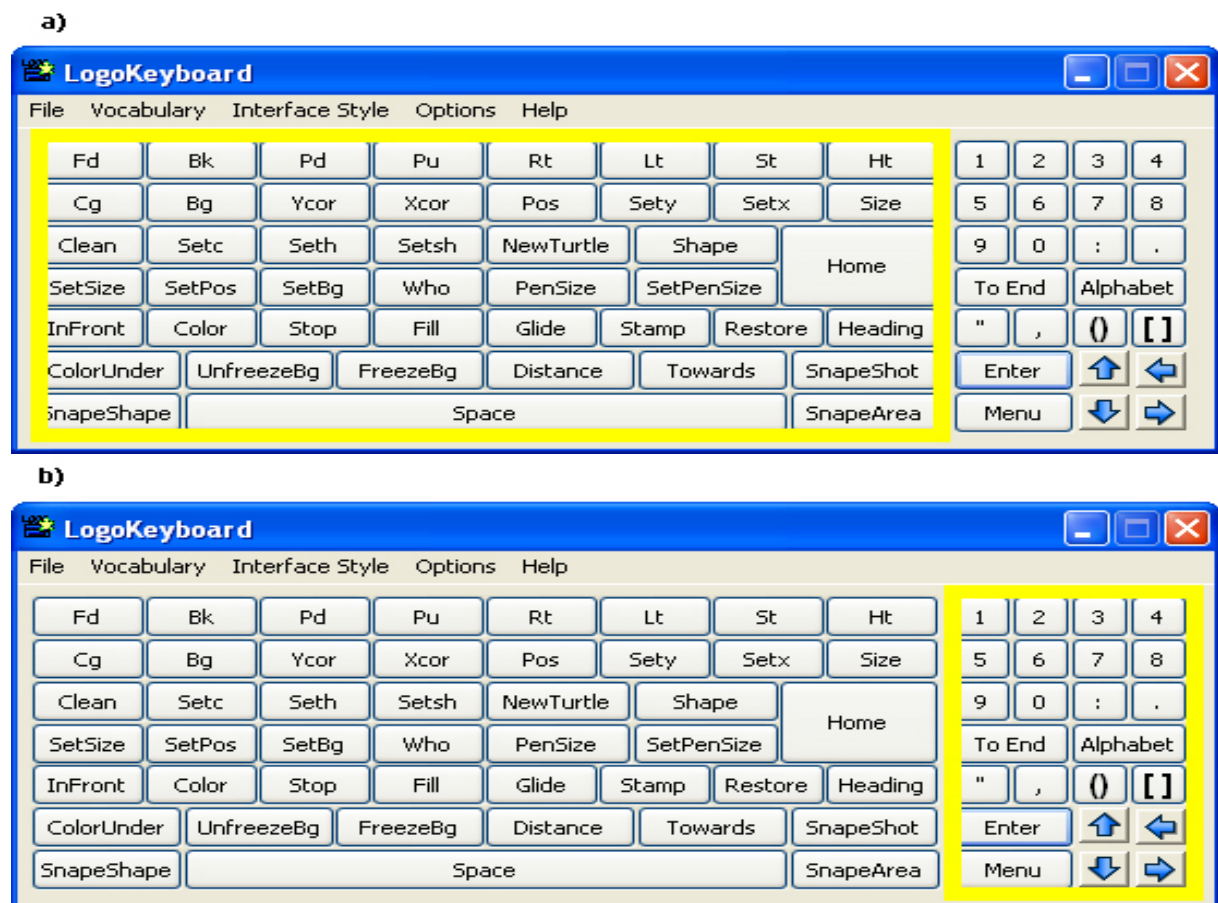


Figura 6.6: a) Selecção do teclado de programação composto pelos comandos de programação em *Logo*; b) Selecção do teclado alfanumérico.

Assim, se por exemplo o utilizador, com deficiência física, necessitar de escrever o comando em *Logo* “SetPenSize”, ele poderá fazê-lo utilizando o sistema de selecção automática. O primeiro passo a realizar será pressionar a *switch* no instante em que a área dos comandos em *Logo* desejada encontrar-se evidenciada com uma deter-

minada cor. A partir deste momento começará o varrimento por linhas do teclado. Quando a linha pretendida, onde se encontra o comando “SetPenSize”, aparecer evidenciada apenas é necessário pressionar novamente o *switch*. Finalmente é gerado um novo sistema de varrimento nessa linha, que focaliza comando a comando com uma determinada cor. Quando o comando “SetPenSize” estiver evidenciado, o utilizador pode então pressionar o *switch*, processo este que enviá-lo-á para o editor de texto pertencente a um interpretador de *Logo* (ver Figura 6.7).

Este processo torna bastante eficiente a manipulação dos comandos em *Logo*, sem que para tal seja necessária a utilização de qualquer teclado convencional. O utilizador poderá utilizar o mesmo método descrito anteriormente para manipular o teclado alfanumérico, ou então seleccionar opções de configuração mais adequadas para as suas necessidades.

6.6 Motivação e Aprendizagem

A experiência com a filosofia *Logo* utilizando computadores em crianças com limitações físicas pode oferecer uma nova forma de educar (Goldenberg, 1979). O computador pode ser uma ferramenta através da qual as crianças desenvolvem um raciocínio abstracto, expressando ideias previamente inacessíveis, e ser um elemento activo e criativo, uma vez que a programação *Logo* possui um grande leque de características que ajuda a desenvolver um ambiente ideal de aprendizagem para crianças com problemas motores (Valente, 1983; Perlman, 1974). Isto porque o material que as crianças precisam de manipular num ambiente em *Logo* não se apresenta como um objecto físico, que requer uma elevada coordenação motora, mas sim objectos controlados pelo computador. O *LogoKeyboard*, como instrumento que ajuda a minimizar a barreira entre a criança e o mundo físico, pode ser utilizado por crianças que apresentem uma coordenação motora suficiente para controlar um único botão para realizarem todas as tarefas que necessitam, sem carecer da ajuda de terceiros. As actividades que a criança desenvolve num ambiente de programação em *Logo* podem ser determinadas pelo seu próprio interesse e imaginação. Isto é muito importante

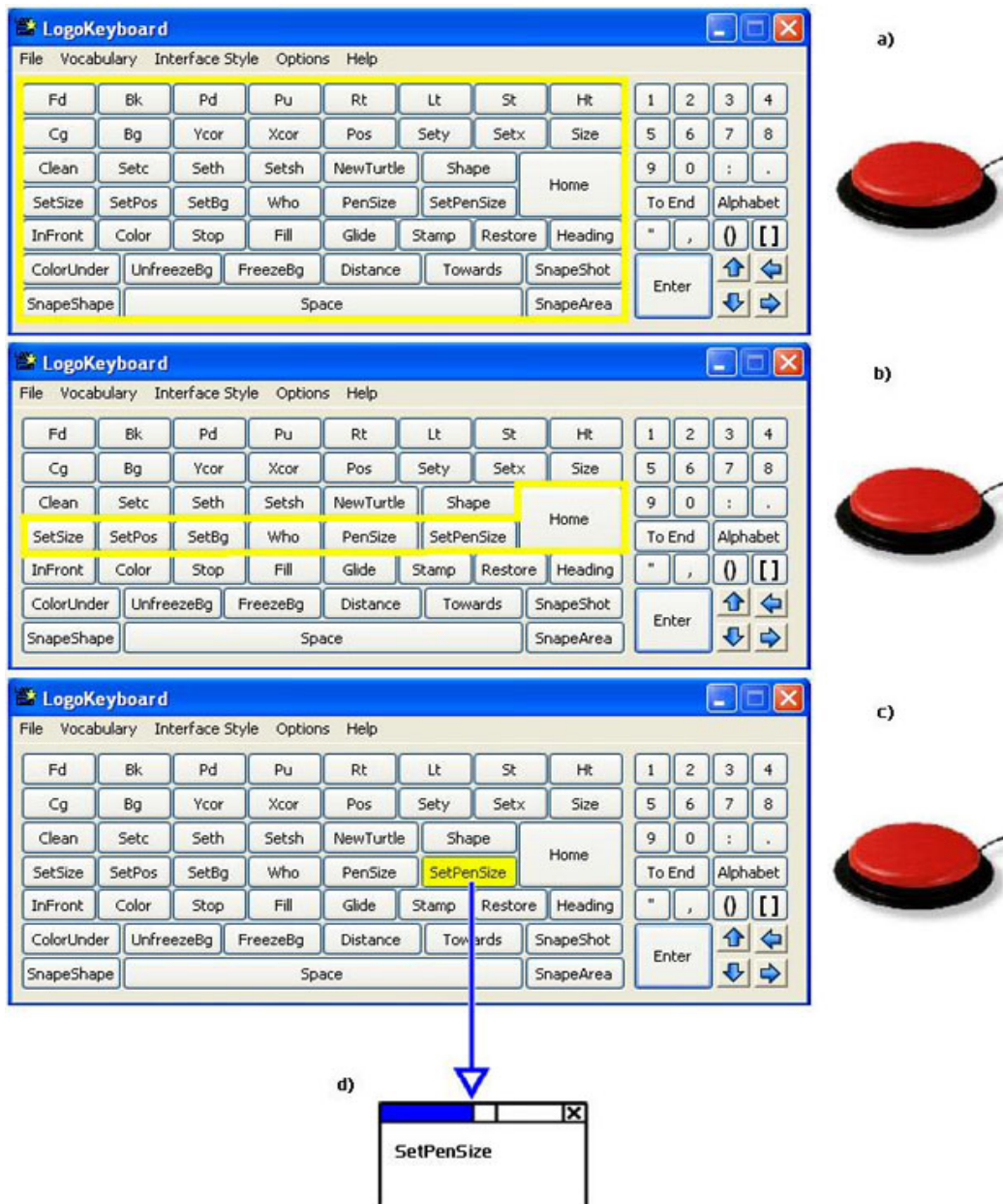


Figura 6.7: a) O utilizador ao pressionar o *switch* quando a área do teclado de interesse se encontra evidenciada com uma determinada cor no teclado desejado permite desta forma efectuar escolhas no *LogoKeyboard*. b) Após ter pressionado o botão de toque outro sistema de varrimento selecciona automaticamente cada grupo de teclas na horizontal. Neste exemplo o utilizador pressionou o botão de toque na linha onde se encontra o comando “SetPenSize”. c) Finalmente é activado outro sistema de selecção que focaliza com uma cor o interior de cada comando. Quando o utilizador visualiza a cor no comando “SetPenSize”, pressiona o botão de toque. d) Desta forma o “SetPenSize” é enviado para o editor de texto do interpretador de *Logo* executando a sua tarefa.

porque permite que o teclado transforme a passividade da criança portadora de deficiência. A perspectiva da educação de acordo com a filosofia *Logo* consiste em oferecer às crianças o poder para alcançar o seu potencial e esquecer as suas deficiências. Tal como é defendido por Seymour Papert, as crianças devem ser providas de um ambiente ideal para serem construtores do seu próprio conhecimento e aprenderem com os seus próprios erros (Papert, 1993b, 1996b). Esta tecnologia permite proporcionar mais oportunidades aos professores para trabalharem com estudantes em projectos onde mutuamente implementarão as suas ideias.

6.7 Testes de Usabilidade

6.7.1 Introdução

Existem diversas metodologias que podem ser utilizadas como ferramenta para a avaliação de uma interface. De entre estas podem-se destacar a observação e monitorização dos utilizadores, a colecta da opinião dos utilizadores, as experiências ou os testes de *benchmark*, a avaliação interpretativa e a avaliação preditiva (Preece *et al.*, 1994). Neste projecto, optou-se por realizar a observação e monitorização dos utilizadores, os testes e a colecta da opinião dos utilizadores, o que vai permitir uma avaliação mais controlada dos resultados. Em comparação com os demais métodos citados, as experiências permitem que os dados sejam colectados e analisados quantitativamente em situações mais específicas.

Esta secção expõe os testes de usabilidade realizados na interface *LogoKeyboard* com estudantes de uma escola básica e secundária. Apesar de terem sido aplicadas metodologias de interfaces Pessoa-Máquina para a avaliação do *LogoKeyboard*, não houve intenção de aplicar formalmente os métodos de “Engenharia de Usabilidade” (Nielsen, 1995) nem de realizar uma análise teórica profunda dos resultados obtidos. As experiências em sala de aula servirão de base para demonstrar que o *software LogoKeyboard* possibilita a programação em *Logo* por parte de indivíduos com deficiências físicas.

6.7.2 Objectivos

A definição dos objectivos no planeamento dos testes de usabilidade é fundamental, nomeadamente para verificar a funcionalidade e a qualidade de uma interface. Um conhecimento preciso do que se deseja compreender com a realização dos testes, facilita a escolha dos participantes, bem como das tarefas que estes devem executar, da métrica e dos atributos a utilizar para uma análise quantitativa e qualitativa (Dumas & Redish, 1999). Como refere Nielsen, a simplificação dos objectivos permite facilitar o estudo de usabilidade de um determinado produto (Nielsen, 1995).

Por conseguinte, o objectivo principal destes testes foi analisar o grau de dificuldade de utilização da interface *LogoKeyboard*, tendo em consideração as seguintes questões: *Os utilizadores têm a capacidade de manipular a interface de forma eficaz?* e *O LogoKeyboard permite substituir a utilização de um teclado convencional para construir programas de computador em Logo?* A realização destes testes focou-se na análise da simplicidade do processo de escrita dos comandos de programação em *Logo*, da facilidade de navegação nos menus, da facilidade de utilização do sistema de ajuda dos comandos em *Logo* e na observação das características principais do *LogoKeyboard* que são utilizadas durante o desenvolvimento de projectos educacionais.

6.7.3 Selecção do Grupo de Utilizadores

Após a definição dos objectivos dos testes de usabilidade, foi necessário decidir em que tipo de população iriam estes ser testados, visto que a escolha de uma população de estudo torna-se fundamental para atingir os objectivos estipulados para os testes usabilidade. Tratando-se de uma aplicação de acessibilidade para um determinado grupo de utilizadores e tendo presente a especificidade do *LogoKeyboard*, foi imprescindível escolher indivíduos portadores de limitações físicas.

Como professor de uma escola básica e secundária, o meu objectivo primordial foi o de encontrar uma turma que na sua constituição apresentasse alunos com defi-

ciências físicas, tarefa que não se revelou fácil. Começou-se por contactar diversas instituições, incluindo várias escolas. Também a instituição da APPC foi novamente contactada mas a população disponibilizada apresentava graves limitações cognitivas, apresentando características semelhantes às da população em estudo no projecto *Mindstorms* referido anteriormente nesta dissertação. Posteriormente, entrou-se em contacto com a Direcção Regional da Educação do Algarve (DREALG) de modo a obter uma listagem de alunos que apresentassem deficiências físicas. Foram efectuados alguns contactos com uma escola básica em Loulé, através dos quais se obtiveram algumas respostas positivas, inclusive a marcação de uma reunião com o seu Director, a fim de ser cedida a autorização para a realização dos testes de usabilidade neste estabelecimento. Com esta reunião compreendi que o processo de obtenção da permissão para a realização destes testes iria ser muitíssimo moroso e bastante burocrático.

Na escola onde lecciono existiam alguns alunos com limitações físicas, no entanto estes não faziam parte da minha turma. Como é essencial a compreensão da linguagem *Logo* para que seja possível o desenvolvimento de projectos educacionais utilizando o *LogoKeyboard*, pedi autorização ao Director da escola para transferir esses alunos para a minha turma. A autorização foi cedida e consegui uma população bastante razoável para a realização do estudo.

