



Universidade do Minho
Instituto de Educação

Miguel Carlos Pinto de Oliveira Marques

**O ensino da programação no
desenvolvimento de jogos através do
ambiente *Scratch***



Universidade do Minho
Instituto de Educação

Miguel Carlos Pinto de Oliveira Marques

**O ensino da programação no
desenvolvimento de jogos através do
ambiente *Scratch***

Relatório de Estágio
Mestrado em Ensino de Informática

Trabalho realizado sob a orientação do
Doutor José Alberto Lencastre

outubro de 2013

Nome: Miguel Carlos Pinto de Oliveira Marques

Endereço eletrónico: mcpom@sapo.pt

Número do Cartão de Cidadão: 10580269 7ZZ4

Título do Relatório: O ensino da programação no desenvolvimento de jogos através do ambiente *Scratch*

Supervisor: Doutor José Alberto Lencastre

Ano de conclusão: 2013

Designação do Mestrado: Mestrado em Ensino de Informática

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTE RELATÓRIO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, ___/___/___

Assinatura: _____

Para as minhas princesas

Agradecimentos

Gostaria de começar por agradecer à minha família, principalmente à minha mulher, pais e irmão, por toda a atenção e incentivo dispensados, à minha filha Lia Miguel a quem dedico este trabalho, por ser a minha fonte de inspiração e todos os dias me ensinar a arte de educar.

Ao meu orientador, Doutor José Alberto Lencastre, pelo seu apoio, disponibilidade, paciência e por toda a ajuda na orientação desta dissertação.

Aos meus colegas de Mestrado, principalmente ao Pedro Teixeira e Cristina Sousa, pelas suas sugestões pertinentes e construtivas ao trabalho realizado.

Ao professor cooperante, Dr. Fernando Filipe, por todo apoio e sugestões ao longo do meu período de estágio na Escola Secundária da Boa Nova em Leça da Palmeira.

Aos alunos do 12º ano das turmas de 'Ciências e Tecnologias' e 'Artes Visuais'. Sem a sua cooperação e entusiasmo não teria sido possível realizar este trabalho.

Resumo

O ensino da programação no desenvolvimento de jogos através do ambiente *Scratch*

Os jogos de computador desempenham um papel muito importante na vida dos nossos jovens (Murray, 1999; Turkle, 1997; Overmars, 2004; Maloney *et al.* 2008; Resnick *et al.*, 2009; Paraskeva *et al.*, 2010). O ambiente de programação *Scratch* permite criar facilmente jogos com potencial educativo. Este relatório apresenta um estudo sobre o ensino da programação no desenvolvimento de jogos através do ambiente de programação *Scratch*. O objectivo da minha intervenção foi ensinar todos os alunos a programar e a entenderam os conceitos de abstração e raciocínio lógico, condições básicas para a programação.

O estudo foi implementado numa turma do 12º ano do ensino secundário na disciplina de Aplicações Informáticas B e decorreu ao longo de vinte e cinco sessões de noventa minutos. Os vinte e seis alunos, numa primeira fase, desenvolveram no *Scratch* um jogo individual e numa segunda fase, em grupos de cinco e seis elementos, um jogo global para toda a turma intitulado *Zschool*. Este projeto global, além de envolver a referida turma B do 12º ano na programação do jogo, também contou com a cooperação da turma E do 12º ano de Artes Visuais que foi responsável pelo desenvolvimento de toda a componente multimédia: cenários, personagens e sons.

Suportada numa metodologia de *Development Research* (Lencastre, 2012; 2009; Coutinho & Chaves, 2001), são descritas neste relatório as fases de análise do contexto e definição de objectivos; a revisão de literatura e desenho do projeto; a implementação, atividades e estratégias desenvolvidas; e a avaliação, que envolveu a análise dos jogos individuais e os projetos de grupo, assim como uma entrevista coletiva do tipo *focus group* que procurou recolher as opiniões dos alunos sobre a minha intervenção.

Os resultados mostram que o *Scratch* permitiu motivar os alunos para a programação e desenvolver o raciocínio lógico/abstracto que a programação exige. Com este estudo consegui verificar que o desenvolvimento de jogos suportados pela ferramenta *Scratch* apresentam um elevado potencial pedagógico no ensino da programação, permitindo ensinar a programar de forma acessível e apelativa. Permitiu ainda promover a interdisciplinaridade entre a turma de Ciências e Tecnologias e a turma de Artes Visuais através de um projeto coletivo global.

Abstract

Programming teaching in game development through Scratch

Computer games play a very important role in the lives of our youth (Murray, 1999; Turkle, 1997; Overmars, 2004; Paraskeva *et al.*, 2010; Resnick *et al.*, 2009; Maloney *et al.* 2008). The programming environment Scratch lets you easily create games with educational potential. This paper presents a study on programming teaching in game development through the Scratch programming environment in an educational context. The purpose of my intervention was to teach all students to program and understand the concepts of abstraction and logical reasoning, basic conditions for programming.

The study was implemented in a class of high school 12th grade in the discipline of Computer Applications B and took place over twenty-five sessions of ninety minutes. The twenty-six students, initially developed an individual game in Scratch and in a second stage, in groups of five and six elements, one global game for the whole class titled Zschool. This global project, besides involving class 12 B also included the cooperation of the class 12 E, also from the 12th grade of Visual Arts, which was responsible for developing all the multimedia components: settings, characters and sounds.

Supported in a Development Research methodology (Lencastre, 2012; 2009; Coutinho & Chaves, 2001), are described in this report the context analysis and objectives definition phases, the literature review and project design, implementation, activities and strategies developed, and the evaluation, which involved the analysis of individual games and group projects created by the students, as well as a focus group conference type that sought to collect students opinions about my intervention.

The results show that Scratch allowed the motivation of students to programming and develop logical/abstract reasoning that programming requires. With this study we could find that the development of games supported by Scratch has high pedagogical potential in programming teaching, allowing teaching in an accessible and appealing way. It also allowed to promote interdisciplinary among the class of Science and Technology and the Visual Arts class through a overall collective project.

ÍNDICE

INTRODUÇÃO	17
Objectivos do estudo	20
Relevância do estudo	20
Organização do estudo	21
1 REVISÃO DA LITERATURA.....	23
1.1 Ensino da programação recorrendo aos Jogos	23
1.2 Desenvolvimento de jogos no ensino da programação.....	25
1.3 Ambiente de Programação <i>Scratch</i>	26
2 CONTEXTO E PLANO DE INTERVENÇÃO.....	31
2.1 Opção metodológica	31
2.1.1 Métodos e Técnicas de recolha dos dados.....	33
2.1.2 Caracterização da escola	34
2.1.3 Caracterização da turma.....	35
2.1.4 Fase de Desenho	37
2.1.5 Avaliação da usabilidade/divertimento do protótipo do jogo RPG desenvolvido em <i>Scratch</i>	39
2.1.5.1 Desenho do protótipo de um jogo RPG	40
2.1.5.2 Avaliação heurística	42
2.1.5.2.1 Objetivos da avaliação heurística	42
2.1.5.2.2 Métodos e técnicas de recolha de dados	42
2.1.5.2.3 Descrição	42
2.1.5.2.4 A amostra e o perfil da mesma.....	43
2.1.5.2.5 Recolha e tratamento dos dados	43
2.1.5.2.6 Síntese dos resultados e recomendações	46
2.1.5.3 Avaliação com utilizadores	47
2.1.5.3.1 Objetivos.....	47
2.1.5.3.2 Métodos e Técnicas de Recolha de Dados	47
2.1.5.3.3 Descrição do teste	48
2.1.5.3.4 A amostra e o perfil da mesma.....	49
2.1.5.3.5 Síntese dos resultados e recomendações	49
2.1.5.4 Conclusão	50
3 DESENVOLVIMENTO E AVALIAÇÃO DA INTERVENÇÃO	53
3.1 Fase de implementação.....	53
3.1.1 Primeira aula	54
3.1.2 Segunda aula.....	55
3.1.3 Terceira aula	58