A população em estudo consiste numa turma do ensino secundário constituída por 21 alunos sendo 16 rapazes e 5 raparigas com idades compreendidas entre os 14 e 17 anos. A turma é bastante heterogénea, sendo composta por alunos de diversas nacionalidades e por dois alunos com deficiências físicas.

O Nilton, um rapaz de 17 anos, apresenta limitações físicas ao nível dos membros superiores. Apresenta graves problemas de desenvolvimento nas mãos e nos dedos, sendo que cada mão apenas apresenta três dedos de grandes dimensões. As suas limitações complicam e limitam a realização de diversas tarefas na escola.

A Ana é uma rapariga de 16 anos que apresenta limitações físicas dos membros inferiores e superiores. Ela necessita da ajuda de uma cadeira de rodas para poder

deslocar-se e, em sala de aula, diversas tarefas são realizadas através do auxílio de um computador portátil.

O Daniel é um rapaz de 15 anos que não apresenta qualquer deficiência física. É um aluno dedicado que gosta de ajudar os colegas em sala de aula.

Apesar de se ter conseguido uma população com elementos importantes para o meu estudo, não fiquei totalmente satisfeito porque também seria importante integrar neste projecto um aluno portador de paralisia cerebral. Através de uma professora da escola básica de Loulé, entrei directamente em contacto com os encarregados de educação de uma aluna, a Beatriz, portadora de paralisia cerebral. Foram explicadas aos pais quais as motivações por detrás deste estudo e estes não só aceitaram como também ficaram extremamente entusiasmados com este projecto. Aliás, foram os próprios pais que entraram em contacto com o Director da escola e obtiveram rapidamente uma resposta positiva para a realização dos testes de usabilidade.

Conheci a Beatriz através dos pais que me explicaram as suas limitações físicas. A Beatriz é uma rapariga de 13 anos com paralisia cerebral que revela distúrbios na postura e na coordenação dos membros inferiores e superiores. Tal como a Ana, a Beatriz também utiliza uma cadeira de rodas para se deslocar, e em sala de aula tem apoio extra de professores. Todas as tarefas em sala de aula são realizadas através do seu computador portátil.

É de salientar que todos estes alunos concordaram em participar neste projecto, tendo todos eles revelado grande maturidade ao longo dos vários meses de trabalho. É verdade que os testes de usabilidade com o *software LogoKeyboard* foram realizados com poucos alunos com limitações físicas. Contudo, apesar da reduzida população com estas características, o *software* também foi testado pelos restantes alunos da turma ao longo do ano lectivo no desenvolvimento de projectos educacionais.

6.7.4 Preparação dos Testes

Os testes de usabilidade são realizados num ambiente controlado e representam um mecanismo que visa a interacção entre os utilizadores e o sistema, com o propósito de desenvolverem diversas tarefas com objectivos específicos. O estudo dos dados recolhidos permite avaliar a qualidade da interface e encontrar lacunas de usabilidade, tornando assim fundamental planear as tarefas e os objectivos a alcançar. A observação directa dos utilizadores ao longo dos testes e a posterior observação das gravações de vídeo dos mesmos, ajudam a compreender a usabilidade da interface. As próximas secções explicam o modo como foram preparados e realizados os testes de usabilidade do sistema *LogoKeyboard*, e como foram escolhidos os utilizadores e as tarefas efectuadas por esses mesmos utilizadores.

6.7.5 Realização dos Testes

Antes de poder implementar uma filosofia *Logo* nas minhas aulas, tive que me dirigir novamente ao Director da escola para pedir autorização para conduzir este projecto. Foi necessário contactar todos os encarregados de educação para uma reunião a fim de aprovar o projecto. Durante a reunião fui confrontado com perguntas acerca do projecto *Logo*. Alguns pais apresentaram dúvidas acerca do projecto, nomeadamente por nunca terem ouvido falar da linguagem de programação *Logo*. No decorrer da reunião fui bastante claro, explicando que se pretendia seguir o currículo educativo da escola mas utilizando uma linguagem de programação diferente. Expliquei também o meu trabalho de pesquisa acerca do *LogoKeyboard* e os pais revelaram grande interesse, sobretudo os encarregados de educação dos alunos com limitações físicas. No final da reunião os encarregados de educação aprovaram o projecto permitindo deste modo avançar com este estudo.

O primeiro contacto dos alunos com a linguagem *Logo* foi bastante interessante, tendo sido revelado um grande entusiasmo por parte da turma, à excepção do Nilton que demonstrou alguma apreensão. Por forma a facilitar a aprendizagem da

linguagem, utilizei o *LogoKeyboard* para explicar o funcionamento dos comandos básicos. Um dos grandes obstáculos inicialmente demonstrados pelos alunos foi a dificuldade de retenção dos comandos e a sua funcionalidade. Recorrendo ao *LogoKeyboard* foi possível simplificar a utilização dos comandos e ajudar na compreensão da linguagem, utilizando para tal o sistema de animação integrado no teclado virtual. Os alunos manusearam facilmente o *LogoKeyboard* utilizando o rato para introduzir no *MicroWorlds* os comandos em *Logo*. A própria linguagem, em Inglês, não revelou ser uma barreira devido aos alunos apresentarem um nível bastante satisfatório. Após algumas aulas de experimentação e observação decidi falar da experiência que tinha tido com outras turmas com o projecto “Jogo do Labirinto”, no qual a tartaruga tem de encontrar a saída do labirinto sem nunca tocar as paredes do mesmo. Os alunos adoraram o projecto e quiseram realizar o mesmo trabalho utilizando o teclado virtual.

O *LogoKeyboard* foi utilizado na construção do jogo e foi implementado no *MicroWorlds*. A primeira fase do projecto consistiu no desenho do labirinto, utilizando para tal os comandos em *Logo*. O entusiasmo dos alunos foi aumentando ao longo do projecto ao ponto de desenvolverem vários esquemas de labirintos. A segunda etapa do projecto foi assinalada pela configuração das páginas do *MicroWorlds*, de modo a obter diversos labirintos organizados por nível de dificuldade. A terceira fase passou pela implementação do código das paredes do labirinto. Pretendeu-se que cada vez que a tartaruga atravessasse uma parede do labirinto, voltaria ao início do jogo. Para programar esta funcionalidade os alunos utilizaram o comando *Logo* “*Colorunder*” para verificar a cor existente por debaixo da tartaruga. Sempre que esta cor coincidissem com a cor da parede, a tartaruga reiniciava o labirinto. Esta fase do projecto suscitou alguns problemas uma vez que alguns alunos criaram paredes muito finas, o que permitia às tartarugas atravessarem as paredes do labirinto sem terem de voltar ao início do labirinto. Este problema foi resolvido utilizando o comando “*Setpenseize*” existente no *LogoKeyboard*. Sempre que os alunos apresentavam dúvidas acerca de algum comando, utilizavam o sistema de animação do

LogoKeyboard.

A quarta fase do projecto resumiu-se à criação de um código que possibilitasse à tartaruga mudar de labirinto e avançar nos diversos níveis existentes. De modo a atingir este objectivo, os alunos colocaram no final de cada labirinto uma porta secreta com uma determinada cor. Quando a tartaruga entrasse em contacto com esta cor, automaticamente a tartaruga era transportada para um novo nível constituído por um outro labirinto.

No decorrer do projecto tornaram-se essenciais as observações realizadas aos alunos para detectar as maiores dificuldades sentidas por estes na utilização da interface. Nesta fase inicial do projecto foram encontrados alguns problemas e foram seguidas algumas sugestões dos alunos, tais como:

- Disponibilizar uma opção para configurar a velocidade dos varrimentos;
- Colocar um número maior de cores no sistema de varrimento;
- Adicionar diferentes sons ao sistema de varrimento para ajudar na escolha dos comandos;
- Colocar o sistema de Zoom a funcionar com o sistema de varrimento.

Tendo em consideração estas sugestões, foram realizadas algumas correcções durante a execução do projecto “Jogo do Labirinto”.

Como nesta fase necessitava de obter dados quantitativos da utilização da interface, de modo a fortalecer a minha convicção de que o *LogoKeyboard* conseguia auxiliar indivíduos com limitações físicas para programarem na linguagem *Logo*, foram realizados testes com os alunos Nilton, Ana, Daniel e Beatriz. Nestes testes, com uma duração de 25 minutos, os alunos foram convidados a realizar dez actividades e a construir um pequeno programa em *Logo*. As actividades efectuadas pelos alunos para testar o teclado foram:

- Activar o sistema de varrimento do *LogoKeyboard* caso necessário;

- Alterar o vocabulário do teclado, utilizando para tal a opção “*Object and Control/Logic*”;
- Modificar a transparência da interface escolhendo a opção “*Disappear*”;
- Pressionar a tecla/comando “*Forever*” utilizando o sistema de varrimento;
- Configurar o sistema de varrimento com uma velocidade de 1000 milissegundos;
- Alterar as cores e o som do varrimento;
- Desactivar o sistema de varrimento;
- Activar o sistema de ajuda dos comandos em *Logo* e observar a descrição do comando “*Repeat*”;
- Desactivar o sistema de ajuda dos comandos em *Logo*;
- Fechar o *LogoKeyboard*.

A segunda fase deste estudo consistiu na construção de um pequeno programa em *Logo* designado por “*Square*”. O programa “*Square*” apresenta a particularidade de conter diversos caracteres que podem suscitar dificuldades de digitação num teclado convencional quando utilizado por indivíduos que apresentam limitações físicas. Este programa é constituído por quatro linhas e 46 caracteres:

```
to square
  make "n 4
  repeat (:n) [fd 30 rt 90]
end
```

Esta tarefa teve como objectivo primordial canalizar os estudantes para a criação de um pequeno quadrado utilizando diferentes técnicas de *input*. Durante a implementação do programa foi contabilizado o número total de teclas pressionadas e o tempo despendido por cada aluno para finalizar o programa. No final dos testes

cada aluno respondeu a um pequeno questionário revelando o grau de satisfação acerca da velocidade, facilidade de utilização e precisão do *LogoKeyboard*.

A última fase dos testes de usabilidade consistiu em quantificar o número de teclas pressionadas e o tempo dispendido para escrever os seguintes comandos de programação: *fd*, *setsize*, *setpensize* e *openworksheet*. Os dispositivos de input testados pelos alunos para escrever os comandos foram os teclados convencionais, o teclado virtual do *Windows* e o *LogoKeyboard*.

6.7.5.1 Caso de Estudo do Participante Daniel

Este aluno teve uma participação excelente no envolvimento deste projecto. No início esteve sempre presente para ajudar o Nilton e a Ana na utilização do *LogoKeyboard*. Para além de ajudar os colegas de turma, teve a capacidade de utilizar a interface de modo autónomo no desenvolvimento do projecto. Quando o *LogoKeyboard* foi introduzido na aula, o Daniel utilizou uma das ferramentas principais do teclado, o sistema de ajuda, que lhe facultou uma descrição dos comandos em *Logo*. Ele confirmou que “esta ferramenta é excelente para aprender a linguagem de programação *Logo*, bem como o modo de a utilizar”. A sua evolução foi fantástica, construiu o seu próprio conhecimento com os seus próprios erros, tendo conseguido desenvolver o projecto “Jogo do Labirinto” sem nunca revelar dificuldades na linguagem nem na utilização da interface. O Daniel realizou as dez actividades sem qualquer problema e de forma rápida. A construção do programa “*Square*” foi bastante rápida e sem qualquer erro de manipulação do teclado.

Na última fase do estudo o Daniel revelou maior rapidez para escrever os vocabulários do *Logo* utilizando um teclado convencional, mas revelou melhor performance utilizando o *LogoKeyboard* em vez do teclado virtual do *Windows*.

6.7.5.2 Caso de Estudo do Participante Nilton

Na sala de aula, o estudante Nilton aprendeu a programar em *Logo*. No entanto, rapidamente se verificou que este aluno estava desmotivado e frustrado por

não conseguir acompanhar o ritmo de trabalho dos colegas ao utilizar um teclado convencional. Decidiu-se então incentivar o Nilton a utilizar o *LogoKeyboard*, como os seus colegas, para continuar o seu trabalho. Manuseando um rato para controlar a interface, o Nilton conseguiu adaptar-se ao teclado, programando rapidamente e, até mesmo, ultrapassando alguns dos seus colegas. Ao longo do ano escolar, o Nilton adquiriu confiança, ultrapassando as suas limitações e revelando ser um bom programador. Uma frase citada pelo próprio Nilton revela a apreciação do teclado: “com o *LogoKeyboard*, eu tive a possibilidade de beneficiar das mesmas oportunidades que os meus colegas - aprender *Logo*”. Este aluno gradualmente revelou grande entusiasmo com o projecto “Jogo do Labirinto”. Diversos esboços de labirintos foram construídos pelo Nilton antes de decidir qual o desenho gráfico que iria escolher no seu projecto (ver Figura 6.8). Todas essas tentativas foram positivas para melhorar a sua performance e adaptação ao teclado virtual. Durante a realização das diversas actividades, o Nilton apenas se esqueceu de fechar o sistema de ajuda dos comandos em *Logo*. Apesar deste pequeno esquecimento, as restantes tarefas foram cumpridas facilmente. A construção do programa “*Square*” foi bastante rápida, revelando grande pratica e segurança na utilização da interface. O Nilton mostrou clara preferência pelo *LogoKeyboard* relativamente ao teclado convencional ou ao teclado virtual do *Windows*. A sua performance foi mais eficiente com o *LogoKeyboard* revelando maior rapidez na construção dos comandos em *Logo*.

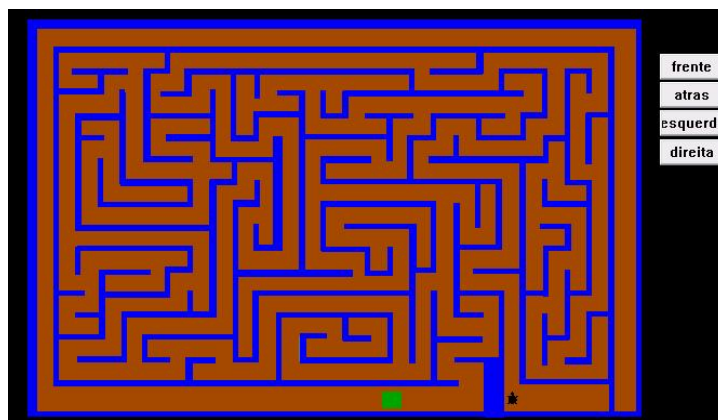


Figura 6.8: Projecto “Jogo do Labirinto” implementado pelo Nilton utilizando o teclado virtual *LogoKeyboard*

6.7.5.3 Caso de Estudo da Participante Ana

A Ana, a única rapariga da turma com limitações físicas, demonstrou uma interacção com o computador bastante razoável apesar de não gostar de utilizar um rato convencional. Em vez de utilizar um rato, a Ana prefere utilizar um *trackball* porque se torna mais confortável e menos cansativo para ela. No início das aulas, para programar em *Logo*, a Ana utilizou o teclado virtual do *Windows*, o qual já estava habituada a manusear. No entanto, esta tentativa foi extremamente negativa e desmotivadora devido ao facto de o tempo dispendido na escolha das teclas ser demasiado elevado. A utilização do sistema automático de selecção de teclas do teclado virtual do *Windows* revelou grande lentidão na escrita dos comandos em *Logo*, reduzindo a possibilidade da aluna acompanhar os colegas em sala de aula. Decidiu-se incentivar a Ana a utilizar o *LogoKeyboard* e manusear o sistema de varrimento. No início da experiência ela apresentou algumas dificuldades, demonstrando alguma confusão na escolha dos comandos. O próprio sistema de varrimento apresentou algumas dificuldades na orientação e posição das cores que são apresentadas no teclado. Este problema foi resolvido através da utilização de sons diferenciados para cada secção do varrimento apresentado no teclado. Os sons permitiram ajudar a Ana na sua orientação e concentração durante a selecção das teclas do *LogoKeyboard*. A aluna revelou também alguma desorientação no processo de configuração da interface.