3.1.4 Quarta aula (supervisionada).....	59
3.1.5 Quinta aula.....	61
3.1.6 Sexta aula (supervisionada).....	62
3.1.7 Sétima aula	65
3.1.8 Oitava aula	66
3.1.9 Nona aula à décima terceira aula.....	66
3.1.10 Décima quarta aula	68
3.1.11 Décima quinta aula.....	69
3.1.12 Décima sexta aula	70
3.1.13 Décima sétima aula à décima oitava aula.....	71
3.1.14 Décima nona aula (supervisionada).....	72
3.1.15 Vigésima à vigésima terceira aula.....	73
3.1.16 Vigésima quarta aula	74
3.1.17 Vigésima quinta aula.....	74
3.2 Fase de avaliação e apresentação de resultados	75
3.2.1 Avaliação e apresentação de resultados dos trabalhos individuais.....	75
3.2.1.1 Projeto <i>Harlem Shake</i>	76
3.2.1.2 Projeto Jogo de Carros.....	77
3.2.1.3 Projeto O Jogo Mais Difícil do Mundo	78
3.2.1.4 Projecto <i>Snake</i>	79
3.2.1.5 Projeto Jogo do <i>Noddy</i>	80
3.2.1.6 Projeto Pika Thunder.....	81
3.2.1.7 Projeto Jogo do Elementos.....	83
3.2.1.8 Projeto <i>Go Ask Alice</i>	84
3.2.1.9 Projeto <i>Snake Five</i>	85
3.2.1.10 Projeto <i>Titanic Beat Ups</i>	86
3.2.1.11 Projeto Jogo de Voleibol	87
3.2.1.12 Projeto <i>Tetris</i>	88
3.2.1.13 Projeto <i>Super Slime Soccer</i>	90
3.2.1.14 Projeto Tiro ao Alvo.....	91
3.2.1.15 Projeto <i>Portal</i>	92
3.2.1.16 Projeto Death Stars! Run!.....	93
3.2.1.17 Projeto Jogo de Carros.....	95
3.2.1.18 Projeto Super Space Mário.....	96
3.2.1.19 Projeto Jogo da Princesas	97
3.2.1.20 Projeto Portal Still Alive	98
3.2.1.21 Projeto Dragon Ball Arena	99
3.2.1.22 Projeto <i>Time Breaker</i>	100
3.2.1.23 Projeto Pokémon	101
3.2.1.24 Projeto <i>Dragon Quest</i>	103
3.2.1.25 Projeto RPG	104
3.2.1.26 Projeto Super-homem no espaço.....	105

3.2.2 Avaliação e apresentação dos resultados dos trabalhos de grupo.....	106
3.2.2.1 Grupo 1	106
3.2.2.2 Grupo 2	109
3.2.2.3 Grupo 3	111
3.2.2.4 Grupo 4	114
3.2.2.5 Grupo 5	118
3.2.3 Entrevista colectiva do tipo <i>Focus Group</i> final	120
4 CONCLUSÕES, LIMITAÇÕES E RECOMENDAÇÕES	125
4.1 Conclusões.....	125
4.2 Limitações.....	128
4.3 Recomendações	129
5 REFERÊNCIAS.....	131
6 ANEXOS	135
Anexo 1 – Planos de aula elaborados.....	136
Anexo 2 - Grelha de registo de avaliação heurística da usabilidade.....	164
Anexo 3 - Grelha de registo de avaliação heurística do divertimento	167
Anexo 5 – Guião do <i>Focus Group</i> inicial	168
Anexo 6 – Guião do <i>Focus Group</i> final.....	170
Anexo 7 – Guião do jogo <i>Zschool</i>	172

ÍNDICE DE FIGURAS

Figura 1 – Exemplo de um <i>script</i>	27
Figura 2 – Exemplo de projecto multimédia <i>Robot Dance</i>	28
Figura 3 - Metodologia de desenvolvimento (adaptado de Coutinho & Chaves, 2001)	32
Figura 4 – Sequência de passos no ensino tradicional da programação	37
Figura 5 – Método utilizado no ensino da programação	38
Figura 6 – Interface inicial do protótipo RPG.....	40
Figura 7 – Interface floresta de Du Waldenvarden	41
Figura 8 – Interface general Molgror	41
Figura 9 – Interface Jörmundur.....	41
Figura 10 - Interface do Projecto <i>Harlem Shake</i>	77
Figura 11 - Interface do Projecto Jogo de Carros	78

Figura 12 – Interface do Jogo mais difícil do Mundo.....	79
Figura 13 – Interface do Jogo <i>Snake</i>	80
Figura 14 – Interface do Jogo do <i>Noddy</i>	81
Figura 15 – Interface do Jogo <i>Pika Thunder</i>	82
Figura 16 – Interface do Jogo do Elementos.....	84
Figura 17 – Interface do jogo <i>Go ask Alice</i>	85
Figura 18 – Interface do Jogo <i>Snake Five</i>	86
Figura 19 – Interface do Jogo <i>Titanic Beat Ups</i>	87
Figura 20 – Interface do Jogo de Voleibol.....	88
Figura 21 – Interface do Jogo <i>Tetris</i>	89
Figura 22 – Interface do Jogo <i>Super Slime Soccer</i>	91
Figura 23 – Interface do Jogo Tiro ao Alvo.....	92
Figura 24 – Interface do Jogo <i>Portal</i>	93
Figura 25 – Interface do Jogo <i>Death Stars! Run!</i>	94
Figura 26 – Interface Jogo de Carros.....	95
Figura 27 – Interface do Jogo <i>Super Space Mario</i>	96
Figura 28 – Interface do Jogo das Princesas.....	97
Figura 29 – Interface do Jogo <i>Portal Still Alive</i>	99
Figura 30 – Interface do Jogo <i>Dragon Ball Arena</i>	100
Figura 31 – Interface do Jogo <i>Time Breaker</i>	101
Figura 32 - Interface do Jogo <i>Pokémon</i>	102
Figura 33 – Interface do Jogo <i>Dragon Quest</i>	103
Figura 34 – Interface do Jogo RPG.....	104
Figura 35 – Interface do Jogo Super-homem no espaço.....	105
Figura 36 – Cena da biblioteca.....	107
Figura 37 – <i>Scripts</i> do momento em que é retirada energia ao PP.....	108
Figura 38 – Cena da cantina.....	109
Figura 39 – <i>Scripts</i> do momento em que são lançadas bombas ao PP.....	110
Figura 40 – Cena do laboratório de química.....	112
Figura 41 - Cenas de sucesso e insucesso.....	112
Figura 42 – <i>Script</i> do algoritmo do jogo do laboratório de química.....	113
Figura 43 – Cena da introdução.....	115

Figura 44 – Cena da entrada da ESBN	115
Figura 45 – <i>Scripts</i> do movimento do personagem principal (PP)	116
Figura 46 – <i>Scripts</i> do momento em que se avançam as falas da entrada da ESBN	117
Figura 47 – Cena da passagem para a biblioteca	118
Figura 48 – <i>Scripts</i> do momento em que é retirada energia ao PP na cena 2	119

ÍNDICE DE TABELAS

Tabela 1 – Heurísticas de Nielsen adaptadas a jogos	45
Tabela 2 – Heurísticas de Malone adaptadas ao jogo RPG.....	45
Tabela 3 - Grupos de trabalho e cenas do guião atribuídas	69
Tabelas 4 e 5 – Conceitos de programação e estrutura do projecto <i>Harlem Shake</i>	76
Tabelas 6 e 7 – Conceitos de programação e estrutura do projecto Jogo de Carros.	77
Tabelas 8 e 9 – Conceitos de programação e estrutura do projecto Jogo mais difícil do Mundo.	78
Tabelas 10 e 11 – Conceitos de programação e estrutura do projeto <i>Snake</i>	79
Tabelas 12 e 13 – Conceitos de programação e estrutura do projecto Jogo do <i>Noddy</i>	81
Tabelas 14 e 15 – Conceitos de programação e estrutura do projecto <i>Pika Thunder</i>	82
Tabelas 16 e 17 – Conceitos de programação e estrutura do projecto Jogo dos Elementos.....	83
Tabelas 18 e 19 – Conceitos de programação e estrutura do projecto <i>Go Ask Alice</i>	84
Tabelas 20 e 21 – Conceitos de programação e estrutura do projecto <i>Go Ask Alice</i>	85
Tabelas 22 e 23 – Conceitos de programação e estrutura do projecto <i>Titanic Beat Ups</i>	87
Tabelas 24 e 25 – Conceitos de programação e estrutura do projecto Jogo de Voleibol.	88
Tabelas 26 e 27 – Conceitos de programação e estrutura do projecto <i>Tetris</i>	89
Tabelas 28 e 29 – Conceitos de programação e estrutura do projecto <i>Super Slime Soccer</i>	90
Tabelas 30 e 31 – Conceitos de programação e estrutura do projecto Tiro ao Alvo.	91
Tabelas 32 e 33 – Conceitos de programação e estrutura do projecto <i>Portal</i>	93
Tabelas 34 e 35 – Conceitos de programação e estrutura do projecto <i>Death Stars! Run!</i>	94
Tabelas 36 e 37 – Conceitos de programação e estrutura do projecto Jogo de Carros.	95
Tabelas 38 e 39 – Conceitos de programação e estrutura do projeto Jogo de Carros.....	96
Tabelas 40 e 41 – Conceitos de programação e estrutura do projecto Jogo das Princesas.....	97
Tabelas 42 e 43 – Conceitos de programação e estrutura do projeto <i>Portal Still Alive</i>	98

Tabelas 44 e 45 – Conceitos de programação e estrutura do projecto <i>Portal Still Alive</i>	100
Tabelas 46 e 47 – Conceitos de programação e estrutura do projeto <i>Time Breaker</i>	101
Tabelas 48 e 49 – Conceitos de programação e estrutura do projeto <i>Pokémon</i>	102
Tabelas 50 e 51 – Conceitos de programação e estrutura do projeto <i>Dragon Quest</i>	103
Tabelas 52 e 53 – Conceitos de programação e estrutura do projecto RPG.	104
Tabelas 54 e 55 – Conceitos de programação e estrutura do projeto Super-homem no espaço.	105
Tabela 56 – Categorias criadas a partir da entrevista do tipo <i>Focus Group</i> final	122

INTRODUÇÃO

A maior parte dos jovens adora jogar jogos de computador. Esta actividade é algo de muito importante nas suas vidas. Os jovens estão dispostos a despende longos períodos de tempo a melhorar os seus desempenhos nos jogos conseguindo manter longos períodos de atenção ao fazê-lo. Esta é uma razão clara para os educadores investigarem como podem ser utilizados os jogos na educação. Desta forma surgiram vários jogos educativos, a formação de uma subdisciplina de design de jogos, chamada *serious gaming*, e um número crescente de investigadores a estudar este fenómeno e suas implicações no sistema educacional (Prensky, 2001; Murray 1999; Turkle, 1997; Overmars, 2004; Paraskeva *et al.*, 2010).

Assistimos, constantemente, ao lançamento de novas consolas, novos jogos e um conjunto de aplicações que visam o entretenimento de crianças, jovens e adultos. Aliás, “*os jogos sempre estiveram presentes na história da humanidade, exercendo um papel especial no desenvolvimento das crianças bem como na forma de transmissão de valores culturais e conhecimento*” (Gomes, 2009, p.24).

Muitos professores utilizam, nos dias de hoje, jogos de computador dentro e fora da sala de aula “*enquanto instrumentos capazes de desenvolver funções cognitivas superiores, nomeadamente a capacidade de comunicação e colaboração, a resolução de problemas e a motivação para a realização de tarefas*” (Gomes, 2009, p.25). Uma aproximação é utilizar jogos lúdicos no contexto educacional, jogos tais como o *SimCity* que permitem aos seus utilizadores criar um entendimento dos sistemas económicos, *Super Monkey Ball*, pode ensinar às crianças certos princípios básicos da Física e *Roman Total War* pode dar uma visão histórica do Império Romano e sobre a forma como este se expandiu ao longo da Europa. No entanto, estes jogos não estão orientados para a prática lectiva, e não é fácil a sua utilização em sala de aula, pelo que alguns professores também utilizam os jogos para compensar os alunos pelo seu trabalho. Hoje em dia a maior parte das escolas estão equipadas com computadores com ligação à Internet e muitos professores reservam os últimos 10 minutos das aulas para os alunos irem para os computadores onde podem aceder à Internet e jogar jogos. No entanto à partida esta não parece ser uma boa estratégia.

A aproximação mais comum na utilização de jogos em sala de aula é a utilização de jogos educativos. Infelizmente, grande parte destes jogos são de fraca qualidade comparada com a dos jogos lúdicos, existindo vários factores para isto acontecer. O mais relevante refere-se ao

facto de os orçamentos para desenvolvimento de jogos educacionais serem muito mais reduzidos que os de desenvolvimento de jogos lúdicos (Overmars, 2004). Desta forma, as crianças ficam muito desiludidas quando jogam os jogos educacionais quando comparados com os jogos a que estão habituadas. O mais importante é o facto dos princípios dos jogos e os objectivos educacionais entrarem em conflito a maior parte das vezes. O objectivo primordial de um jogo educacional é colocar o controlo sobre aquilo que é aprendido nas mãos do Professor, enquanto, que nos jogos lúdicos o jogador é que deverá ter controlo sobre a acção do mesmo (Overmars, 2004). Os jogos lúdicos através do divertimento permitem aos seus jogadores explorarem determinados domínios, perceber mecanismos intrínsecos e efetuar escolhas de forma motivada, ao passo que os jogos educacionais encontram-se focados na aprendizagem de determinada tarefa ou habilidade sem fornecer uma motivação adequada dentro do mundo do jogo. Existe ainda um grande caminho a ser percorrido, até que o sistema educacional mude para que os jogos tenham um papel importante dentro deste sistema (Overmars, 2004).

Com este trabalho, abordei uma terceira utilização de jogos em contexto educacional. Em vez de utilizar jogos pretendi que os alunos criassem os seus próprios jogos. Ao criar os jogos, para além de estarem a desenvolver competências na área da programação, que foi o principal objectivo, desenvolveram competências que facilmente se podem relacionar com outras disciplinas da escola (a tão falada interdisciplinaridade que deve fazer parte de qualquer planificação). Para além disso, ao desenvolver um jogo sobre determinado matéria ou assunto, é uma excelente forma de motivar os alunos a dominar essa matéria ou assunto. O desenvolvimento de jogos pode ser usado em várias áreas desde o ensino do Português, passando pela Matemática ou Física. No entanto com este trabalho pretendi que os alunos desenvolvessem as suas competências na área da Programação. No âmbito do ensino da programação, Gomes *et al.* (2004) referem que existem professores que consideram a falta de empenho e dedicação dos alunos determinante para algum insucesso e professores que salientam a existência de alunos empenhados e dedicados, mas que não conseguem ser bem-sucedidos. França *et al.* (2010) referem que a falta de *abstração lógica* implica a pouca capacidade de *raciocínio lógico*, fundamental nesta área, desmotiva o aluno, além de diminuir a sua auto-estima. Mais referem que essa falta de motivação torna-se especialmente crítica quando o conteúdo é apresentado sem uma linguagem de programação que exemplifique a sua utilidade prática, tornando-se difícil a sua compreensão. Borges (2000) enfatiza a forma como alguns professores abordam esta matéria - exposição teórica do conteúdo, apresentação de

exemplos práticos, proposta de resolução de exercícios e proposta de elaboração de projetos – que em nada contribui para aumentar a motivação. Esta posição é reforçada por França *et al.* (2010) quando refere: “*podem-se amenizar esses problemas com melhorias pedagógicas e maneiras interativas e intuitivas ao ensino da programação por parte dos professores e dos alunos. Uma dessas formas pode ser através da utilização de ferramentas computacionais que o auxiliem nesse processo de aprendizagem*” (p. 4).

Para além da metodologia escolhida pelo professor, uma das características mais relevantes nesta área é o facto das linguagens de programação apresentarem sintaxes complexas, não havendo representação visual dos algoritmos construídos. Neste contexto, aproveitando o facto da Direção Geral de Educação (DGE), através da equipa de Recursos e Tecnologias Educativas (ERTE), propor como objetivo “*promover a utilização educativa do Scratch através do apoio, formação e partilha de experiências na comunidade educativa portuguesa*” (Educação, Direcção Geral, s/d), desenhei uma intervenção pedagógica em que utilizei o *Scratch* para contornar todas as dificuldades atrás referidas. O *Scratch* é uma nova linguagem gráfica de programação que possibilita a criação de histórias interativas, animações, simulações, jogos e músicas, e a partilha dessas criações na Web. Na página do ‘*Scratch | imagina, programa e partilha*’¹ podemos ler que o *Scratch* “*permite exercitar a criatividade e o raciocínio científico, lógico e matemático, ao disponibilizar ferramentas de programação informática simplificadas, com base nas quais podem ser desenvolvidos projetos*” (s/d). Por outro lado, podemos ainda ler que o *Scratch* permite que utilizadores “*desenvolvam competências de forma interativa e lúdica, constituindo por isso um poderoso contributo para o desenvolvimento educacional das novas gerações suportado no acesso a novas tecnologias*” (s/d). A utilização desta ferramenta surge como forma de alterar as metodologias tradicionalmente utilizadas para o ensino destes conteúdos e que se têm revelado insuficientes para a motivação e estímulo dos alunos na compreensão e aplicação de certos conceitos abstratos de programação.