Após algumas aulas e algum treino, a Ana melhorou o seu desempenho revelando maior segurança na escolha dos comandos. Inclusivamente, a estudante conseguiu configurar a interface para permitir aumentar a velocidade do sistema de varrimento e aumentar, dessa forma, o seu desempenho na implementação do projecto “Jogo do Labirinto”. A Ana conseguiu concluir o projecto com grande maturidade e confidenciou que em casa gostaria de continuar a desenvolver projectos em *Logo* com o auxílio do *LogoKeyboard* (ver Figura 6.9).

Ao longo da construção do programa “*Square*”, a Ana mostrou grande entusiasmo e conseguiu alcançar os objectivos propostos, apesar de uma certa lentidão em

alguns casos.

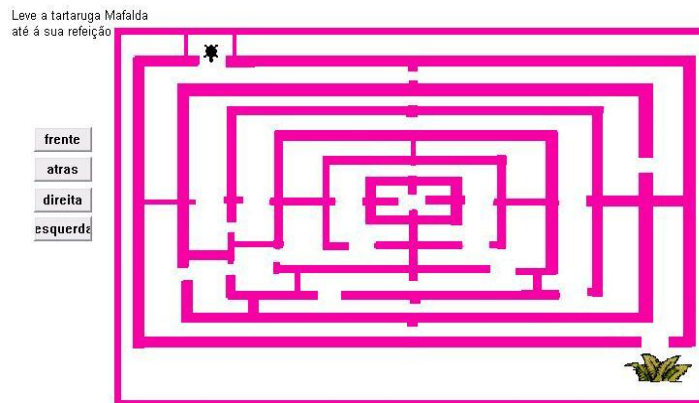


Figura 6.9: “Jogo do Labirinto” criado pela Ana utilizando o *LogoKeyboard*.

6.7.5.4 Caso de Estudo da Participante Beatriz

A Beatriz é uma rapariga simpática com grande interesse em aprender coisas novas. O primeiro contacto com a Beatriz foi muito positivo, tendo-se rapidamente criado uma grande afinidade. Na primeira sessão expliquei à Beatriz o meu trabalho e mostrei-lhe alguns trabalhos realizados por outros alunos. A aluna achou bastante interessante e perguntou se poderia “utilizar o *Logo* para fazer desenhos na disciplina de Área de Projectos”, questão à qual respondi afirmativamente. As sessões realizadas com a Beatriz decorreram durante a aula de Área de Projecto.

O resto da turma seguia os conteúdos programáticos da professora Carla Dias, que envolviam tarefas práticas de desenho e montagem de figuras. Devido às limitações físicas da Beatriz, estas tarefas tornavam-se praticamente impossíveis de serem realizadas. O manuseamento de objectos torna-se muito complicado e a utilização do computador é a única ferramenta que a consegue ajudar. O primeiro contacto com a linguagem *Logo* foi bastante interessante. Fiquei bastante surpreendido com a rapidez de aprendizagem da Beatriz, percebeu facilmente como movimentar a tartaruga no ecrã. Para a realização desta tarefa a utilização de um teclado convencional surgia como um enorme obstáculo. Através do uso do *LogoKeyboard* e de um rato, ela conseguiu realizar os primeiros desenhos utilizando a linguagem *Logo*.

Os primeiros desenhos realizados pela Beatriz foram um quadrado e um rectângulo. A dificuldade inicial sentida pela aluna foi encontrar no *LogoKeyboard* os comandos que ela necessitava para controlar a tartaruga.

Ao longo de três meses acompanhei a Beatriz com aulas de 90 minutos semanais. Durante as várias sessões a Beatriz melhorou o seu desempenho apesar de por vezes revelar alguma dificuldade no manuseamento do rato. Expliquei-lhe que existia um sistema de varrimento que poderia ajudá-la na realização destas tarefas. Ela referiu que não conhecia este método e que preferia utilizar o rato. Durante as sessões a aluna experimentou diversos comandos em *Logo* e realizou diversas figuras. No segundo mês de actividades incentivei a Beatriz para realizar o projecto do Labirinto. Mostrei-lhe alguns trabalhos de alunos e ela gostou. Durante as nossas sessões e em casa, a Beatriz começou a implementar o projecto. Na Figura 6.10 está apresentado o trabalho desenvolvido pela Beatriz.

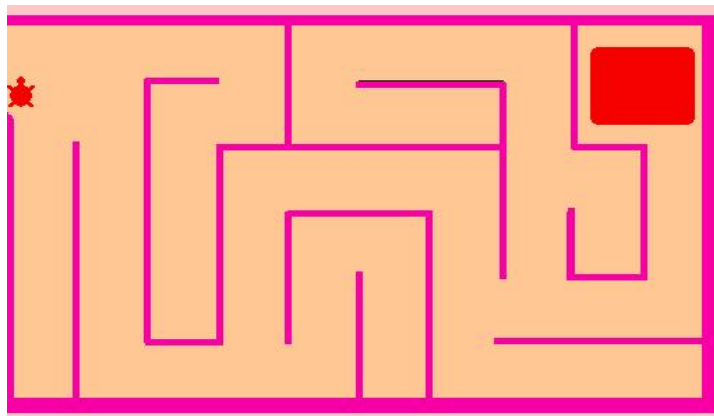


Figura 6.10: “Jogo do Labirinto” desenvolvido pela Beatriz utilizando o *LogoKeyboard*.

Durante a realização dos testes a Beatriz não revelou dificuldades na execução das actividades propostas, tendo apenas cometido um erro na construção do programa “*Square*”. Na última fase dos testes a Beatriz revelou ser incapaz de utilizar um teclado convencional para escrever os comandos em *Logo*. A utilização do teclado virtual do *Windows* foi descrita por ela de “aborrecida” pelo facto da escrita de um comando ser demasiado morosa. Com o *LogoKeyboard* a experiência foi positiva apesar de ela preferir utilizar o rato em vez do sistema de varrimento do teclado.

6.7.6 Análise e Resultados

Ao longo de vários meses de trabalho foi possível apurar que um indivíduo com deficiências físicas que consiga manipular um dispositivo de apontar, tal como um rato, ou pressionar um botão de toque para manipular o *LogoKeyboard*, é capaz de programar mais rapidamente do que utilizando um teclado convencional. Observou-se que a própria organização das teclas na interface reflectiu um desempenho positivo no desenvolvimento de projectos em *Logo*, facilitando a procura e a utilização dos comandos de programação. O *LogoKeyboard* foi adequado para auxiliar os estudantes positivamente, aumentando o desempenho, a velocidade e a acessibilidade para programar. Os utilizadores com graves limitações físicas que não conseguem utilizar um teclado convencional ou um dispositivo de apontar tiveram a oportunidade de usufruir do sistema de varrimento do *LogoKeyboard*. Durante as primeiras aulas, este sistema provocou alguma ansiedade nos utilizadores devido ao tempo de espera para seleccionar os comandos em *Logo* no teclado. O tempo de espera não se tornou um aspecto negativo, devido ao facto de um comando completo em *Logo* poder ser seleccionado de uma só vez, em vez de digitar os caracteres individuais que constituem o comando. Este aspecto permitiu melhorar o potencial de digitação de indivíduos com limitações físicas. Os indivíduos que não apresentam problemas motores não têm grandes vantagens em utilizar o *LogoKeyboard*, excepto para utilizar o sistema de ajuda disponível no teclado. Para conseguir esmiuçar a relevância da interface, foram realizados diversos testes na qual se encontra inserido a implementação do projecto “Jogo do Labirinto”. Os alunos com deficiências físicas conseguiram desenhar os labirintos e implementar funções e procedimentos utilizando de forma positiva as ferramentas da interface. Estes estudantes conseguiram acompanhar e realizar os mesmos trabalhos que os outros colegas da turma sem grandes dificuldades.

O objectivo do estudo realizado com o programa “*Square*” permitiu-me testar as maiores dificuldades sentidas pelos estudantes com limitações físicas: a manipulação simultânea de duas teclas num teclado convencional. Esta combinação de teclas é necessária para produzir caracteres especiais que são utilizados frequentemente num

programa de computador (por exemplo, [,], :, (,), "). O *LogoKeyboard* minimizou este problema permitindo introduzir esses caracteres pressionando apenas uma única tecla. Este aspecto foi sem dúvida, uma das grandes vantagens que os estudantes apreciaram. A tabela 6.1 sumaria o número total de teclas pressionadas e o tempo despendido por cada aluno durante a criação do programa “*Square*”. Na última fase dos testes de usabilidade conseguiu-se obter resultados importantes para perceber a eficácia do *LogoKeyboard* comparativamente com o uso de um teclado convencional e do teclado virtual do *Windows*. As tabelas 6.2, 6.3, 6.4 e 6.5 evidenciam os resultados obtidos pelos alunos.

No final cada aluno respondeu a um pequeno questionário revelando o grau de satisfação acerca da velocidade, da facilidade de utilização e da precisão do *LogoKeyboard* (ver Tabela 6.6).

Tabela 6.1: Resumo dos dispositivos utilizados pelos alunos, número de teclas pressionadas e tempo despendido na construção do programa “*Square*”.

Estudante	Dispositivo	Número de teclas pressionadas	Tempo (seg)
Daniel	Teclado Físico	56	46
Ana	Teclado Virtual	108	404
Nilton	Teclado Virtual	27	73
Beatriz	Teclado Virtual	101	323

Tabela 6.2: Teclado convencional utilizado pelos alunos para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: *Fd*, *Setsize*, *Setpensize* e *OpenWorkSheet* (A Ana e a Beatriz não conseguiram interagir com o teclado).

Teclado Convencional					
Comandos	Nº de Teclas pressionadas	Tempo (seg)			
		Daniel	Nilton	Ana	Beatriz
Fd	2	2	6	-	-
Setsize	7	5	18	-	-
Setpensize	10	7	25	-	-
OpenWorkSheet	13	11	41	-	-

As observações e os estudos realizados com os alunos com limitações físicas

Tabela 6.3: Teclado virtual da *Microsoft* utilizado pelos alunos com o sistema de varrimento para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: *Fd*, *Setsize*, *Setpenseize* e *OpenWorkSheet*.

Teclado virtual da Microsoft (com sistema de varrimento)					
Comandos em Logo	Nº de Teclas pressionadas	Tempo (seg)			
		Daniel	Nilton	Ana	Beatriz
Fd	4	5	11	10	13
Setsize	14	35	51	49	63
Setpenseize	20	55	74	83	106
OpenWorkSheet	27	102	189	201	244

Tabela 6.4: *LogoKeyboard* utilizado pelos alunos com o sistema de varrimento para determinar o número de teclas pressionadas e tempo despendido para escrever os comandos: *Fd*, *Setsize*, *Setpenseize* e *OpenWorkSheet*.

LogoKeyboard (com sistema de varrimento)					
Comandos em Logo	Nº de Teclas pressionadas	Tempo (seg)			
		Daniel	Nilton	Ana	Beatriz
Fd	4	4	6	5	8
Setsize	4	8	9	10	17
Setpenseize	4	13	16	15	22
OpenWorkSheet	11	30	42	46	95

Tabela 6.5: Resultado do número de teclas pressionadas por cada aluno utilizando um rato para manusear o teclado virtual da *Microsoft* e o *LogoKeyboard*.

	Teclado virtual da Microsoft	LogoKeyboard
Comandos em Logo	Nº de Teclas Pressionadas	Nº de Teclas Pressionadas
Fd	2	
Setsize	7	1
Setpenseize	10	
OpenWorkSheet	13	3

foram benéficos para a compreensão das necessidades e problemas que os alunos com limitações físicas enfrentam na construção de programas de computador utilizando a linguagem de programação *Logo*. Os testes de usabilidade tiveram um

Tabela 6.6: Resultados do questionário relativamente a velocidade, facilidade de utilização e precisão quando utilizado o *LogoKeyboard* e um teclado físico. Os resultados são apresentados numa escala de 1 (Fraco) até 5 (Bom). (F- Físico, V- Virtual)

Estudante / Teclado	Velocidade		Facilidade de utilização		Precisão	
	F	V	F	V	F	V
Daniel	5	4	4	4	4	5
Ana	1	2	2	5	2	4
Nilton	2	5	2	5	3	4
Beatriz	1	4	1	5	1	5

teor importante para analisar as necessidades e dificuldades dos alunos, mas é de salientar que todas as actividades em sala de aula tiveram um papel fundamental para a construção e aperfeiçoamento do próprio *LogoKeyboard*.

A utilização do *LogoKeyboard* revelou ser bastante eficaz na aprendizagem da linguagem de programação *Logo* com todos os estudantes da turma. O teclado permitiu também que o acesso aos comandos se tornasse muito mais eficaz, nomeadamente devido à utilização do sistema de ajuda que possibilita a aprendizagem de um maior número de vocabulário *Logo*.

6.8 Discussão

Os testes de usabilidade revelaram-se uma ferramenta extremamente importante para a construção do desenho da interface e para a compreensão dos aspectos que deveriam ser modificados e melhorados. Desde a criação do primeiro modelo do teclado *Logo*, entendeu-se que o melhor mecanismo para ajudar indivíduos com deficiências motoras seria a implementação de um sistema automático que permitisse facilitar a interacção dos utilizadores com os interpretadores de *Logo*. O sistema de varrimento surgiu como sendo, desde logo, o mais indicado para manusear e configurar o *LogoKeyboard*. As experiências com os participantes contribuíram fortemente para alcançar um produto final. Cada participante revelou capacidades físicas diferentes o que permitiu construir diversas opções de desenho da interface, nomeadamente

a possibilidade de modificar a cor do sistema de varrimento, a sua velocidade e tamanho do teclado. Todas estas opções foram importantes para facilitar o manuseamento do *LogoKeyboard*. A própria possibilidade de poder utilizar um sistema de ajuda para aprender o vocabulário da linguagem *Logo* serviu de apoio na aprendizagem dos alunos. A estrutura e organização dos testes de usabilidade revelam sobretudo um carácter qualitativo focalizado principalmente na observação e contextualização. Todo o processo apresenta uma ordem cronológica bem estudada. Primeiramente, como demonstrado no capítulo 5, foi realizado um estudo da linguagem *Logo* com duas turmas do ensino secundário com o objectivo de entender o funcionamento da linguagem *Logo* junto dos utilizadores. Não se pode construir uma aplicação sem se conhecer minuciosamente as potencialidades da linguagem. Depois de se conhecerem as dificuldades dos alunos na manipulação da linguagem *Logo*, foi fundamental estudar a interacção dos alunos portadores de deficiência motora com esta plataforma. Teve-se em consideração que quanto mais diversificado fosse o conjunto de deficiências, mais completo seria o estudo realizado. Deve ser claro que apesar dos utilizadores apresentarem deficiências motoras diversificadas este aspecto não teve impacto negativo nos resultados. O principal objectivo na recolha dos resultados foi verificar se todos os utilizadores teriam capacidade de programar na linguagem *Logo* com a ajuda do *LogoKeyboard* e verificar se a interface facilita ou não a realização dessa tarefa. Com base nos resultados apresentados nas tabelas (6.2, 6.3, 6.4 e 6.5), pode-se constatar que o número de teclas pressionadas para digitar um comando em *Logo* é inferior quando a interface é utilizada. É de salientar que o tempo dispendido para escrever os comandos pode variar dependendo das deficiências motoras que o utilizador revela e se para a interacção com a aplicação foi utilizado o sistema de varrimento ou um rato. No entanto, seja qual for a deficiência motora do utilizador, desde que tenha capacidade suficiente para pressionar qualquer dispositivo, o número de teclas a pressionar para escrever um comando *Logo* será sempre o mesmo. A utilização de um teclado constituído por comandos em *Logo* foi muito bem conseguida e elogiada pelos utilizadores. Os testes realizados com os alunos revelam que o *LogoKeyboard* consegue facilitar a digitação dos coman-

dos em *Logo* quando comparado com a utilização de um teclado convencional ou do teclado virtual da Microsoft. É fundamental referir que com o uso de um teclado convencional a Ana e a Beatriz seriam incapazes de digitar os comandos em *Logo* e programar. Quando se trabalha com alunos que revelam deficiências motoras e cujos conteúdos curriculares se encontram abrangidos por linguagens de programação, torna-se complicado motivar e exigir concentração em sala de aula. Por este motivo foi necessário por vezes encontrar soluções para conseguir motivar estes estudantes. O projecto “Jogo do Labirinto” foi um dos processos para cativar e revelar o que se pode fazer com as linguagens de programação. Um acontecimento interessante que ocorreu no final deste estudo foram os comentários de Nilton e Ana que me motivaram a construção de outra aplicação: “O professor deveria ter um programa lúdico para ajudar os utilizadores a treinar com os sistemas de varrimento” e “O professor deveria facultar um sistema que permitia que os alunos relaxem nas aulas durante a criação dos programas em *Logo* e assim permaneçam concentrados”.