¹ <http://kids.sapo.pt/scratch/about>

Objectivos do estudo

Com este estudo pretendi:

1. motivar os alunos para a programação informática através da linguagem de programação *Scratch*;
2. estimular os alunos a desenvolverem os seus próprios jogos a partir da linguagem de programação *Scratch*;
3. ensinar a programar através da linguagem de programação *Scratch*, desenvolvendo o raciocínio lógico/abstracto que a programação exige;
4. promover a interdisciplinaridade entre a turma de Tecnologias e a turma de Artes.

Relevância do estudo

Nos dias que correm existe uma grande procura por profissionais na área da Tecnologias de Informação (TI). No entanto as instituições de ensino, quer ao nível técnico quer superior, têm grande dificuldade em formar profissionais nesta área. Uma das principais razões pelo não ingresso ou dificuldade em concluir estes cursos deve-se à dificuldade na aprendizagem da programação e estruturas de dados, pois ambas exigem um raciocínio abstracto e lógico, grande capacidade de concentração e dedicação (Wiendenbeck, 2004; Gomes *et al.*, 2012; Ambrósio *et al.*, 2011). Também, segundo (Gomes *et al.*, 2008, p. 93), “*o ensino das linguagens de programação tem como propósito conseguir que os alunos desenvolvam as suas capacidades, adquirindo os conhecimentos básicos necessários para conceber programas capazes de resolver problemas reais simples*”, segundo estes autores a dificuldade em adquirir estas competências básicas estão na base das grandes taxas de insucesso nas disciplinas de programação dos cursos superiores nas área das TI.

Este estudo apresenta uma experiência de ensino de forma a motivar os alunos a desenvolverem os seus próprios jogos a partir da linguagem de programação *Scratch*, permitindo aos mesmos dominar as técnicas básicas de programação de uma forma agradável e produtiva. Os resultados obtidos com este estudo com uma turma do 12º ano, que desenvolveram jogos em *Scratch* de forma individual e em grupo de forma colaborativa através de uma metodologia de desenvolvimento, apontam de forma positiva num sentido promissor de motivar os alunos

para a programação. Pretende-se desta forma que os alunos que decidam enveredar por cursos superiores nas áreas das TI, quando deparados com as dificuldades das disciplinas de programação e estruturas de dados, consigam ultrapassar facilmente essas dificuldades reduzindo as taxas de reprovação e evasão deste cursos.

Organização do estudo

O presente relatório encontra-se dividido em quatro partes, antecedidas por esta introdução ao estudo, onde foram apresentados os objectivos, a relevância do estudo e a organização do mesmo. Numa primeira parte do trabalho denominada **revisão da literatura** é feita a contextualização teórica da aplicação de jogos no ensino da programação. São apresentados estudos de vários autores que defendem o desenvolvimento de jogos no ensino da programação, é feita uma apresentação do *Scratch* e seu potencial no ensino da programação.

Numa segunda parte do trabalho, **contexto e plano de intervenção**, é apresentada a metodologia de investigação utilizada: a *Development Research*. São descritas as fases de análise do contexto e definição de objectivos (Investigação Preliminar); a revisão de literatura e desenho do projeto (Embebimento Teórico).

Na terceira parte do trabalho, **desenvolvimento e avaliação da intervenção**, são descritas as aulas no âmbito do plano de intervenção sendo ainda feita a apresentação e análise dos resultados obtidos (Testagem Empírica), nomeadamente no que diz respeito aos jogos desenvolvidos de forma individual e em grupo.

Termina com a quarta parte, **conclusões, limitações e recomendações** (*Documentação, Análise e Reflexão sobre o Processo e Resultados*), procurando avaliar e refletir sobre todo o processo e sobre os resultados obtidos de modo a que possam conduzir a investigações futuras.

1 REVISÃO DA LITERATURA

1.1 Ensino da programação recorrendo aos Jogos

As chamadas habilidades do século XXI (National Research Council, 2012) cativam a escola para novos mundos, incluindo o dos jogos digitais que, segundo Prensky (2001) são, juntamente com o computador, a ferramenta de aprendizagem mais poderosa alguma vez conhecida.

Em conformidade com dados recolhidos pelo mesmo autor, os jovens de hoje passam mais de 10.000 horas a jogar jogos digitais. Porque não então aproveitar para incluir os jogos digitais no processo de ensino-aprendizagem, aliando lazer e saber?

O ser humano sempre demonstrou o seu fascínio por jogos. Existem achados arqueológicos de jogos com mais de 5.000 anos (Fox, 2002). As razões que nos levam a jogar jogos ainda não são claras nem explicadas com exactidão pela ciência, podendo ser várias, desde dinheiro, competição, diversão, entre outras. Nos dias de hoje, o mercado dos jogos de computador movimenta biliões de euros em todo o mundo, sendo considerado um mercado em plena expansão, não só apenas para o mercado de entretenimento como também para mercados específicos tais como os jogos educacionais.

As investigações sobre a relação dos jogos digitais com a aprendizagem começaram em meados da década de 80 (Alves, 2008). Destes estudos pode-se retirar duas importantes conclusões. Uma é que os jogos pedagógicos não despertam grande interesse na indústria dos jogos de computadores, sendo estes dependentes de financiamentos governamentais, o que *“tem levado à produção de jogos com baixa qualidade que buscam enfatizar conteúdos curriculares, sem se preocupar com a interface, com a qualidade das imagens, jogabilidade e interactividade”* (Alves, 2008, p.4). Assim, a maior parte das vezes os jogos pedagógicos não apresentam a mesma qualidade e interactividade que os jogos lúdicos, não conseguindo desta forma cativar a geração que tem vindo sendo denominada por diferentes rótulos (nativos digitais, geração Net, geração multitarefa, entre outros) (Prensky, 2001). A outra é que todos os jogos podem ser adaptados à educação, dependendo do objectivo que pretendemos atingir, *“qualquer jogo pode ser utilizado no espaço pedagógico não existindo uma dicotomia entre jogos electrónicos para entretenimento e jogos electrónicos para educação”* (Alves, 2008, p.4).

São vários os autores (Murray, 1999; Turkle, 1997; Overmars, 2004; Paraskeva *et al.*, 2010), que defendem o impacto positivo dos jogos digitais na educação, nomeadamente na forma como os alunos desenvolvem as capacidades de improvisação, raciocínio lógico, resolução de problemas e construção de novas realidades. Segundo Murray (1999), quando um adolescente está a jogar tem a possibilidade de criar cenários e caminhos imaginários, construindo narrativas próprias e indefinidas, imergindo em um universo de histórias autocriadas. Turkle (1997) afirma que os jogos electrónicos abrem novos caminhos aos estudantes, instaurando lógicas não lineares, desenvolvendo as capacidades cognitivas e emocionais. Segundo a autora, a interacção com os computadores facilita o “*pluralismo nos estilos de utilização. Oferecem coisas diferentes a pessoas diferentes; permitem o desenvolvimento de diversas culturas da computação*” (Turkle, 1997, p. 66).

Segundo esta linha de ideias mas ao mesmo tempo complementando-as, Pecchinenda (2003) afirma que os jogos electrónicos configuram-se em instrumentos para pensar, divertir, produzir ideias e representações da realidade e de nós mesmos e afirma que compreender a lógica dos jogos electrónicos significa compreender a cultura do computador como uma cultura de regras e sobretudo de simulação.

Todos os autores citados atrás concordam numa coisa: todos os jogos têm potencial pedagógico, podendo ou não ser inseridos na educação com os mais diversos objetivos, no entanto “*levar os jogos digitais para a escola porque seduzem os nossos alunos, sem uma interação prévia, sem a construção de sentidos, buscando enquadrar esse ou aquele jogo no conteúdo escolar a ser trabalhado, resultará em um grande fracasso e frustração por parte dos docentes e dos discentes*” (Alves, 2008, p.8). Ou seja, do meu ponto de vista usar os jogos apenas como recompensa de bom comportamento e/ou aproveitamento no final da aula poderá ser uma má estratégia. Pelo contrário, ter objetivos educativos bem definidos fará com que a utilização do jogo possa ser uma mais valia no processo de ensino e aprendizagem.