6.9 Sumário

Neste capítulo foi relatado um estudo com o *LogoKeyboard* realizado com uma turma de 21 alunos do ensino secundário e de uma aluna do ensino básico. Pretendeu-se analisar o grau de dificuldade de utilização da interface *LogoKeyboard*, tendo em consideração algumas questões. Os estudantes em estudo têm idades compreendidas entre os 13 e 17 anos e apresentam diferentes características. Duas alunas apresentam limitações físicas nos membros superiores e inferiores e um aluno apresenta graves limitações físicas nos dedos e nas mãos.

Ao longo de várias aulas cada aluno realizou diversas actividades com a linguagem *Logo*, desde a implementação de um jogo, denominado por “Jogo do Labirinto”, passando pela construção do programa “*Square*” e pela execução de diversas acções utilizando o *LogoKeyboard*.

Os resultados obtidos foram bastante optimistas, revelando que alguns alunos

não poderiam programar na linguagem *Logo* sem o auxílio de uma ferramenta adaptada como o *LogoKeyboard*. No final das experiências foram efectuados questionários para medir qualitativamente o grau de satisfação dos alunos acerca da velocidade, facilidade de utilização e precisão do *LogoKeyboard*.

7

Ambientes Lúdicos de Aprendizagem Computacional

7.1 Introdução

A criação, nas escolas, de ambientes propícios para a concentração e o desenvolvimento do raciocínio lógico, depende essencialmente das estratégias elaboradas pelo professor. Diversas técnicas têm sido utilizadas para melhorar o desempenho dos alunos na sala de aula como sendo a música, os jogos, a Internet e os espaços multimédia. Também nas aulas de informática são sentidas grandes dificuldades na aprendizagem da programação de computador. Robins *et al.* (2003) relata que o problema dos estudantes no início da aprendizagem da programação de computador advém da falta de motivação e concentração. Vários estudos realizados por Winslow (1996) concluem que os programadores novatos estão limitados na organização do conhecimento, apresentam carência de modelos mentais detalhados, possuem pouca memorização de estratégias na resolução de problemas e apresentam também dificuldades na criação de programas com raciocínio abstracto. Os jogos educacionais proporcionam um ambiente de aprendizagem flexível permitindo aumentar o nível de concentração e motivação na sala de aula (Moursund, 2006). Com o auxílio dos professores, os alunos podem aprender num ambiente mais reflexivo e alcançar os objectivos de aprendizagem de modo mais explícito (Kirschner *et al.*, 2006).

Os alunos com limitações físicas podem apresentar dificuldades no manuseamento de jogos educacionais. As dificuldades sentidas pelo utilizador na incapacidade ou dificuldade em manusear correctamente uma aplicação lúdica podem provocar um ambiente desmotivador e reduzir o nível de concentração da criança. É importante disponibilizar técnicas em aplicações lúdicas que possibilitam a interacção do utilizador com o computador. Neste capítulo explora-se o modo como os jogos podem influenciar a aprendizagem dos alunos no meio educacional, salientando a importância das técnicas que permitem a entrada de dados numa aplicação informática em contexto lúdico, de modo a auxiliar pessoas com deficiências físicas.

7.2 Utilidade dos Jogos na Aprendizagem

A utilização de *software* educativo nas salas de aula tem sofrido alterações ao longo dos anos. Com a acessibilidade fornecida pela Internet, diversos professores focam-se na importância das animações para apresentar diferentes conceitos e combinam essas abordagens com a utilização de jogos (Conati & Zhao, 2004). Trabalhos anteriores demonstram que agentes pedagógicos animados amplificam a motivação e empenho dos alunos (Johnson *et al.*, 2000). Os alunos inseridos num ambiente pedagógico atractivo e divertido trabalham melhor e durante mais tempo e são mais susceptíveis de aumentar o conhecimento aprendido do que aqueles que simplesmente desejam concluir o curso (Mitrovic & Suraweera, 2000). Os mundos virtuais dos jogos são contextos de aprendizagem bastante ricos, uma vez que tornam possível aos estudantes desenvolverem um conjunto de práticas sociais efectivas. Os estudantes aprendem melhor com experiências directas, actividades e descobertas (Barnes *et al.*, 2007).

O famoso jogo do *Sudoku* tornou-se uma ferramenta educacional para professores nas escolas, uma vez que proporciona uma excelente ajuda no pensamento lateral e no desenvolvimento analítico. Norte e Lobo (2008a) utilizaram o puzzle *Sudoku* numa escola básica para auxiliar estudantes com deficiências físicas a permanecer relaxados e a desenvolverem competências, uma vez que este tipo de puzzle ajuda

a melhorar a concentração e o pensamento lógico (Moursund, 2006). O *Sudoku* auxilia as crianças a raciocinar para a resolução de problemas e apenas precisam de ter algumas informações em mente para definir as estratégias de jogo. No entanto, há que também salientar que na maior parte das vezes o grande problema dos alunos na sala de aula é a falta de confiança (Downes, 1993).

7.3 *Sudoku Access*: O Jogo do *Sudoku* para Pessoas com Deficiências Físicas

7.3.1 O Jogo do *Sudoku*

O jogo do *Sudoku* é um puzzle com 81 células colocadas numa grelha 9x9. A regra do jogo é trivial: basta preencher a grelha do puzzle de modo que os dígitos de 1 a 9 ocorram exactamente uma só vez em cada linha, coluna e em cada grelha 3x3. No início do jogo a grelha é composta por vários dígitos que já se encontram colocados dentro das células e que não podem ser alterados (ver Figura 7.1). O objectivo é preencher as células vazias seguindo a regra explicada anteriormente. Para resolver o jogo não são necessários conhecimentos especiais de matemática ou de cálculos, trata-se apenas de um simples e divertido jogo de lógica, no qual apenas é requerida concentração.

O puzzle *Sudoku* mais comum é composto por uma grelha 9x9, mas não é o único que é utilizado, existindo também outras versões. Com qualquer número inteiro a positivo, podemos representar uma grelha de *Sudoku* de ordem a com a^2 linhas, a^2 colunas, e a^2 blocos. A grelha tem um total de a^4 células, que necessitam de ser preenchidas com números inteiros no intervalo de 1 a a^2 . Muitas pessoas focam-se num objectivo durante o processo de solução quando é realizado o preenchimento de células conjecturando ter colocado o número na posição correcta. Por vezes ocorre erros na resolução do puzzle forçando o jogador a recuar e alterar células na grelha do *Sudoku*. A utilização desta técnica pode ser encarada como um processo lógico. Quando uma pessoa preenche uma célula e verifica que as outras células não têm

Problema									Solução									
				3	1			9	6	7	2	4	3	1	5	8	9	
				2			6	1		3	9	8	5	2	7	6	1	4
		5							2	4	1	5	9	8	6	3	7	2
1				7		9	4		8	1	3	6	7	5	9	4	2	8
	8			6		3			1	2	8	9	6	4	3	7	5	1
5	4							3		5	4	7	2	1	8	9	3	6
				3		2		9	7	8	5	4	3	6	2	1	9	7
7	6				9	5	2	4	3	7	6	1	8	9	5	2	4	3
	2	3			7		8			9	2	3	1	7	4	8	6	5

Figura 7.1: Exemplo de um puzzle *Sudoku* e a respectiva solução. O objectivo é preencher os espaços vazios com dígitos de 1 até 9 de modo que cada dígito ocorra apenas uma vez em cada linha, coluna e quadrado 3x3 assinalados com linhas a negrito.

solução, então ele/ela descobriu uma relação lógica entre as células. Para jogar ao *Sudoku* no computador a maioria das pessoas utiliza um rato ou um teclado convencional para preencher as células. Quando o jogador é uma pessoa com deficiência física, esta interacção pode tornar-se um grande problema, sendo necessárias alternativas.

7.3.2 Metodologias de Desenho do Jogo

O desenvolvimento de interfaces para jogos educacionais e a sua implementação requerem uma análise rigorosa e selectiva. Este processo torna-se ainda mais complexo quando a interface é direccionada para pessoas com deficiências físicas. O jogo do *Sudoku* desenvolvido por nós para este tipo de utilizadores foi implementado com o nome *Sudoku Access*. Esta interface foi cuidadosamente elaborada para proporcionar uma interacção fácil para pessoas com limitações físicas, apresentando um ambiente adaptado para suportar diferentes tipos de ajuda. Para interagir com o jogo, os utilizadores com limitações físicas podem recorrer a um teclado convencional, a um botão de toque (switch) ou a um sistema de reconhecimento de voz. Um utilizador que apenas pode efectuar um ou dois movimentos físicos pode utilizar botões de toque para enviar informação e interagir com o jogo. Através do reconhe-

imento de voz é possível introduzir dados no *Sudoku Access* apenas pronunciando números. O reconhecimento de voz é uma ferramenta eficaz para indivíduos que necessitam de soluções alternativas de interação com aplicações informatizadas (Dai *et al.*, 2004; Karimullah & Sears, 2002; Mihara *et al.*, 2005). Foi utilizado o *software Speech Application Programming Interface* (SAPI), uma API desenvolvida pela Microsoft para permitir a utilização do reconhecimento e síntese de voz em aplicações Windows.

Relativamente às características do puzzle, estas podem ser utilizadas e configuradas sem qualquer ajuda de um assistente. O utilizador pode configurar o nível de dificuldade do puzzle, a velocidade de varrimento, o número de ciclos de repetição do varrimento, o som do varrimento, a cor do varrimento, o dispositivo utilizado para introduzir dados no puzzle (rato, botão de toque, tecla de espaços ou o sistema de reconhecimento de voz) e o tamanho do puzzle. Todas estas opções podem ser controladas através de um único botão de toque, microfone ou por qualquer dispositivo que simula o toque de um rato. Na figura 7.2 está representado o *layout* do *Sudoku Access*. O jogo pode ser descarregado grátis no endereço: <http://w3.ualg.pt/~snorte/SudokuAccess.htm>.

Dependendo do tipo de incapacidade física apresentada pelo utilizador, diferentes técnicas foram utilizadas para interagir com o jogo. Utilizadores com paraplegia ou distrofia muscular podem manusear o puzzle *Sudoku* recorrendo ao método do varrimento (Majaranta & Rähkä, 2002; Norte & Lobo, 2008a). Este método apresenta-se como uma mais valia porque permite ao utilizador usar apenas uma única tecla ou botão de toque para efectuar escolhas no puzzle. Como se pode verificar na figura 7.3, existem dois grupos principais de varrimento. O primeiro é utilizado para seleccionar a grelha do *Sudoku* e o segundo para seleccionar as opções do menu.

Quando um dos grupos é escolhido o varrimento é dividido em pequenos grupos (subgrupos) de itens e estes são, posteriormente, seleccionados individualmente. A hierarquia dos subgrupos é realçada com uma determinada cor até atingir a selecção



Figura 7.2: *Sudoku Access*: o jogo do *Sudoku* para pessoas com limitações físicas.

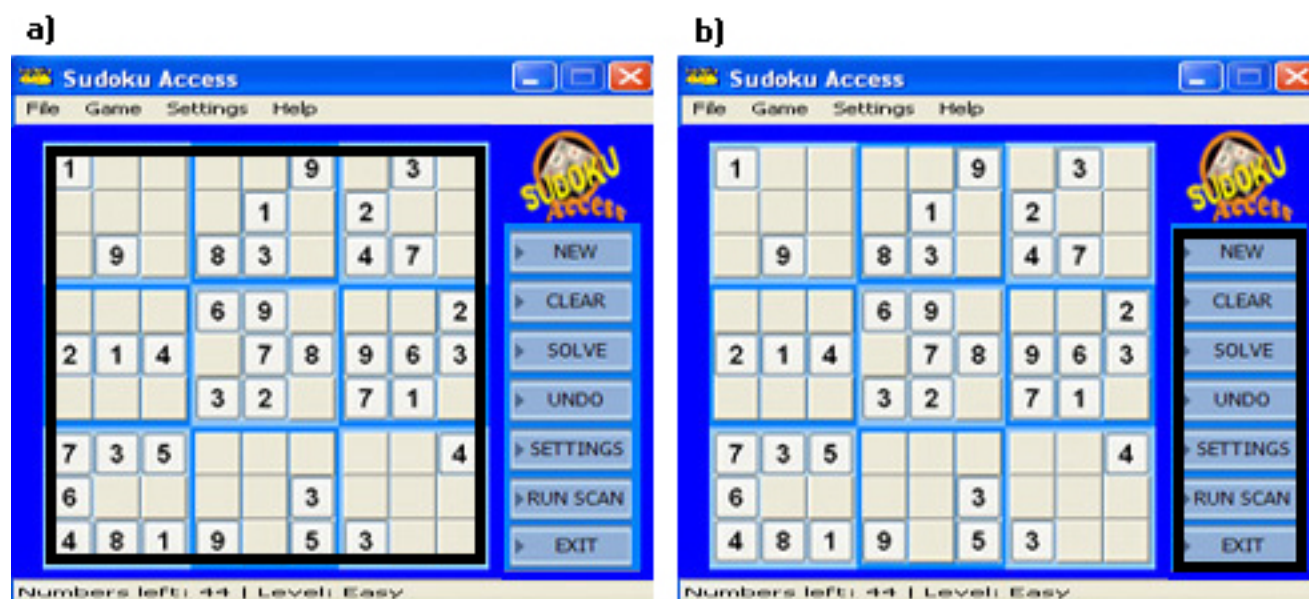


Figura 7.3: Exemplo do funcionamento do sistema de selecção por varrimento no jogo *Sudoku Access*, onde os itens (ou grupos de itens) vão sendo seleccionados automática e sequencialmente, após um determinado tempo predefinido. Existem dois tipos principais de varrimento: a) varrimento para seleccionar a grelha do *Sudoku*, permitindo desta forma ao utilizador introduzir números; b) varrimento para escolher as opções de configuração do puzzle.

individualizada de cada item (Norte & Lobo, 2007; Shein, 1997; Nisbet *et al.*, 1998). Quando o item é seleccionado numa célula do *Sudoku* e é realçado com uma cor, permanece à espera que o utilizador efectue a escolha numérica. Nesta fase, outro sistema de varrimento é activado num teclado numérico, sendo que a próxima escolha será o número desejado. Para manusear o sistema de varrimento, o utilizador necessita de usar um botão de toque (switch) ou pressionar a tecla “Espaço” (Condado *et al.*, 2004). Na figura 7.4 está representado o processo descrito anteriormente.