Piaget (1979) e Vygotsky (1978) são unânimes sobre a importância da utilização dos jogos no processo de ensino-aprendizagem. Segundo Piaget os jogos tem uma dupla função, consolidam as estruturas já formadas (aprendizagens adquiridas) e dão prazer e/ou equilíbrio emocional à criança. Piaget classifica os jogos em várias fases de acordo com as estruturas mentais das crianças. Vygotsky realça a influência do lúdico no desenvolvimento infantil, através dos jogos as crianças aprendem a agir, têm a curiosidade estimulada e adquirem iniciativa e autoconfiança, proporcionando o desenvolvimento da linguagem, do pensamento e da

concentração. Paraskeva *et al.* (2010), resumem as vantagens da utilização dos jogos em sala de aula:

- Aumenta a probabilidade de transferir os conceitos adquiridos.
- A prática conduz à consolidação automática de conceitos na memória. Permitindo assim, que o aluno se concentre em novas informações.
- A Aprendizagem pode ser alcançada através de tentativa e erro, onde o aluno continua a ser o tomador de decisão.
- Feedback imediato é recebido após cada acção, resultando em um ambiente que incentiva a exploração e experimentação.
- Os jogos podem motivar o aluno e aumentar a sua confiança.

Todos estes requisitos podem ser conseguidos com o auxílio do *Scratch*, razão suficiente para a minha abordagem. No entanto, segundo Vos *et al.* (2011) serão obtidos maiores benefícios se os alunos construírem os seus próprios jogos em vez de simplesmente jogá-los. Que é isso que fiz na minha intervenção pedagógica.

1.2 Desenvolvimento de jogos no ensino da programação

Jogar jogos de computador é uma actividade recreativa bastante popular entre os jovens. De forma não surpreendente, muitos destes jovens sonham em um dia criar eles próprios o seu jogo de computador. Assim, segundo alguns autores (Overmars, 2004; Silva & Martins, 2007; Leutenegger & Edgington, 2007; Depradine, 2011), porque não usar o desenvolvimento de jogos como uma ferramenta para ensinar aos nossos alunos programação? O desenvolvimento de jogos envolve vários aspectos das TI, incluindo computação gráfica, inteligência artificial, interacção homem-máquina, segurança, programação distribuída, simulação e engenharia de *software*. Ao mesmo tempo introduz aspectos das ciências liberais, sociais e psicologia.

Criar um jogo de computador comercial é uma tarefa muito árdua que envolve orçamentos da ordem dos muitos milhares de euros e uma equipa de especialistas na ordem dos quarenta elementos ou mais. No entanto podem ser criadas alternativas mais simples dentro da sala de aula com os alunos (Bethke, 2003). Escrever um jogo como o *Pac-Man* a partir do nada através de uma linguagem de programação moderna (C, C++, Java, entre outras), ainda é uma tarefa bastante difícil. Muitos programadores talentosos embarcaram em projectos deste tipo para no final verem as suas tentativas terminar em frustração. Felizmente, existem hoje em

dia várias ferramentas e ambientes de programação que tornam o desenvolvimento de jogos, uma tarefa mais fácil e acessível aos alunos que estão a dar os primeiros passos na programação (Overmars, 2004), de que é exemplo o *Scratch*. Estas aproximações focam-se nos conceitos básicos de programação, escondendo a complexidade inerente às linguagens de programação aplicadas no desenvolvimento de jogos. Este tipo de ambientes permite que as actividades de programação sejam efectuadas e que um *feedback* imediato seja recebido. Utilizando metáforas apropriadas, os alunos podem realizar mais facilmente as suas actividades movendo-se fora de um ambiente abstracto inerente à programação. Por exemplo, “The Programming Land MOOseum” utiliza um “museu” de metáforas que permitem aos estudantes explorarem um conjunto de conceitos básicos de programação (Slator et al, 2004). Outro exemplo é o sistema DELYS, desenvolvido por Dagdidelis et al (2003), que utiliza um ambiente simples de linguagem de programação que permite controlar os elementos de um micromundo. Inclusive muitos destes ambientes são utilizados para ensinar os conceitos básicos da programação a crianças com idades em torno dos seis anos, como podem demonstrar os estudos de Kahn (2001) e Kelleher e Paush (2005), que recorreram à ferramenta “ToonTalk”. Hoje em dia são utilizados sistemas mais avançados, tais como o Alice (Cooper, 2010), que permite desenvolver jogos 3D ou o Greenfoot (Kolling 2010), que tenta estabelecer uma relação mais próxima entre os conceitos de aproximação visual ou metafórica e a linguagem de programação subjacente. No entanto neste estudo vamos recorrer a uma ferramenta, que nos foi apresentada no primeiro ano do Mestrado em Ensino de Informática, na disciplina de Ensino de Programação, que nos permitiu um primeiro contacto com o *Scratch* e descobrir as potencialidades destes ambientes.

1.3 Ambiente de Programação *Scratch*

Quando os computadores pessoais foram introduzidos nas escolas dos Estados Unidos da América (EUA), nos finais da década de 70 e início de 80, houve um grande entusiasmo em ensinar às crianças como programar. Segundo Resnick *et al.* (2009), milhares de escolas ensinaram a milhões de alunos a escreverem programas simples em LOGO ou BASIC. Os estudos de Papert (1980) apresentaram o LOGO como a grande revolução para repensar a educação e aprendizagem. Apesar de alguns alunos e professores se sentirem bastante motivados por estas novas possibilidades, a maior parte das escolas direccionaram o uso dos

computadores para outras actividades. Segundo este autor, nos últimos 20 anos os computadores passaram a fazer parte integrante da vida dos nossos jovens, no entanto poucos aprenderam a programar. Hoje em dia, a maior parte das pessoas vêem a programação de computadores como algo muito limitativo, técnico e apropriado somente para um pequeno nicho da população. A razão do entusiasmo inicial ter desaparecido deveu-se a vários factores:

- As linguagens de programação iniciais eram muito difíceis de utilizar. A maior parte das crianças tinha dificuldade em dominar a sintaxe da programação.
- A programação era apresentada para realizar actividades, tais como gerar listas de números primos, ou efectuar desenhos simples, que não se relacionavam com os interesses dos jovens e experiências
- A programação muitas das vezes era introduzida em contextos em que não havia ninguém com conhecimentos suficientes para poder ajudar quando as coisas corriam mal ou encorajar e motivar para novos conhecimentos quando as coisas corriam bem.

Segundo Papert (1980), as linguagens de programação deveriam ser fáceis de começar a utilizar (*low floor*) e criar oportunidades para desenvolver projectos cada vez mais complexos ao longo do tempo (*high ceiling*). Para a equipa de Resnick (Resnick *et al.*, 2009), é também importante que as linguagens de programação suportem diferentes tipos de projectos (*wide walls*), para que pessoas com diferentes interesses e ritmos de aprendizagem se possam entrosar com o ambiente de programação. Encontrar linguagens de programação que satisfaçam estas três premissas não tem sido fácil, no entanto, para atingir estes objectivos foi criada a linguagem de programação *Scratch*.

O *Scratch* é uma linguagem de programação gráfica que permite a criação de histórias, animações, simulações, jogos e outros produtos multimédia com o recurso a uma técnica inovadora, onde os diferentes conjuntos de blocos de comandos coloridos são colocados de uma forma relacional (*script*), como se estivéssemos a construir um projecto com os blocos LEGO (Figura 1).

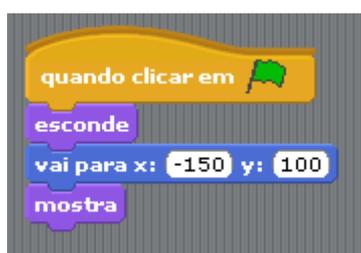


Figura 1 – Exemplo de um *script*

A programação com o *Scratch* assemelha-se às brincadeiras das crianças com Legos. À medida que vão fazendo as suas construções com os legos, a estrutura emergente vai fornecendo novas ideias, planos e objectivos. Esta forma de programar, além de ser mais fácil para o programador, é bastante motivadora uma vez que, de forma natural e sem grandes conhecimentos técnicos, o programador consegue desenvolver projectos com elevado grau de complexidade.