Todas as opções de desenho do *Sudoku Access* acompanharam as necessidades dos utilizadores. A possibilidade de poder configurar o sistema de varrimento permite aos utilizadores progredirem na utilização da aplicação. As cores das células das grelhas da aplicação ajudaram a compreender a lógica do jogo. Os sons diferenciados também tiveram como objectivo melhorar orientação dos utilizadores nas suas escolhas. A escolha da construção do sistema de reconhecimento de voz surgiu como opção para poder oferecer outro meio de “Input” que fosse mais confortável.

O sistema de reconhecimento de voz converte a voz em texto permitindo a interacção com a grelha do *Sudoku* e a configuração de diversas opções. Assim sendo, para introduzir números no *Sudoku*, o utilizador necessita de escolher qual a linha da grelha que pretende (1-9) apenas pronunciando o número. No caso do número escolhido não ser reconhecido correctamente, o utilizador deverá pronunciar a palavra “No”. Se estiver correcto o utilizador deverá dizer “Yes”. No caso da escolha ser positiva o próximo passo é escolher a coluna (1-9) utilizando o mesmo mecanismo. Finalmente, o utilizador pode escolher o número que irá preencher a célula desejada. A figura 7.5 apresenta um exemplo do *Sudoku Access* controlado por voz.

Com o sistema de reconhecimento de voz também é possível configurar outras opções. Por exemplo, no menu de opções é possível escolher as opções “New”, “Clear”, “Solve”, “Undo”, “Settings” e “Exit”, pronunciando números entre 10 e 15. Existem ainda outras opções de configuração que também podem ser controladas por voz: o nível de dificuldade do puzzle, a cor das linhas e colunas da grelha, e o tamanho do



Figura 7.4: Sistema de selecção do jogo *Sudoku Access*. Este sistema de selecção permite ao utilizador preencher as células do puzzle com os números, utilizando um simples botão de toque ou pressionando a tecla “Espaço”. a) Selecção do primeiro grupo, b) selecção do primeiro subgrupo, c) selecção da primeira linha do primeiro subgrupo, d) selecção do primeiro item da primeira linha do primeiro subgrupo, e) um outro sistema de selecção é activado no teclado numérico (visualizado no lado direito do puzzle), sendo que nesta fase o próximo “click” corresponde à escolha do número desejado, f) o número desejado é enviado à célula.



Figura 7.5: O sistema de reconhecimento de voz como processo de selecção no jogo *Sudoku Access*. Utilizando a voz, o utilizador escolhe números entre 1 e 5 para escolher a tarefa desejada. Neste exemplo o utilizador escolheu o número “4” que corresponde à activação do sistema de treino do reconhecimento de voz da Microsoft.

puzzle.

É de salientar que o *Sudoku Access* também possibilita a configuração do sistema de treino do reconhecimento de voz da Microsoft. O processo de ensinar o computador a reconhecer a voz é designado por “training” e leva apenas alguns minutos com a maioria dos utilizadores, sendo que quanto mais tempo o computador for treinado melhor são os resultados alcançados. Após o utilizador escolher o número correcto (“14”) para seleccionar a opção “Settings” o utilizador necessita de pronunciar outros números entre 1 e 5 para escolher a opção desejada (ver Figura 7.6).



Figura 7.6: Na primeira figura o utilizador escolhe a linha desejada (1-15) e a coluna (1-9) utilizando a própria voz. Na segunda figura, é exemplificado a escolha do número pretendido para preencher uma célula no *Sudoku Access* (linha 5, coluna 4).

7.4 Testes de Usabilidade

7.4.1 Introdução

Optou-se por conduzir os testes de usabilidade do *Sudoku Access* utilizando uma metodologia idêntica à utilizada nos testes de usabilidade com o *LogoKeyboard*. Foram realizadas experiências, observações, colecta de opiniões e monitorização com

utilizadores. Os testes realizados com o *Sudoku Access* foram implementados com estudantes de uma escola básica. A utilização dos testes servirão de base para revelar quais as dificuldades sentidas pelos alunos e de que forma podemos melhorar o desempenho de indivíduos com limitações físicas no manuseamento do jogo do *Sudoku*.

Esta secção descreve os testes de usabilidade realizados com alunos de uma escola de terceiro ciclo com o *software Sudoku Access*, bem como os resultados obtidos. A população em estudo revelou-se fundamental para auxiliar no melhoramento do *software* e proporcionar resultados bastante interessantes.

7.4.2 Objectivos

Segundo Nielsen (1994), um teste de usabilidade é organizado por cinco dimensões: Aprendizagem, Memorização, Erros, Eficiência e Satisfação. Quando se constrói um sistema é recomendado utilizar pelo menos duas destas dimensões, aquelas que apresentarem mais importância para o produto em questão. No caso do *Sudoku Access* as dimensões mais relevantes são a Eficiência e a Satisfação. Em *software* como jogos é importante proporcionar uma experiência agradável e eficiente para o utilizador, uma vez que isto pode afectar directamente a jogabilidade (Nielsen, 1995).

É neste contexto que os testes de usabilidade revelam um papel fundamental. Os testes implementados com os alunos que apresentam graves deficiências físicas são essenciais para averiguar as maiores dificuldades sentidas durante a utilização do *Sudoku Access*.

Os objectivos dos testes focaram-se essencialmente em dois aspectos importantes. O primeiro é encontrar qualquer característica da interface que seja confusa para o utilizador, o segundo é avaliar as funcionalidades do sistema de entrada de dados utilizando o sistema de selecção automático e o sistema de reconhecimento de voz. Por conseguinte, algumas questões revelaram grande importância neste estudo:

“o *Sudoku Access* permite melhorar a concentração e raciocínio lógico dos alunos em sala de aula?” e “o *Sudoku Access* oferece ferramentas necessárias para um utilizador com graves deficiências físicas poder utilizar a interface?”.

7.4.3 Selecção do Grupo de Utilizadores

O objectivo do desenvolvimento de testes de usabilidade torna-se praticamente estruturado quando se sabe em que população o estudo vai ser realizado. Neste caso era imprescindível efectuar testes com indivíduos que apresentassem limitações físicas.

Tive a oportunidade de obter alguns contactos de uma professora de uma escola básica de Loulé que sabia que o meu trabalho envolvia alunos com deficiências físicas. Entrei em contacto com o Director da Escola e pedi uma reunião para explicar o meu trabalho. Nessa reunião foi-me informado que existiam duas alunas com graves deficiências físicas, sendo que uma delas não podia frequentar as aulas fisicamente mas apenas por intermédio da Internet por vídeo-conferência. Além das aulas via Internet, a aluna recebia apoio educativo no seu domicílio duas vezes por semana.

Assim sendo, a população em estudo consistiu essencialmente em três estudantes (Daniela, Fátima e Jorge) de uma escola básica de Loulé com idades compreendidas entre os 13 e 15 anos. Um dos alunos não apresenta qualquer problema motor enquanto os restantes são portadores de graves problemas físicos. Um dos estudantes apresenta também algumas limitações cognitivas e problemas de dicção.

A Daniela é uma rapariga de catorze anos com uma doença degenerativa caracterizada por uma paralisia progressiva dos músculos e esta doença impossibilita-a de se deslocar à escola. As suas limitações físicas são de tal modo severas que apenas consegue movimentar os olhos e os dedos indicadores. A Daniela depende completamente do apoio da família e dos amigos. A Fátima é uma rapariga de treze anos com uma malformação congénita do tubo neural. Este *handicap* resulta de uma falha na formação das vértebras na coluna do feto, falha esta que, abrindo e tornando vul-

nerável o seu interior, provoca danos, por vezes graves, no sistema nervoso central. Esta doença é designada por *Spina Bífida*. A Fátima apresenta severas limitações para se movimentar, manipular objectos, e pronunciar palavras. Ela necessita do auxílio de uma cadeira de rodas para se movimentar. Na sala de aula a Fátima utiliza um computador para tentar suavizar as suas tarefas e colmatar as suas limitações. O terceiro elemento deste estudo é o Jorge, um rapaz de 15 anos que não apresenta limitações físicas, e que revela um perfil apropriado para a realização deste projecto.

7.4.4 Preparação dos Testes

A preparação dos testes de usabilidade com o *Sudoku Access* incidiram principalmente na construção de um ambiente controlado em que fosse possível efectuar gravações de vídeo para posterior análise. Os testes foram idealizados para serem aplicados a cada aluno sem a presença de outros utilizadores de modo a não perturbar a experiência. Os testes aplicados incidiram em tarefas que aprenderam ao longo de vários meses de utilização com a interface.

7.4.5 Realização dos Testes

A preparação dos testes é uma fase que requer muita atenção e dedicação, sendo uma etapa bastante burocrática quando é necessário envolver uma população que não se encontra directamente ao nosso alcance. Neste caso, houve a necessidade de pedir várias autorizações para poder realizar os testes de usabilidade. Tive de falar novamente com o Director da escola, a fim de pedir permissão para entregar autorizações aos pais da Fátima e do Jorge. No caso da Daniela o problema burocrático foi afastado porque recebi autorização verbal dos pais e o combinado foi que os testes seriam realizados na casa da Daniela com a presença de pelo menos um familiar. Assim sendo, avancei com os testes de usabilidade da Daniela enquanto aguardava pelas autorizações dos encarregados de educação dos restantes alunos. Tive que esperar cerca de 4 semanas antes de poder realizar actividades com a Fátima e o

Jorge.

A realização dos testes dividiu-se essencialmente em 3 fases fundamentais. A primeira fase consistiu na experimentação e exploração da interface, em que se observa quais as maiores dificuldades e os mecanismos que podem ser melhorados no *software*. Nesta etapa os alunos detectaram alguns problemas e dificuldades na interface, tais como:

- Complexidade nas grelhas do *Sudoku Access* 9x9;
- Dificuldades na interpretação das grelhas do puzzle por conter apenas números;
- No sistema de varrimento foi encontrado um problema quando se efectuam escolhas no *Sudoku Access* pressionando continuamente a tecla “Espaço”.

Esta fase teve uma duração de cerca de cinco meses, onde os alunos foram testando a interface e esta foi sendo alterada até chegar a uma versão estável. No final desta etapa os alunos foram convidados a responder a um pequeno questionário relativamente à facilidade de utilização e de aprendizagem, à velocidade, à agradabilidade, ao conforto e ao contentamento com a interface.

Na segunda fase do estudo, os alunos já revelavam alguma prática com o *Sudoku Access*, tendo sido então realizados testes com uma duração de 30 minutos, nos quais os alunos foram convidados a realizar 10 tarefas:

- Criar uma nova grelha no *Sudoku*;
- Preencher correctamente a primeira linha do *Sudoku*;
- Preencher correctamente a terceira coluna do *Sudoku*;
- Eliminar a jogada anterior;
- Modificar o grau de dificuldade do jogo para “médio”;
- Alterar as cores do varrimento/selecção do *Sudoku* para vermelho;

- Apagar todas as jogadas efectuadas no *Sudoku*;
- Verificar a solução do jogo do *Sudoku*;
- Alterar o tamanho do *Sudoku*;
- Fechar o jogo do *Sudoku*.

A última fase dos testes teve como objectivos analisar quantitativamente o tempo despendido pelos alunos para completar uma grelha 9x9 e registar o número de células preenchidas. Nesta etapa foi ainda quantificado o número de teclas pressionadas pela Fátima na interacção com a interface e foi também registado o número de palavras pronunciadas pelo Jorge. No decorrer deste estudo os alunos foram convidados a preencher um pequeno questionário acerca da satisfação com a interface.

7.4.5.1 Caso de Estudo da Participante Daniela

O primeiro contacto com a Daniela e a sua família foi extremamente gratificante. Foram muito amáveis e observei um grande carinho da família em todas as tarefas da Daniela ao longo do seu dia-a-dia. Inicialmente falei com o pai da Daniela e expliquei com que tipo de aplicação ela iria trabalhar. O pai ficou radiante por eu querer oferecer à filha um *software* que lhe fosse acessível por computador, tendo em consideração as suas graves limitações físicas. No primeiro dia em casa da Daniela fiquei realmente chocado e sentido ao observar os graves problemas físicos desta menina. Verifiquei que ela passava o dia inteiro numa cadeira adaptada, praticamente deitada, e que os pais ou o irmão a colocavam de frente a uma secretária que continha um computador. A utilização do computador só era acessível através de um teclado virtual e com o auxílio da família. Observei também que o computador possuía uma *webcam*, a qual era utilizada para observar as aulas em tempo real. O irmão confidenciou-me que em alguns momentos a Daniela tentou utilizar os *softwares Click-N-Type* e *REACH Interface Author 5*, mas quando a conheci, ela manuseava apenas o teclado virtual de acessibilidade da Microsoft. Infelizmente,

como ela própria comentou, “o teclado virtual da Microsoft não apresenta recursos suficientes para me ajudar quando necessito de introduzir palavras no computador”.

Quando falei com a Daniela e lhe expliquei o jogo do *Sudoku* ela ficou bastante entusiasmada por poder experimentar o *Sudoku Access*, uma vez que devido aos seus graves problemas físicos, esta tarefa tornava-se bastante complicada. Verifiquei que a Daniela só conseguia movimentar os olhos, falar e mover os dedos indicadores. A família da Daniela revelava dificuldades económicas e a obtenção de dispositivos de *hardware* ou *software* tornava-se bastante complicado. Deste modo utilizámos o próprio teclado da Daniela como dispositivo físico para introduzir informação no computador. Através do toque na tecla “Espaço” e utilizando o sistema de selecção automático do *Sudoku Access*, a Daniela foi capaz de interagir sozinha com o jogo do *Sudoku*. Foi neste primeiro contacto que realmente percebi que todo o meu esforço, ao longo destes anos, podia realmente ajudar indivíduos com deficiências físicas.

A Daniela compreendeu rapidamente a lógica do jogo do *Sudoku* e a própria família ficou entusiasmada participando diversas vezes no preenchimento de células de diversos puzzles. Em pouco tempo a Daniela foi capaz de preencher sozinha um *Sudoku* 9x9 e foi capaz de usar e configurar o *software* sem qualquer ajuda de um assistente. Ao longo dos jogos ela foi configurando o *Sudoku* à sua maneira, de modo a aumentar a sua acessibilidade, nomeadamente o nível de dificuldade, a velocidade de selecção automática, a cor da selecção e o tamanho do *Sudoku*. A sua interacção com o sistema de selecção para efectuar todas estas alterações foram muito bem conseguidas permitindo ganhar autonomia nas suas tarefas.

A Daniela revelou preferir jogar com uma grelha 9x9 e com números e cores (ver adiante). Ela obteve resultados surpreendentes e demonstrou grande entusiasmo. Passo a citar uma frase da Daniela: “o que eu apreciei de melhor no jogo foi o sistema de selecção automático porque deu-me a possibilidade de preencher a grelha do *Sudoku*”.

Com o passar dos meses apercebi-me que o estado de saúde da Daniela se estava a deteriorar. A Daniela ficou constipada, o que agravou a sua deficiência provocando

graves dificuldades respiratórias. Infelizmente, numa tarde de Domingo liguei à mãe da Daniela para confirmar a minha presença no dia seguinte e recebi a triste notícia de que a Daniela não tinha resistido. Foi uma fase extremamente complicada e apercebi-me que neste tipo de trabalho temos de estar preparados para vivenciar acontecimentos deste tipo e que, independentemente do nosso estudo, ficamos sempre com ligações fortes a estas pessoas. Nunca me esquecerei de uma frase vinda do pai da Daniela e que me comoveu bastante: “ela gostava tanto do jogo”.

7.4.5.2 Caso de Estudo da Participante Fátima

A Fátima é uma menina alegre e bastante motivada em sala de aula que revela graves limitações na comunicação oral e no desenvolvimento motor. Nas aulas, a Fátima depende do apoio dos professores e colegas para conseguir acompanhar o plano curricular. Para poder movimentar-se, utiliza uma cadeira de rodas eléctrica composta por um joystick adaptado às suas necessidades. Nas aulas, utiliza um computador portátil que lhe serve de caderno digital para guardar a informação disponibilizada pelos professores. Infelizmente, a sua interacção com computadores é difícil, o que representa algumas dificuldades no acompanhamento das aulas. Para poder conhecer melhor a aluna assisti a algumas aulas e apercebi-me rapidamente das dificuldades físicas reveladas por ela. Antes de lhe apresentar o *Sudoku Access* observei a sua acessibilidade com os computadores. A utilização de um rato convencional revelou-se ser possível mas com algumas restrições. Ela própria declarou não gostar de manipular um rato porque tem dificuldades em movimentar o cursor na tela do computador. A utilização de um teclado convencional é possível mas com grande lentidão na digitação de palavras. A Fátima apenas consegue movimentar um braço e com algumas dificuldades. Quando expliquei à aluna a lógica do jogo do *Sudoku*, verifiquei um pouco de apreensão da parte dela. Perguntei-lhe se ela utilizava algum sistema de teclado virtual para lhe ajudar na digitação de caracteres e ela respondeu que desconhecia tal aplicação.

A Fátima utilizou a tecla “Espaço” de um teclado convencional para activar o

sistema de selecção automático mas teve algumas dificuldades na compreensão do mecanismo. A Figura 7.7 mostra a Fátima a manusear o seu computador, utilizando a tecla “Espaço”, com uma única mão, para controlar o jogo do *Sudoku Access*.

Após diversas semanas de treino com o *Sudoku Access*, verificou-se que a Fátima apresentava não só problemas físicos como também cognitivos. A Fátima admitiu apresentar algumas dificuldades com a interpretação dos números e no preenchimento da grelha 9x9 do *Sudoku*. O puzzle 9x9 revelou ser demasiado complexo para ser concluído por ela.

Devido às dificuldades sentidas pela Fátima com a utilização do sistema de selecção automático resolveu-se experimentar o sistema de reconhecimento de voz do *Sudoku Access*. Infelizmente não tivemos sucesso devido às dificuldades de pronuncia da aluna. Tivemos algum receio que a Fátima perdesse motivação, mas pelo contrário, foi ela própria que nos motivou para implementarmos novas características no *Sudoku* a fim de colmatar as suas dificuldades. Consequentemente, decidiu-se desenvolver uma grelha de *Sudoku* mais simples (4x4) e acrescentar duas novas opções para resolver o problema da utilização de números.

Começou-se por utilizar cores em vez de números. Em cada célula colorida foi incluída um pequeno círculo de modo a diferenciar as células que poderiam ser alteradas. As células com círculos azuis podem ser alteradas no jogo a qualquer altura enquanto os círculos pretos não podem ser modificados. Foi implementada também a possibilidade de utilizar combinações de números e cores simultaneamente. Cada número colocado na respectiva célula corresponde a uma cor específica (ver Figura 7.8).

Estas alterações permitiram à Fátima progredir significativamente na compreensão do jogo e entender o funcionamento do *Sudoku Access*. A sua confiança aumentou e as suas limitações foram ultrapassadas através do uso do sistema de selecção automático com grelhas mais pequenas (4x4) e utilizando cores em vez de números.

Na segunda fase dos testes a Fátima conseguiu realizar as dez tarefas propostas



Figura 7.7: A Fátima, uma rapariga de treze anos com *Spina Bífida*, está a manusear o seu computador utilizando a tecla “Espaço” para controlar o jogo do *Sudoku Access*.

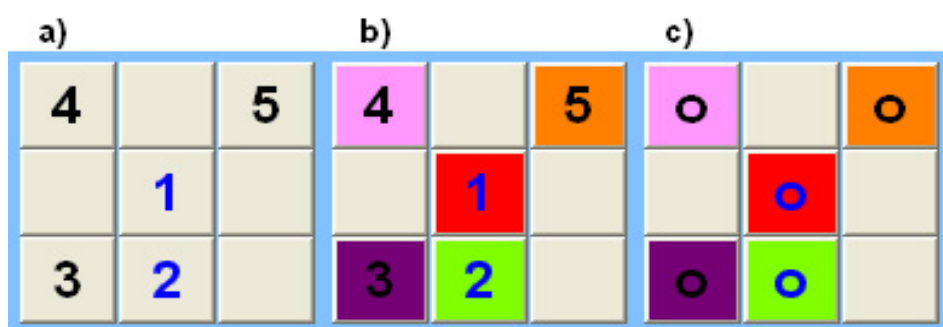


Figura 7.8: O *Sudoku Access* pode ser configurado para jogar: a) apenas com números, b) cores e números, c) apenas com cores.

por nós, apesar de revelar alguma lentidão e por vezes uma certa indecisão. No caso da Fátima, o puzzle utilizado nessas tarefas foi uma grelha 9x9 composta por números e cores. Ao longo dos testes observou-se que a Fátima revelava cada vez mais interesse em experimentar grelhas 9x9 compostas por números e cores. Na última fase dos testes quisemos determinar o tempo despendido pela Fátima para preencher todas as células de uma grelha 9x9 após ela ter visualizado a solução da grelha. A aluna demonstrou alguma lentidão mas conseguiu preencher as 55 células em 19 minutos e 27 segundos, pressionando a tecla “Espaço” 275 vezes. Todavia, a Fátima conseguiu completar rapidamente uma grelha 4x4 em 2 minutos e 5 segundos utilizando a tecla “Espaço” 50 vezes.

7.4.5.3 Caso de Estudo do Participante Jorge

O Jorge é um rapaz que não apresenta qualquer deficiência física ou cognitiva. A sua interação com computadores é muito boa, revelando um grande interesse em jogos de computador. Na escola o Jorge é considerado um bom aluno que não apresenta qualquer dificuldade em sala de aula. Nas aulas de informática revela grande capacidade de autonomia e de criatividade. Nos tempos livres o Jorge gosta de jogar diversos jogos incluindo o *Sudoku*. No centro de aprendizagem observei uma característica interessante quando o Jorge estava a tentar jogar um jogo de xadrez. Verifiquei que ele estava a tentar usar apenas o teclado para interagir com o jogo. Fiquei então a saber que em casa o Jorge não aprecia a utilização de um rato convencional quando interage com jogos de computador, preferindo utilizar um teclado comum em vez de usar um rato. Por vezes e em determinados jogos ele utiliza um *joystick*.

O Jorge teve conhecimento do nosso trabalho e mostrou grande interesse para nos ajudar. Como ele conhecia o jogo do *Sudoku*, lancei-lhe logo o desafio. Ao longo de três semanas ele deveria testar diferentes *Sudokus*, utilizando apenas um teclado convencional. Ao longo desta experiência o Jorge manipulou dez interfaces diferentes do *Sudoku* e concluiu que apenas conseguiu interagir na totalidade com

um *Sudoku* recorrendo ao teclado.

Durante os testes observou-se um certo descontentamento por parte do Jorge relativamente aos puzzles do *Sudoku*, nomeadamente por não conseguir utilizar o teclado para interagir com as grelhas e introduzir os números nas células. Analisando este facto, verificou-se que os jogos não estavam adaptados para serem manipulados sem a utilização de um rato convencional. A própria configuração de opções no *Sudoku* revelou os mesmos problemas. O Jorge teve então que utilizar um rato para manipular os puzzles. Com esta experiência o Jorge compreendeu a importância de existir interfaces adaptadas para auxiliar utilizadores que apresentem deficiências físicas.

Para obter algum feedback do Jorge relativamente ao *Sudoku Access* resolveu-se utilizar um microfone. Antes de usar o sistema de voz, o Jorge utilizou o sistema de treino de reconhecimento de voz da Microsoft para obter um melhor desempenho. Depois de ter concluído esta tarefa, o Jorge começou a experimentar o *Sudoku Access*. Rapidamente ele entendeu o modo de funcionamento da escolha de números e a sua introdução nas células do puzzle. Com alguma prática foi capaz de concluir com sucesso uma grelha completa do jogo. Foi então melhorando o seu desempenho, o que permitiu desta forma ser capaz de configurar opções do *Sudoku* apenas usando o sistema de voz. Conseguiu desta forma realizar sem qualquer problema as dez tarefas impostas na segunda fase deste estudo.

Na última fase dos testes decidiu-se avaliar o número total de algarismos/palavras pronunciados e o tempo dispendido para a introdução dos números nas células. Foi solicitado ao Jorge que completasse uma grelha 9x9 após ter visualizado a sua solução. O Jorge preencheu as 55 células em 7 minutos e 32 segundos, interagindo 330 vezes com o computador.

No final da experiência o Jorge ficou satisfeito por ter participado neste estudo e comentou: “com este jogo do *Sudoku* eu posso usar a minha voz para jogar e configurar todas as opções do puzzle apenas pronunciando números”.

7.4.6 Análise e Resultados

O envolvimento dos estudantes neste estudo demonstrou ser muito útil para a compreensão dos problemas e necessidades dos utilizadores com deficiências físicas para interagir com jogos educativos, nomeadamente com o jogo do *Sudoku*. Foi possível verificar que o jogo *Sudoku Access* contribuiu para gerar um grande entusiasmo e concentração na sala de aula. Através dos testes de usabilidade verificou-se que o *Sudoku Access* ofereceu aos utilizadores a possibilidade de controlar e manipular os puzzles do jogo, aumentando positivamente a acessibilidade, a velocidade e a performance de interacção com o jogo.

Ao longo dos testes com a Fátima foi possível constatar que o sistema de selecção automático pode gerar um tempo de espera prolongado para preencher uma célula na grelha do *Sudoku*. Porém, tanto a prática como a configuração do sistema de selecção permitem aumentar a velocidade do jogo. Outro aspecto relevante observado durante os testes de usabilidade foi a utilização de sons diferenciados no sistema de varrimento. O som revelou-se útil no jogo uma vez que surgiu como uma mais-valia no processo de selecção do item desejado (ou grupos de itens), privilegiando assim a tarefa do utilizador. De facto, o movimento e o som do sistema de selecção contribuíram para o aumento da concentração e do entusiasmo. Os utilizadores manusearam o método de selecção como se também ele fosse um jogo. Sem o auxílio deste sistema, pessoas como a Daniela ou a Fátima não teriam hipótese de jogar ao *Sudoku*.

Os utilizadores que conseguem utilizar o sistema de reconhecimento de voz podem potencialmente aumentar a velocidade de interacção com o *Sudoku Access*. O Jorge foi bem sucedido na utilização do sistema de voz mas para aperfeiçoar os seus resultados teve que usar o sistema de treino de reconhecimento de voz. Contudo, verificamos ser essencial utilizar o mesmo microfone que foi usado no processo de treino de reconhecimento de voz para preencher as grelhas do jogo. O sistema de reconhecimento de voz foi muito útil para preencher as células do *Sudoku* bem como para seleccionar as várias opções existentes no menu. No entanto, por vezes o *Speech*

Application Programming Interface (SAPI), não reconhece muito bem as palavras pronunciadas por utilizadores com deficiências físicas. Tentou-se minimizar este problema reduzindo a complexidade de articulação das palavras utilizando números e as palavras “Yes” e “No”, permitindo deste modo facilitar a interacção com a interface.

A tabela 7.1 apresenta-se como um sumário dos dispositivos utilizados pelos alunos, bem como do número de células preenchidas, número de teclas pressionadas, número de palavras pronunciadas e tempo despendido para completar uma grelha 9x9 do *Sudoku*. É de salientar que não foi possível obter resultados com a Daniela devido aos graves problemas relacionados com a sua doença.

Tabela 7.1: Resumo dos dispositivos utilizados pelos alunos, número de células preenchidas, número de teclas pressionadas, número de palavras pronunciadas e o tempo despendido para completar uma grelha 9x9 do *Sudoku*

Dispositivo	Estudante		
	Daniela	Fátima	Jorge
Nº. de Células preenchidas	55	55	55
Teclas pressionadas	-	275	-
Palavras pronunciadas	-	-	330
Tempo (seg)	-	1167	452

No final dos testes foi solicitado aos participantes que respondessem a um questionário relativo ao *Sudoku Access*, de forma a determinar o grau de satisfação dos alunos acerca da facilidade de utilização, facilidade de aprendizagem, velocidade, agradabilidade, conforto e contentamento da interface (ver Figura 7.9).

7.5 Discussão

A utilização de ambientes lúdicos proporcionou em sala de aula um fio condutor para permitir que os estudantes permanecessem entusiasmados e concentrados (Robertson & Howells, 2008). Os estudos realizados com o *Sudoku Access* tiveram

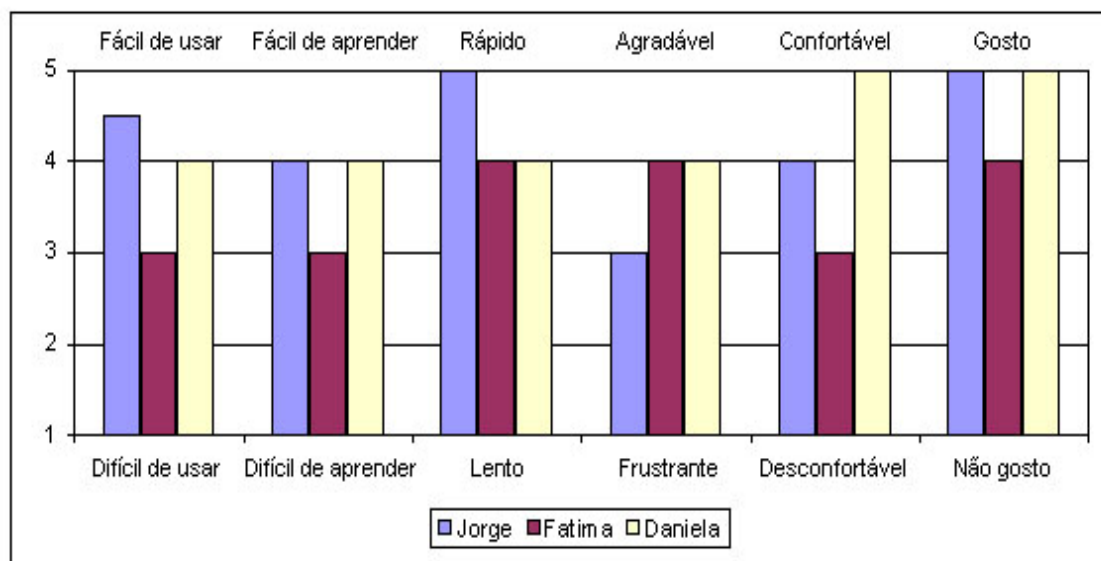


Figura 7.9: Resultados do questionário de satisfação realizado aos alunos relativamente à facilidade de utilização, facilidade de aprendizagem, velocidade, agradabilidade, conforto e contentamento com a interface *Sudoku Access*.

um papel muito importante para fortalecer as minhas experiências com utilizadores portadores de deficiências motoras. Posso mesmo afirmar que o pico mais alto da dissertação foi a realização dos testes com a Daniela. A aluna revelou-me que todo o meu esforço nessa área científica é claramente relevante e de grande valor para a qualidade de vida e integração de pessoas com deficiências motoras. Neste caso concreto, a Daniela só poderia interagir com o *Sudoku* utilizando o sistema de varrimento. O grau de deficiência motora era tão elevado que utilizar um rato ou um teclado era impossível. Os sistemas de varrimento foram muito bem manuseados pelos utilizadores. Foi muito gratificante ouvir os comentários e sugestões dos utilizadores com o objectivo de melhorar a interface. Os próprios pais tornaram-se muitas vezes participantes, a fim de conseguir construir uma aplicação que fosse possível de ser manuseada pelos filhos. Todo esse processo iterativo demora vários meses e requer muita observação, experimentação e recolha de opiniões. Neste estudo os dados quantitativos revelaram apenas o tempo dispendido para preencher uma grelha 9x9 do *Sudoku*. É obvio que esses dados não representam um valor fundamental nesta experiência, mas premeiam o facto de podermos melhorar a vida

das pessoas construindo aplicações adaptadas às suas necessidades.