O design original do *Scratch* foi motivado pelos interesses e necessidades de jovens com idades entre os 8 e 16 anos, que frequentavam os centros de computadores de actividades de tempos livres, tais como o clube de computadores da Intel (Resnick *et al*, 2009). Podemos atribuir a sua criação e desenvolvimento ao *Lifelong Kindergarten Group* do Laboratório de Media do Massachusetts Institute of Technology (MIT) em colaboração com o grupo de investigação de Yasmin Kafai da Universidade da Califórnia (UCLA) (Maloney, 2008), tendo esta ferramenta sido apresentada em 15 de maio de 2007. O termo *Scratch* está intimamente relacionado com a técnica de *Scratching* usada pelos DJ's nomeadamente no Hip-Hop que giram com as mãos os discos de vinil de forma criativa. O próprio slogan da ferramenta (imagina, programa, partilha) pode ser mote para a sua apresentação. Desde logo, uma das suas características fundamentais é a capacidade que a ferramenta apresenta de mutação, ou seja, de criar qualquer tipo de produto multimédia sendo a nossa imaginação o limite, a Figura 2 apresenta um exemplo de uma criação multimédia, existente na biblioteca de exemplos do *Scratch*.



Figura 2 – Exemplo de projecto multimédia *Robot Dance*

Desta forma, para criar em *Scratch* é tão necessário ter algumas bases de programação como ter alguma dose de imaginação. Por isso, esta ferramenta tem vindo a ganhar relevância no ensino, nomeadamente no ensino das novas tecnologias. Outra das grandes características desta ferramenta é, sem dúvida, a possibilidade que os seus utilizadores têm de partilhar as suas criações na *Web*, o que a torna, por um lado muito atractiva e facilitadora de conhecimento, nomeadamente em contexto escolar e, por outro lado permite, sem grandes problemas, o trabalho colaborativo de uma forma muito eficaz.

No que se refere à faixa etária aconselhada para os utilizadores desta ferramenta, ela deve ser usada a partir dos oito anos embora existam algumas experiências de sucesso com crianças mais novas, Correia (2012), relata as suas experiências de utilização do *Scratch* com crianças entre os 4 e os 5 anos.

O *Scratch* enquanto *software* inovador assenta no facto de ser uma nova linguagem gráfica de programação que possibilita a criação de histórias interativas, animações, simulações, jogos e músicas, e a partilha dessas criações na *Web*, capaz de ser utilizado por jovens sem experiência de programação ou de utilização de interfaces sofisticadas. Para além disso, esta ferramenta, permite exercitar a criatividade e o raciocínio científico, lógico e matemático, ao disponibilizar ferramentas de programação informática simplificadas, com base nas quais podem ser desenvolvidos projetos. O *Scratch* é uma ferramenta de aprendizagem que permite que utilizadores a partir do primeiro ciclo desenvolvam competências de forma interativa e lúdica, constituindo por isso um poderoso contributo para o desenvolvimento educacional das novas gerações suportado no acesso a novas tecnologias.

2 CONTEXTO E PLANO DE INTERVENÇÃO

2.1 Opção metodológica

Seguidamente apresento o contexto e o plano de intervenção a partir da *Development Research* (metodologias de desenvolvimento - Coutinho & Chaves, 2001, p.895), enquanto opção metodológica. Esta opção foi decidida porque me permitiu desenhar uma intervenção no estágio iniciando o processo de uma forma mais global no âmbito da Unidade Curricular (UC) de Avaliação e Concepção de Materiais Escolares de Informática do Mestrado de Ensino de Informática da Universidade do Minho.

Fiz esta opção metodológica porque a *Development Research*, além de propor “*uma mistura de teoria e prática que se ajusta tão bem à investigação realizada em ambientes de aprendizagem onde se usam tecnologias*” (Coutinho & Chaves, 2001, p.898), valoriza o meu papel enquanto docente, investigador e *designer* no desenvolvimento do protótipo e considera a complexidade do contexto de aprendizagem (a sala de aula), as características do público-alvo ou o próprio meio de distribuição, ao contrário de outras abordagens de investigação que veem apenas as respostas finais, muitas vezes demasiado superficiais e tardias para serem úteis (Lencastre, 2012).

No contexto da problemática definida e de acordo com a metodologia de desenvolvimento, escolhi desenhar e desenvolver um protótipo de um jogo em *Scratch* no âmbito da UC de Avaliação e Concepção de Materiais Escolares de Informática do Mestrado que serviu de apoio às aulas no ensino presencial à disciplina de Aplicações Informáticas B, cenário da minha intervenção. Assim, este trabalho tem duas dimensões fundamentais: o desenho e desenvolvimento do protótipo, que foi também a ocasião de testagem da usabilidade, obtendo dados resultantes da aplicação de diversos instrumentos de recolha de informação; e a implementação do protótipo, que foi o momento de avaliar também os seus efeitos na população alvo deste estudo.

Apesar de sabermos que a *Development Research* tem várias vertentes utilizamos a interpretação que é dada por Lencastre (2009; 2012) e seguimos as etapas que o autor propõe, a saber: (i) *Investigação Preliminar* (análise do problema prático no terreno), (ii) *Embebimento Teórico* (desenvolvimento de uma solução dentro de um quadro teórico de referência), (iii) *Testagem Empírica* (avaliação e testes da solução no terreno) e (iv) *Documentação, Análise e*

Reflexão sobre o Processo e Resultados (documentação e reflexão sobre todo o processo). Através da análise da Figura 3 é possível verificar que “*a development research parte da análise de problemas práticos para o desenvolvimento de soluções no quadro de um referencial teórico, seguindo-se uma avaliação e teste de soluções no terreno para se fazer a documentação e reflexão que possam conduzir investigações futuras*” (Lencastre, 2012: p.53).

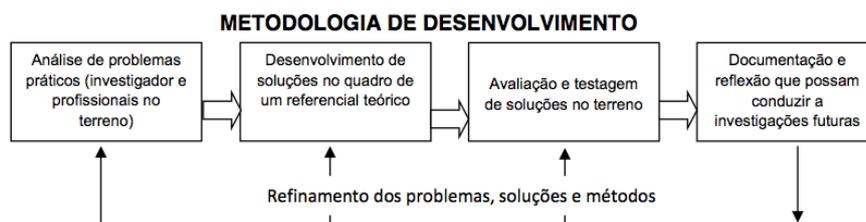


Figura 3 - Metodologia de desenvolvimento (adaptado de Coutinho & Chaves, 2001)

(i) Análise de problemas práticos - Lencastre (2012) refere que para se começar bem um projeto é necessário conhecer o público-alvo, com as suas expectativas e motivações mas também as suas limitações. Nesta primeira etapa, além de conhecer o contexto e os seus atores, dever-se-á fazer a revisão da literatura e pesquisar alguns estudos anteriores que permitam situar a nossa ideia de projeto. O autor indica, ainda, que nesta etapa é necessário ter em consideração as competências que se querem obter, as aptidões do público-alvo, os recursos que temos para serem utilizados, bem como identificar que tipo de atividades podem ser uma mais-valia para o público. Esta etapa do meu estudo coincidiu com o período de observação na escola onde a intervenção seria realizada. Recolhi informação através de documentos oficiais, da observação das aulas do professor cooperante, da análise e avaliação dos conteúdos produzidos pelos alunos e através da troca de opiniões com colegas, professor cooperante e orientador de estágio. A recolha destes dados permitiu-me, nomeadamente, caracterizar a escola e a turma. Fiz, também, uma entrevista coletiva do tipo *Focus Group* com o objetivo de conhecer melhor o público-alvo com quem iria trabalhar, as suas motivações, aptidões, características, competências e limitações.

(ii) Desenvolvimento de soluções no quadro de um referencial teórico. Na fase de embasamento teórico é necessário proceder a um esforço sistemático de articulação entre o “estado da arte” sobre a construção de jogos em geral e sobre o nosso jogo em particular. Há a necessidade de “*integração de todo o tipo de conhecimentos teóricos (comprovados e hipotéticos) e tecnológicos no sentido de se encontrarem soluções viáveis para a complexidade dos problemas em análise*” (Lencastre, 2012: 49). Refere o autor, que “*a concepção de uma solução para o problema em causa deve ser fundamentada quer do ponto de vista teórico quer do ponto de vista prático, e articulada com os objetivos de aprendizagem*” (Lencastre, 2012: 49-50). É fundamental que as nossas opções resultem, também, do conhecimento sobre outras experiências e de investigadores que analisaram a fundo as questões do design e das condições de usabilidade. Estes exemplos são boas ajudas e auxiliam-nos a atingir os objetivos. Nesta fase fiz uma revisão de literatura e o desenho da minha intervenção, em que o desenvolvimento do protótipo do jogo para o projeto foi determinante.

(iii) Avaliação e testes da solução no terreno). Este tópico teve dois momentos: os testes de usabilidade com peritos e com utilizadores do protótipo do jogo para o meu projeto e a minha intervenção propriamente dita com a turma.

(iv) Documentação e reflexão sobre todo o processo. Corresponde à redação de todo o relatório, onde descrevo e analiso de forma reflexiva todo o processo de intervenção.

Para a recolha e análise de dados, a *Development Research* utiliza os mesmos instrumentos – métodos e técnicas de recolha e análise de dados - das abordagens empíricas tradicionais da investigação quantitativa e mesmo qualitativa (Coutinho & Chaves, 2001). A diferença situa-se na forma como abordam os problemas e como se concebe o projeto da investigação em si.

2.1.1 Métodos e Técnicas de recolha dos dados

O delineamento metodológico do meu estudo envolveu a mobilização dos seguintes métodos e técnicas de recolha de dados:

- Observação direta aos alunos durante as aulas e na execução das tarefas e tirar notas para caracterização do público alvo aquando do momento de observação do projecto de estágio;
- Inquérito por entrevista coletiva tipo *Focus Group* inicial para conhecer melhor o público-alvo com quem iria trabalhar, as suas motivações, aptidões, características, competências e limitações;
- Testes de usabilidade para avaliação da usabilidade do protótipo (O método de observação, através do recurso à observação direta e ao Think Aloud; dois questionários: um de avaliação heurística e um de avaliação do divertimento);
- Registo de observação com tomada de notas aquando do momento de intervenção do projecto de estágio;
- Grelha com os critérios de avaliação dos projetos individuais e de grupo;
- Inquérito por entrevista coletiva tipo *Focus Group* final para aferir as percepções e opiniões dos alunos sobre o processo de intervenção.

2.1.2 Caracterização da escola

A escola onde desenvolvi o estudo está situada no concelho de Matosinhos. No que toca à análise do *Projecto Educativo da Escola* (Escola Secundária da Boa Nova, 2011, p.5), o mesmo identifica alguns problemas transversais ao agrupamento, que se refletem na turma onde vou intervir, a saber:

- *“o reduzido número de equipamentos informáticos e/ou outros de apoio às diferentes disciplinas;*
- *os incipientes hábitos de leitura são uma característica que emerge da análise da ocupação dos tempos livres dos alunos;*
- *fraco grau de participação dos Pais/EE nos contactos/reuniões proporcionados pela Escola;*
- *dificuldade em atrair e manter o público potencial”.*

2.1.3 Caracterização da turma

A turma escolhida para a minha intervenção é o 12º B – Ciências e Tecnologia. É uma turma com 26 alunos, 18 rapazes e 8 raparigas.

Da observação direta que fiz durante os meses de outubro e novembro (um total de nove aulas de 90 minutos), constatei que, de uma maneira geral, é um grupo de discentes educados e cumpridores das normas estabelecidas para a sala de aula. Da análise da ficha de diagnóstico, dada pelo docente titular da disciplina aos alunos no início do ano lectivo, constatei que existe uma percentagem de alunos (32%) que pretende aprender mais sobre programação informática e seguir os seus estudos nesta área da informática. Por outro lado, existe uma percentagem de alunos (21%) que pretendem seguir outra área de estudos e optaram por esta disciplina por forma a melhorar a sua média de acesso ao ensino superior. No que toca à programação em particular, 50% dos alunos já possuem conhecimentos prévios de programação em diferentes linguagens e 50 % dos alunos não possuem qualquer tipo de conhecimento prévio de programação.

No sentido de ir mais fundo no processo de análise da turma, nomeadamente em conhecer melhor o público-alvo com quem iria trabalhar, as suas motivações, aptidões, características, competências e limitações, promovi uma atividade exploratória coletiva do tipo *Focus Group* (Rubin & Chisnell, 2008, p. 17).

O *Focus Group* realizou-se em 21 de novembro de 2012, na Escola Secundária da Boa Nova, em dois momentos distintos: o primeiro com alunos da turma B – Ciências e Tecnologias - e o outro com alunos da turma E – Artes Visuais. Optou-se pela separação dos alunos por turma para que a presença de uns não influenciasse ou inibisse a resposta dos outros. Cada uma das entrevistas coletivas teve a duração de aproximadamente 45 minutos e para a sua realização foi necessário um gravador de áudio e um bloco de notas.

Na turma B, uma vez que o número total de alunos é de 26, pediu-se 12 voluntários para participação no *Focus Group* que se realizou numa sala independente, na presença de um moderador e de um moderador/assistente.

Na turma E participaram todos os alunos presentes nesse dia (11), tendo lugar na própria sala de aula onde se encontravam. Para além de um moderador e de um moderador/assistente, o professor orientador cooperante também esteve presente.

Em ambas as situações foram tidas em conta questões éticas relacionadas com o pedido de autorização para a gravação áudio das entrevistas (autorização previamente concedida). Antes do início da gravação os participantes foram ainda informados dos objetivos gerais da atividade, esclarecendo-se que o objetivo principal se prendia com a recolha generalizada de opiniões enquanto grupo/turma e não com opiniões particulares dos alunos. Salientou-se ainda o facto de o ênfase estar nas expetativas e motivações dos alunos face à disciplina em causa e na forma como ela tem vindo a ser abordada e não especificamente no professor titular da disciplina e na sua forma de atuação.

O guião das entrevistas tipo *Focus Group* foi elaborado com base nos seguintes objetivos:

- Avaliar as expetativas e representações dos alunos em relação aos conteúdos da disciplina de Aplicações Informáticas B (12^o ano) e às metodologias de ensino aprendizagem;
- Avaliar as perspetivas dos alunos para o futuro em termos da forma como pensam vir a atuar em contexto do ensino superior e de trabalho;
- Avaliar a perceção dos alunos de Artes sobre os de Ciências e Tecnologias e vice-versa;
- Avaliar o conhecimento e interesse dos alunos por determinadas ferramentas da Web 2.0;
- Avaliar a predisposição dos alunos para o trabalho colaborativo/cooperativo inter-turmas.

Da avaliação desta intervenção foi possível apurar que a maioria dos alunos pretende continuar a estudar e ingressar no ensino superior. Mais verifiquei que é notória a vontade dos alunos em trabalhar em projetos de índole mais prática.

As intervenções dos alunos e posterior análise de resultados permitiu ainda gerar ideias base para a conceção do protótipo do jogo RPG: os alunos do 12^o B mostraram maior interesse pela área da programação e os alunos do 12^o E pela área da multimédia; a maioria dos alunos mostrou-se disposta a desenvolver trabalho cooperativo/colaborativo, pois dizem perceber a sua importância para o seu futuro académico e profissional; quanto a um possível projeto a desenvolver de modo cooperativo/colaborativo, os alunos das duas turmas mostraram maior interesse pela criação de um jogo e a turma B considerou que o *Scratch* seria uma aplicação viável e interessante para esse efeito. Apenas dois alunos da turma E referenciaram o *Scratch*, uma vez que os outros não conheciam nenhuma ferramenta de desenvolvimento de software.

Nesse sentido, pensei em propor o desenho e desenvolvimento de um jogo, como forma de incentivo ao trabalho individual e inter turmas. Assim, pretendi que eles aprendessem a utilizar a ferramenta *Scratch*, que alia perfeitamente o plano educativo ao plano lúdico, sendo por isso, uma mais-valia na motivação destes. Com base nesta análise, consideramos que o desenvolvimento de um jogo RPG seria um trabalho viável de ser desenvolvido colaborativamente e cooperativamente e um projeto adaptado às necessidades e expetativas dos alunos. Este protótipo teve em consideração a faixa etária do público-alvo, bem como o seu perfil e as expetativas evidenciadas no *Focus Group*.

Por forma a auxiliar os alunos no que toca ao desenvolvimento do projeto final de grupo (Jogo), resolvi desenvolver um protótipo de um jogo em *Scratch* que permitiria mostrar todas as potencialidades da ferramenta e motivar os alunos para os conteúdos a serem leccionados aquando da minha intervenção. Pensei na altura que o protótipo por mim desenvolvido poderia ser um elemento importante na motivação dos alunos para o trabalho colaborativo.

2.1.4 Fase de Desenho

A maior parte das planificações de ensino de programação ao nível do ensino secundário apresentam os seus conteúdos com base numa linguagem de programação, normalmente o PASCAL, de forma estruturada. A Figura 4 Apresenta os passos típicos no ensino da programação dita tradicional.