7.6 Sumário

Neste capítulo ficou demonstrada a importância que os jogos educacionais podem ter para os estudantes, tendo a capacidade de melhorar a concentração e o pensamento lógico ao longo das actividades. Infelizmente, estudantes com deficiências físicas apresentam severas limitações no manuseamento de jogos computacionais. De modo a colmatar problemas de interacção em jogos educacionais foi desenvolvida uma nova interface para o jogo do *Sudoku* designada por *Sudoku Access*, que permite auxiliar indivíduos com deficiências físicas a manusear o jogo do *Sudoku*. Esta interface apresenta características especiais que possibilitam a sua utilização através de um sistema de voz ou de um sistema de selecção automática activado por um botão de toque.

Neste capítulo ficou evidenciado um estudo com o *Sudoku Access* realizado com três estudantes de uma escola básica do Algarve. Os estudantes que participaram no estudo apresentam idades compreendidas entre os 13 e 15 anos, sendo que dois deles revelam graves problemas físicos. O outro estudante não apresenta qualquer limitação física. Ao longo de vários meses de experiências com os alunos foram realizados diversos testes para averiguar as vantagens oferecidas pela interface. Diversos *Sudokus* foram implementados, testados e analisados, de forma a descobrir quais as maiores dificuldades que um utilizador pode sentir quando manipula um jogo de *Sudoku* sem o auxílio de um rato convencional. Foram também efectuados questionários aos alunos de forma a medir qualitativamente o grau de satisfação relativamente à facilidade de utilização, facilidade de aprendizagem, velocidade de manuseamento do puzzle, agradabilidade, conforto e contentamento com a interface.

8

Sumário, Trabalho Futuro e Conclusões

Este capítulo resume o trabalho contido na dissertação, revela uma lista de importantes contribuições, delinea investigações futuras, e finalmente termina revelando como o conhecimento na área das acessibilidades foi alterado com este trabalho de investigação.

8.1 Sumário

Na realidade da nossa sociedade não se pode negar que, apesar de algum esforço de inclusão por parte de diversas entidades e até mesmo por parte de nós enquanto indivíduos, as pessoas com deficiências físicas ainda vivem num mundo paralelo. Todos os indivíduos têm os mesmos direitos logo, palavras como “acessibilidade” nem deveriam existir porque deveriam ser um dado adquirido à nascença. No entanto, como não vivemos num mundo perfeito, as pessoas com deficiências físicas continuam a ser excluídas do seu ambiente e impedidas, desta forma, de usufruir de condições de vida normais. Os ambientes educativos não são excepção e os alunos com limitações físicas começam a sentir frustração por não conseguirem realizar todas as tarefas impostas no meio escolar.

Numa tentativa de atenuar este sentimento de rejeição, direccionou-se este estudo para a análise e o desenvolvimento de ferramentas educativas que auxiliem indivíduos com limitações ao nível da coordenação física. A ênfase deste trabalho concentra-se no desenvolvimento de um sistema informático que possibilite que pessoas com limitações físicas possam ganhar uma maior independência na implementação dos seus próprios programas de computador.

Esta dissertação começa por revelar os problemas de interacção Pessoa-Máquina em ambientes de aprendizagem, mais concretamente com pessoas com deficiências físicas. É salientada a importância que os ambientes virtuais podem ter no dia-a-dia destas pessoas, nomeadamente para aliviar parcialmente as limitações físicas em ambientes de aprendizagem computacionais. Inserida na aprendizagem computacional encontra-se a programação, actividade importante que abrange um grande número de estudantes na nossa sociedade. O processo de aprendizagem de linguagens de programação e as dificuldades encontradas pelos estudantes constituem temas recorrentes nas investigações da área das ciências educacionais da computação.

A linguagem de programação utilizada neste estudo é a linguagem *Logo*, linguagem esta que se baseia numa corrente construcionista que proporciona uma filosofia de educação caracterizada pela auto-aprendizagem e exploração do conhecimento. O *Logo* é uma evolução contínua de uma família de linguagens de programação que ajuda na concretização do processo educacional. É uma linguagem simples, flexível e poderosa, que proporciona um ambiente de aprendizagem interactivo e exploratório. Na filosofia do *Logo*, está implícita a ideia de que a aquisição do conhecimento não se dá em função do desenvolvimento do ser, mas sim pela forma como as pessoas interagem com o meio que os rodeia. Diversos testes foram realizados com duas turmas do secundário a fim de implementar a filosofia do *Logo* e compreender a importância desta ferramenta no meio educacional.

Posteriormente, desenvolveu-se uma ferramenta que possibilita indivíduos com limitações físicas utilizarem a programação *Logo* nas escolas. Ao longo de várias experiências foi possível implementar um método útil e simples para auxiliar indivi-

duos com limitações físicas a interagirem com o computador e a criação de projectos em *Logo*. Este método foi designado por sistema de selecção automático ou varrimento. Além deste sistema foi importante disponibilizar na interface um grande leque de opções configuráveis de modo a ser adaptável ao maior número possível de utilizadores. Ficou explícito que este método é essencial na área das acessibilidades, umas vezes que este tipo de filosofia pode ser adaptado a qualquer outro sistema virtual. Ainda é demonstrado outro exemplo onde este método de varrimento pode ser fundamental. Os *softwares* com características lúdicas podem ser uma mais valia no processo de aprendizagem educativo das crianças.

Os ambientes lúdicos computacionais podem oferecer aos alunos grandes vantagens de aprendizagem. O *Sudoku Access*, é um software lúdico que foi implementado especialmente para utilizadores com limitações físicas. Esta interface foi testada com alunos de uma escola básica e proporcionou grandes vantagens.

8.2 Trabalho Futuro

Existe um grande interesse no desenvolvimento de interfaces educacionais especialmente com a linguagem de programação em *Logo*. No futuro pretendemos melhorar o *LogoKeyboard* com novas opções, novas configurações e novos sistemas de varrimentos. Queremos continuar a desenvolver testes de usabilidade com o *LogoKeyboard* e interagir com um maior número de indivíduos com limitações físicas incluindo pessoas com paralisia cerebral. Pretendemos também implementar interfaces que permitam auxiliar indivíduos com graves limitações físicas em controlar rapidamente e facilmente os movimentos de um dispositivo de apontar em diferentes ambientes de programação *Logo*. Ambicionamos também em trabalhar com outras linguagens de programação para ajudar utilizadores em interagir com outras interfaces educacionais.

No futuro desejamos também explorar novas opções no *Sudoku Access* para permitir que indivíduos com problemas visuais possam usufruir do jogo do Sudoku.

Queremos também explorar novos meios de interacção Pessoa-Máquina para facilitar a interacção dos utilizadores, e melhorar a utilização dos sistemas de reconhecimento de voz com as nossas aplicações.

8.3 Conclusões

A ênfase deste trabalho, e consequentemente desta dissertação, assenta no desenvolvimento do sistema *LogoKeyboard* e do *Sudoku Access*, demonstrando o impacto que estas aplicações assumiram na qualidade de vida de pessoas com deficiências físicas. Infelizmente, na nossa sociedade as interfaces em ambientes de aprendizagem revelam severas limitações quando manuseadas por indivíduos com necessidades especiais. Torna-se importante criar condições de acessibilidade nas interfaces de modo a permitir a utilização do *software* ao maior número possível de utilizadores. Para alcançar este objectivo é fundamental alimentar novas mentalidades quando se constrói um novo *software*.

Esta dissertação revelou com clareza que diversas técnicas podem ser implementadas em interfaces para auxiliar pessoas com limitações físicas. O nosso trabalho centralizou-se essencialmente na implementação de uma interface direccionada para a linguagem de programação em *Logo*. O objectivo principal em utilizar a linguagem *Logo*, no lugar de uma outra linguagem de programação, esteve relacionado com as características da população em estudo. Em primeiro lugar, o *Logo* apresenta uma filosofia de educação construcionista fundamental para o desenvolvimento da criança como uma real auto-aprendizagem. Este aspecto torna-se fundamental quando se trabalha com indivíduos que apresentam limitações físicas porque estes revelam grande entusiasmo quando efectuam actividades por eles próprios. Qualquer tarefa que concluem sozinhos representa uma grande vitória. Segundo, as linguagens de programação e a sua utilização não são triviais. O *Logo* permitiu minimizar este factor por se revelar um mecanismo atractivo, simples e extremamente gráfico. Para além destes aspectos, possui a particularidade de permitir um contacto imediato com o computador revelando em tempo real a execução dos comandos introduzidos

pelo utilizador.

Através dos testes de usabilidade e de longas semanas de observação, concluímos que a utilização da linguagem *Logo* para determinados indivíduos com necessidades especiais só se tornou possível com a utilização do *software LogoKeyboard*. Esta ferramenta de programação *Logo* proporciona grandes vantagens para pessoas que não conseguem manipular com facilidade os periféricos de *input* de um computador. Os testes de usabilidade tornaram-se um processo fundamental para a construção da interface porque permitiram avaliar quais as dificuldades sentidas pelos utilizadores e desta forma melhorar o *LogoKeyboard*. Cada utilizador pode apresentar diversas limitações físicas o que dificulta a construção de um *software* para que seja acessível a um grande número de pessoas. É quase como se fosse necessário construir uma interface à medida de cada utilizador. Todo este processo torna-se bastante complicado porque não é fácil conseguir uma população que apresente limitações físicas e que tenha disponibilidade para interagir num projecto desta dimensão. Para além de ter que aprender a trabalhar com a interface, acrescenta-se outra dificuldade que é a de aprender a linguagem de programação *Logo*. O aspecto burocrático foi também um factor que dificultou e retardou este projecto devido à grande quantidade de autorizações que foi necessário adquirir das escolas e dos encarregados de educação. Apesar da minha insistência em querer obter uma população que fosse razoável para o meu estudo, posso afirmar que esse objectivo foi alcançado e foi sem dúvida gratificante para a concretização deste projecto.

A linguagem de programação pode por vezes assustar os estudantes que começam a interagir com os novos conceitos, por ser uma ciência que necessita de muito treino e dedicação. Qualquer processo de aprendizagem requer tempo e força de vontade para assimilar os conhecimentos e aplicá-los. É por este motivo que a aprendizagem da linguagem de programação *Logo* revela grande interesse por se enquadrar numa filosofia construcionista, onde a aquisição do conhecimento é realizada pela interacção das pessoas com o meio que as rodeia. Através do *Logo*, o utilizador aprende princípios, técnicas e habilidades que ajudam no processo de aprendizagem e na res-

olução de problemas. Esta linguagem foi concebida e direcionada para as crianças e ao longo de vários anos de estudo revelou ser uma ferramenta de excelência com utilizadores que apresentam deficiências físicas. O *Logo* permite a esses indivíduos autonomia e confiança quando estão a programar.

Todo este trabalho requer uma postura por parte do investigador de grande sensatez e paciência para lidar com populações que apresentam graves deficiências físicas. Muitas vezes não foi fácil interagir com estes indivíduos e observar pessoalmente as graves dificuldades sentidas por eles. Por vezes senti que eles próprios sentiam vergonha por parecer que estavam a dar trabalho. Foi fundamental a relação de proximidade que existiu entre o utilizador e o investigador. Todo esse processo demora tempo para que ambas as partes ganhem confiança e comecem a construir algo.

Coloco então novamente a mesma pergunta realizada no princípio da dissertação: “Será que uma pessoa com deficiência física consegue programar na linguagem *Logo*”? A resposta é afirmativa desde que o utilizador tenha capacidades cognitivas suficientes para compreender a linguagem. O tempo de aprendizagem pode não ser inferior ao de uma pessoa que não apresenta dificuldades físicas mas para isso é necessário que esta tenha as ferramentas necessárias para poder trabalhar.

Utilizando a mesma filosofia que o *LogoKeyboard* quisemos demonstrar que os ambientes lúdicos de aprendizagem podem ser ferramentas motivadoras em sala de aula. A criação do *Sudoku Access* permitiu reforçar a ideia que qualquer *software* que circule no mercado deveria apresentar características suficientes para ser manipulado por indivíduos que apresentam deficiências físicas. Através da implementação de um sistema de varrimento ou de um sistema de reconhecimento de voz seria possível ajudar uma grande parte da população que apresenta determinadas deficiências físicas e torná-las dessa forma mais autónomas.

Referências Bibliográficas

- Abelson, H. & Sessa, A. D. (1981). *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. MIT Press, Massachusetts.
- Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference? In *Constructivism: Uses and Perspectives in Education*. Geneva, Research Center in Education, 85–94.
- Alimisis, D. (2007). Teacher education to promote constructivist use of ict: Study of a logo-based project. In *Proceedings of the 11th European Logo Conference*. Bratislava, Slovakia.
- Apiola, M., Lattu, M., & Pasanen, T. A. (2010). Creativity and intrinsic motivation in computer science education: experimenting with robots. In *ITiCSE '10: Proceedings of the 15th International ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 199–203.
- Arden, N. K., Arden, E., & Hunter, D. (2008). *Osteoarthritis*. Oxford University Press Editions.
- Arnold, S. C., Mark, L., & Goldthwaite, J. (2000). Programming by voice, vocal-programming. In *ASSETS '00: Proceedings of the Fourth International ACM Conference on Assistive Technologies*. New York, NY, USA: ACM, 149–155.
- Asakawa, C. & Itoh, T. (1998). User interface of a home page reader. In *ASSETS '98: Proceedings of the Third International ACM Conference on Assistive Technologies*. New York, NY, USA: ACM Press, 149–156.

- Barnes, T., Richter, H., Powell, E., Chaffin, A., & Godwin, A. (2007). Game2learn: Building CS1 learning games for retention. In *ITiCSE '07: Proceedings of the 12th International ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM Press, 121–125.
- Beers, M. H. (2006). *The Merck Manual 18th Edition*. Centennial Edition.
- Beynon, M. & Roe, C. (2004). Computer support for constructionism in context. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*. Washington, DC, USA: IEEE Computer Society, 216–220.
- Blenkhorn, P., Evans, G., King, A., Kurniawan, S. H., & Sutcliffe, A. (2003). Screen magnifiers: Evolution and evaluation. *Proceedings of the 2003 IEEE International Conference on Computer Graphics and Applications Special Issue: Multimodal Interfaces*, 23(5), 54–61.
- Bray, M. B. A. & Green, T. (2004). *Technology and the Diverse Learner: A Guide to Classroom Practice*. Corwin Press.
- Clough, J. D. (2006). *Arthritis: A Cleveland Clinic Guide*. Cleveland Clinic Press Editions.
- Conati, C. & Zhao, X. (2004). Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. In *IUI-CADUI '04: Proceedings of the 9th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM Press, 6–13.
- Condado, P. A., Miquelina, P. F., Norte, S., Castilho, N., Lobo, F. G., & Shahbazkia, H. R. (2004). Information and communication technologies for people with disabilities. In *Interactive Computer Aided Learning International Conference*. Carinthia Technology Institute, Villach, Austria.
- Dai, L., Goldman, R., Sears, A., & Lozier, J. (2004). Speech-based cursor control: A study of grid-based solutions. In *ASSETS '04: Proceedings of the 6th International*

- ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM Press, 94–101.
- Darragh, J. J. & Witten, I. H. (1992). *The Reactive Keyboard*. New York, USA: Cambridge University Press.
- DiSessa, A. A. & Abelson, H. (1986). Boxer: a reconstructible computational medium. *Communications of the ACM*, 29(9), 859–868.
- Downes, T. (1993). Student teachers' experiences in using computers during teaching practice. *Journal of Computer Assisted Learning*, 9(1), 17–33.
- Dumas, J. S. & Redish, J. C. (1999). *A Pratical Guide to Usability Testing*. Intellect Books.
- Duvoisin, R. & Sage, J. (2001). *Parkinson's Disease: A Guide for Patient and Family*. Lippincott Williams, Wilkins Editions.
- Edwards, A. (1995). *Extra-Ordinary Human-Computer Interaction - Interfaces for Users with Disabilities*. New York, USA: Cambridge University Press.
- Emery, A. (2000). *Muscular Dystrophy: TheFacts*. Oxford University Press, USA.
- Fairweather, P. G., Hanson, V. L., Detweiler, S. R., & Schwerdtfeger, R. S. (2002). From assistive technology to a web accessibility service. In *ASSETS '02: Proceedings of the Fifth International ACM Conference on Assistive Technologies*. New York, NY, USA: ACM Press, 4–8.
- Fosnot, C. (Ed.) (1996). *Constructivism: Theory, Perspectives, and Practice*. New York, USA: Teachers College Press.
- Frietman, E. E., Bruggeman, F., & C.J., V. S. (1984). Eyeball-position-controlled communication system epcos. *Microprocessing and Microprogramming*, 14(3), 173–175.

- Ghedira, S., Pino, P., & Bourhis, G. (2009). Conception and experimentation of a communication device with adaptive scanning. *ACM Trans. Access. Comput.*, *1*, 14:1–14:23.
- Goldenberg, E. P. (1979). *Special Technology for Special Children*. Baltimor: University Park Press.
- Gorp, M. J. V. & Grissom, S. (2001). An empirical evaluation of using constructive classroom activities to teach introductory programming. *Computer Science Education*, *11*(3), 247–260.
- Govender, I. & Grayson, D. J. (2008). Pre-service and in-service teachers' experiences of learning to program in an object-oriented language. *Computer Education*, *51*(2), 874–885.
- Harel, I. & Papert, S. (1991). *Constructionism*. New York: Ablex Publishing.
- Hasselbring, T. S. & Glaser, C. H. W. (2000). Use of technology to help students with special needs. *The Future of Children*, *2*(2), 102–122.
- Hawking, S. (2004). *Computer Resources for People With Disabilities: A Guide to Assistive Technologies, Tools, and Resources for People of All Ages, 4th Edition*. Hunter House Publishers, CA.
- Hlauschek, W., Panek, P., & Zagler, W. L. (2009). Involvement of elderly citizens as potential end users of assistive technologies in the living lab schwechat. In *PETRA '09: Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM Press, 1–4.
- Hudak, P. (1989). Conception, evolution, and application of functional programming languages. *ACM Comput. Surv.*, *21*, 359–411.
- Ida, B. M. F. (2006). *Tetraplegia and Paraplegia: A Guide for Physiotherapists*. Churchill Livingstone; 6th Edition.

- Ingalls, D., Kaehler, T., Maloney, J., Wallace, S., & Kay, A. (1997). Back to the future: The story of squeak, a practical smalltalk written in itself. In *OOPSLA '97: Proceedings of the 12th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*. New York, NY, USA: ACM Press, 318–326.
- Jacko, J. & Sears, A. (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Lawrence Erlbaum Associates Editions.
- Jochemczyk, W., Kranas, W., & Oledzka, K. (2007). Logo in polish schools - cases. In *Proceedings of the 11th European Logo Conference*. Bratislava, Slovakia.
- Johnson, L. L., Rickel, J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11, 47–78.
- Jones, P. (1998). Virtual keyboard with scanning and augmented by prediction. In *ECDVRAT '98: Proceedings of Second European Conference on Disability, Virtual Reality and Associated Technologies*. Skövde, Sweden, 45–51.
- Kafai, Y. B. & Resnick, M. (Eds.) (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. New Jersey, USA: Lawrence Erlbaum.
- Kahn, D., Heynen, J., & Snuggs, G. (1999). Eye-controlled computing: The visionkey experience. In *Proceedings of the Fourteenth International Conference on Technology and Persons with Disabilities (CSUN '99)*.
- Kahn, K. (1996). Toontalk - an animated programming environment for children. *Journal of Visual Languages and Computing*, 7(2).
- Karimullah, A. S. & Sears, A. (2002). Speech-based cursor control. In *ASSETS '02: Proceedings of the Fifth International ACM Conference on Assistive Technologies*. New York, NY, USA: ACM Press, 178–185.

- Kate, J. T., Frietman, E. E., Willems, B., W. Ter Haar Romeny, & Tenkink, E. (1979). Eye-switch controlled communication aids. In *Proceedings of the 12th International Conference on Medical and Biological Engineering*. 19–20.
- Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41 (2), 75–86.
- Klopfer, E., Resnick, M., Maloney, J., Silverman, B., diSessa, A., Begel, A., & Hancock, C. (2004). Programming revisited: the educational value of computer programming. In *ICLS '04: Proceedings of the 6th International Conference on Learning Sciences*. International Society of the Learning Sciences, 16–18.
- Kumar, A. N. (2004). Three years of using robots in an artificial intelligence course: lessons learned. *Journal of Education Resources in Computing*, 4 (3), 1–15.
- Liffick, B. W. (2003). Assistive technology in computer science. In *ISICT '03: Proceedings of the First International Symposium on Information and Communication Technologies*. Trinity College Dublin, 46–51.
- Lin, Y. L., Wu, T. F., Chen, M. C., Yeh, Y. M., & Wang, H. P. (2008). Designing a scanning on-screen keyboard for people with severe motor disabilities. In *Proceedings of the 11th international conference on Computers Helping People with Special Needs*. Berlin, Heidelberg: Springer-Verlag, ICCHP '08, 1184–1187.
- MacKenzie, I. S. (2009). The one-key challenge: searching for a fast one-key text entry method. In *ASSETS '09: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM, 91–98.
- Majaranta, P. & Rähkä, K. J. (2002). Twenty years of eye typing: Systems and design issues. In *ETRA '02: Proceedings of the 2002 Symposium on Eye Tracking Research Applications*. New York, NY, USA: ACM Press, 15–22.

- Malik, M. A. (1998). Evolution of the high level programming languages: a critical perspective. *SIGPLAN Not.*, *33*, 72–80.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. New York, NY, USA: ACM Press, 367–371.
- Martin, F., Mikhak, B., Resnick, M., Silverman, B., & Berg, R. (2000a). To mindstorms and beyond: Evolution of a construction kit for magical machines. In *Druin, A. and Hendler, J., editors, Robots for Kids: Exploring New Technologies for Learning Experiences*. San Francisco, Morgan Kaufman, Academic Press.
- Martin, F., Mikhak, B., & Silverman, B. (2000b). Metacrickit: a designer’s kit for making computational devices. *IBM Systems Journal*, *39*(3-4), 795–815.
- Martin, F. G. (1988). *Children, cybernetics, and programmable turtles*. Master’s thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology.
- Meerbaum-Salant, O. & Hazzan, O. (2010). An agile constructionist mentoring methodology for software projects in the high school. *Trans. Comput. Educ.*, *9*, 21:1–21:29.
- Mihara, Y., Shibayama, E., & Takahashi, S. (2005). The migratory cursor: Accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations. In *ASSETS '05: Proceedings of the 7th International ACM Conference on Computers and Accessibility*. New York, NY, USA: ACM, 76–83.
- Mitrovic, A. & Suraweera, P. (2000). Evaluating an animated pedagogical agent. In *Intelligent Tutoring Systems*. 73–82.
- Moursund, D. (Ed.) (2006). *Introduction to Using Games in Education: A Guide for Teachers and Parents*. University of Oregon.

- Myers, B. A., Wobbrock, J. O., Yang, S., Yeung, B., Nichols, J., & Miller, R. (2002). Using handhelds to help people with motor impairments. In *ASSETS '02: Proceedings of the Fifth International ACM Conference on Assistive Technologies*. New York, NY, USA: ACM Press, 89–96.
- Nielsen, J. (1995). *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Nisbet, P., Gray, C., Taylor, G., & Townend, S. (1998). An introduction to switch access technology. In *Communication Matters National Symposium*.
- Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books.
- Norte, S., Castilho, N., Condado, P., & Lobo, F. G. (2005). Gogoboard and logo programming for helping people with disabilities. In *Proceedings of 10th European Logo Conference: Digital Tools for Lifelong Learning*. Warsaw, Poland, 171–178.
- Norte, S., Condado, P., & Lobo, F. G. (2007). Learning logo at a high school: Constructionism versus objectivism. In *Proceedings of the 11th European Logo Conference*. Bratislava, Slovakia.
- Norte, S. & Lobo, F. G. (2007). A virtual logo keyboard for people with motor disabilities. In *ITiCSE '07: Proceedings of the 12th International ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM Press, 111–115.
- Norte, S. & Lobo, F. G. (2008a). Sudoku access: A sudoku game for people with motor disabilities. In *ASSETS '08: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM Press, 161–168.
- Norte, S. & Lobo, F. G. (2008b). A sudoku game for people with motor impairments. In *ITiCSE '08: Proceedings of the 13th International ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 319–319.

- O'Hara, K. J. & Kay, J. S. (2003). Investigating open source software and educational robotics. *Journal of Computing Sciences in Colleges*, 18(3), 8–16.
- Paciello, M. (2000). *Web Accessibility for People with Disabilities*. Lawrence, KS: CMP Books.
- Pane, J. F., Myers, B. A., & Miller, L. B. (2002). Using HCI techniques to design a more usable programming system. *Proceedings of IEEE Symposium on Human Centric Computing Languages and Environments*, 198–206.
- Papert, S. (1972). Teaching children to be mathematicians vs. teaching about mathematics. *International Journal of Mathematics Education and Science Technology*.
- Papert, S. (1976). An evaluative study of modern technology in education. Logo Memo 26, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.
- Papert, S. (1993a). *The children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books.
- Papert, S. (1993b). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 2nd ed.
- Papert, S. (1996a). Computers in the classroom: Agents of change. *The Washington Post Education Review*.
- Papert, S. (1996b). *The Connected Family: Bridging the Digital Generation Gap*. Longstreet Press, Inc.
- Patterson-McNeill, H. & Binkerd, C. L. (2001). Resources for using lego mindstorms. In *Proceedings of the Seventh Annual Consortium for Computing in Small Colleges Central Plains Conference on the Journal of Computing*. USA: Consortium for Computing Sciences in Colleges, 48–55.
- Perlman, R. (1974). Tortis: Toddler's own recursive turtle interpreter system. Logo Memo 9, Massachusetts Institute of Technology, Logo Laboratory, Cambridge.

- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-Computer Interaction*. Addison-Wesley.
- R. Goldman, S. S. & Ju, T. (2004). Turtle geometry in computer graphics and computer aided design. *Computer Aided Design*, 36(4), 1471–1482.
- Repenning, A. (1993). *Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments*. Ph.D. thesis, University of Colorado at Boulder.
- Resnick, M. (1996). Distributed constructionism. In *Proceedings of the International Conference on the Learning Sciences, Association for the Advancement of Computing in Education*. Northwestern University.
- Resnick, M. (1997). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. MIT Press.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67.
- Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable bricks: Toys to think with. *IBM Systems Journal*, 35(3-4), 443–452.
- Resnick, M. & Ocko, S. (1991). Lego/logo: Learning through and about design. In *Constructionism*. Norwood, NJ, USA: Ablex Publishing.
- Resnick, M., Ocko, S., & Papert, S. (1988). Lego, logo, and design. *Children's Environments Quarterly*, 5(4), 14–18.
- Robbins, J. (2006). *Web Design in a Nutshell: A Desktop Quick Reference*. O'Reilly Editions.
- Robertson, J. & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Comput. Educ.*, 50, 559–578.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.

- Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design: beyond human-computer interaction*. New York, NY, USA: J. Wiley.
- Sheehan, R. J. (2005). *The Icicle Programming Environment for Children*. Ph.D. thesis, The University of Auckland.
- Shein, F. (1997). *Towards Task Transparency in Alternative Computer Access: Selection Of Text Through Switch-Based Scanning.* Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Toronto.
- Shein, F., Hamann, G., Brownlow, N., Treviranus, J., Milner, D., & Parnes, P. (1991). Wivik: A visual keyboard for windows 3.0. In *RESNA'91: Proceedings of the Annual Conference of the Rehabilitation Engineering Society of North America*. Arlington, VA, 160–162.
- Sipitakiat, A. & Blikstein, P. (2010). Think globally, build locally: A technological platform for low-cost, open-source, locally-assembled programmable bricks for education. In *TEI '10: Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*. New York, NY, USA: ACM Press, 231–232.
- Sipitakiat, A., Blikstein, P., & Cavallo, D. (2002). The Gogo board: Moving towards highly available computational tools in learning environments. In *Interactive Computer Aided Learning International Workshop*. Villach, Austria.
- Sipitakiat, A., Blikstein, P., & Cavallo, D. P. (2004). Gogo board: augmenting programmable bricks for economically challenged audiences. In *ICLS '04: Proceedings of the 6th International Conference on Learning Sciences*. International Society of the Learning Sciences, 481–488.
- Skinner, B. (Ed.) (1953). *Science and Human Behavior*. New York, USA: Free Press.
- Smith, D. C., Cypher, A., & Spohrer, J. (1994). Kidsim: Programming agents without a programming language. *Communications of the ACM*, 37(7), 54–67.

- Stager, G. S. (2007). Towards the construction of a language for describing the learning potential of computing activities. *Informatics in education*, 6, 429–442.
- Steriadis, C. E. & Constantinou, P. (2003). Designing human-computer interfaces for quadriplegic people. *ACM Transactions on Computer-Human Interaction*, 10(2), 87–118.
- Trewin, S. (2004). Automating accessibility: The dynamic keyboard. In *ASSETS '04: Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM Press, 71–78.
- Uri, W. (1999). Netlogo: Center for connected learning and computer-based modeling, Northwestern University. Evanston, IL.
- Valente, J. A. (1983). *Creating a Computer-Based Learning Environment For Physically Handicapped Children*. Ph.D. thesis, Massachusetts Institute of Technology.
- Valente, J. A. (2003). How logo has contributed to the understanding of the role of informatics in education and its relation to the learning process. *Informatics in education*, 2, 123–138.
- Watt, D. H. (1988). The computer as microworld and microworld maker. *Special Interest Group on Computer Uses in Education (ACM SIGCUE Outlook)*, 20(1), 81–89.
- Wilson, A. D. & Agrawala, M. (2006). Text entry using a dual joystick game controller. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 475–478.
- Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *Special Interest Group on Computer Science Education*. New York, NY, USA: ACM Press, 28(3), 17–22.
- Wobbrock, J. & Myers, B. (2006). Trackball text entry for people with motor impairments. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM Press, 479–488.

Zhai, S., Hunter, M., & Smith, B. A. (2000). The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*. New York, USA: ACM Press, 119–128.