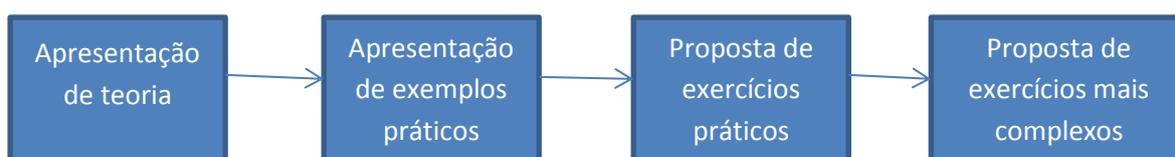


Figura 4 – Sequência de passos no ensino tradicional da programação

Este modo tradicional não consegue motivar para a programação aqueles alunos que têm mais dificuldades em entender os conceitos de abstracção exigidos pela programação, sendo essa falta de motivação especialmente crítica quando os alunos não conseguem visualizar os resultados imediatos dos seus programas. Como é possível motivar os alunos a desenvolver

programas em Pascal, quando o resultado final destes programas em nada se assemelha com aquilo que estes nativos digitais estão habituados a lidar no seu dia-a-dia?

Desenhei a minha proposta de intervenção tendo em mente resolver alguns dos inconvenientes da abordagem tradicional do ensino da programação, aproximando-me mais do construtivismo piagetiano, em que o saber não é apenas transmitido para o aluno, sendo o aluno levado a interagir com os objectos de aprendizagem (computador) de forma a desenvolver concepções e ideias diferenciadas.

No meu ponto de vista, esta abordagem, encontra-se mais de acordo com as necessidades de aprendizagem da sociedade actual, pelo que na primeira fase da minha intervenção pensei recorrer ao *Scratch* baseando-me numa nova proposta de ensino de programação que reordena as fases do ensino tradicional (Figura 4), apresentando na Figura 5 o método utilizado nesta primeira fase do trabalho.

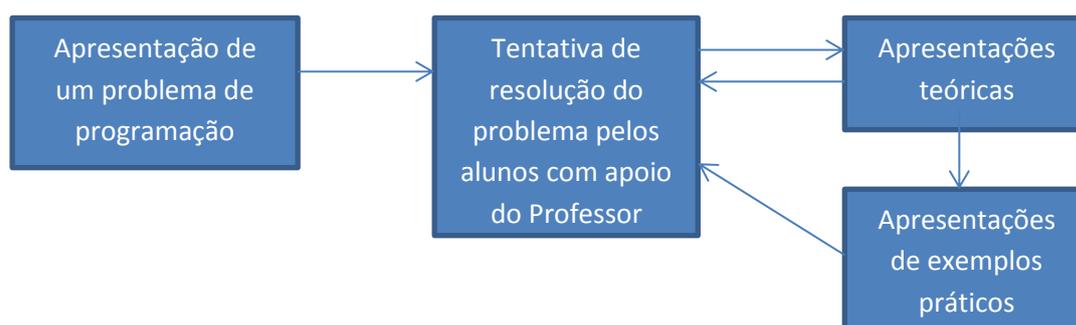


Figura 5 – Método utilizado no ensino da programação

Assim, decidi que ao longo das primeiras aulas de intervenção iria apresentar vários problemas de programação que os alunos resolveriam com base nos seus conhecimentos. À medida que os problemas fossem resolvidos, ou havia necessidade para novos conhecimentos, esses novos conteúdos seriam introduzidos de forma teórica e recorrendo a exemplos práticos, fechando-se o ciclo com uma nova oportunidade dos alunos resolverem o problema na posse dos novos conhecimentos. A minha preocupação seria identificar os alunos com maiores dificuldades, apoiando-os de forma individualizada. No final do tempo dado para a resolução de cada problema, a solução teria que ser apresentada, recorrendo sempre que possível a um exemplo resolvido por algum aluno de forma original e motivadora.

Para a segunda fase do meu desenho de intervenção, optei por desenvolver um jogo RPG em *Scratch* que elucidasse todas potencialidades desta ferramenta e motivasse os alunos para a realização de um trabalho semelhante. Este desenho da minha intervenção pareceu-me coerente com a *Development Research* do ponto de vista da dimensão investigativa do meu projeto. Esta metodologia tem como primordial objectivo contribuir de forma directa para a melhoria dos processos educativos, medindo o impacto desses contributos. No caso específico deste trabalho, a partir da construção de um jogo do tipo RPG, recolher informação de forma participativa que fundamente as escolhas que se vão realizando no processo de desenvolvimento desse jogo, criando condições para um feedback permanente que permita melhorar esse jogo e o processo educativo subjacente. Desta forma, toda a investigação orienta-se no sentido de obter indicações sobre a forma como diferentes aspectos do nosso problema podem ser resolvidos e antevistos (Lencastre 2012).

2.1.5 Avaliação da usabilidade/divertimento do protótipo do jogo RPG desenvolvido em *Scratch*

Como já apresentei anteriormente, são inúmeras as vantagens apontadas por vários autores relativamente ao potencial dos jogos no processo de ensino-aprendizagem, nomeadamente no processo de resolução de problemas inesperados e muitas vezes complexos (Menezes, 2003). Neste sentido, recorri à elaboração de um jogo RPG em *Scratch* que ilustrasse as potencialidades desta ferramenta e motivasse os alunos para a realização de um trabalho de grupo.

Sendo a usabilidade definida como sinónimo de funcionalidade do sistema para o utilizador (Lencastre & Chaves, 2007), o sucesso de qualquer aplicação informática depende significativamente da usabilidade que apresenta para o seu público-alvo. Especificamente no contexto educacional, torna-se imprescindível perceber, não só a usabilidade do ponto de vista da funcionalidade do sistema, mas também do ponto de vista da aprendizagem que poderá proporcionar.

É apresentado, seguidamente, o processo de construção do protótipo do jogo RPG desenvolvido no âmbito da disciplina de Avaliação e Concepção de Materiais Escolares de Informática do Mestrado de Ensino de Informática da Universidade do Minho em articulação com a minha intervenção no estágio.

2.1.5.1 Desenho do protótipo de um jogo RPG

O protótipo do jogo que desenvolvi é caracterizado por ser um RPG, “*uma das modalidades de jogos utilizados para fins educacionais*” (Bittencourt & Giraffa, 2003, p. 684). Os RPG são jogos cooperativos onde os jogadores “*assumem os papéis de personagens e criam narrativas*” (Brasil, 2009). O jogo desenrola-se de acordo com um sistema de regras preestabelecido, permitindo vivenciar um mundo imaginário, onde o jogador consegue improvisar de forma a ultrapassar os desafios.

O jogador encarna o papel da personagem principal do nosso RPG, o último dos druidas, tendo como objetivo recuperar o cálice mágico que foi roubado ao mestre dos druidas por Jörmundur (Figura 6).



Figura 6 – Interface inicial do protótipo RPG

Para cumprir a sua missão, o jogador deverá superar múltiplos desafios, nomeadamente: atravessar a floresta de Du Waldenvarden (Figura 7), fustigada pelos perigosos tornados; vencer Molgror o general dos Ra'zac (Figura 8), que detém a chave do castelo onde se encontra Jörmundur e no final o jogador tem de destruir Jörmundur (Figura 9) e recuperar o cálice mágico.



Figura 7 – Interface floresta de Du Waldenvarden



Figura 8 – Interface general Molgror



Figura 9 – Interface Jörmundur

Para completarmos esta missão o jogador tem a sua energia inicial com 10 unidades, que vai sendo consumida ao longo dos embates com os inimigos, e conta com a ajuda do mestre dos druidas que fornecerá poção de magia e escudo. O jogo termina se o jogador ficar sem energia ou se recuperar o cálice e o entregar ao mestre dos druidas, completando desta forma a missão.

2.1.5.2 Avaliação heurística

2.1.5.2.1 Objetivos da avaliação heurística

Antes do jogo ser utilizado foi necessário avaliarmos a sua usabilidade, de forma a serem detetados erros grosseiros e de índole mais técnica que dificultassem a sua utilização. Para esse efeito, recorremos a especialistas capazes de identificar problemas com referência aos princípios de usabilidade que não foram respeitados – as heurísticas (Nielsen, 1993). Visto tratar-se de um produto de carácter lúdico, através do qual é suposto o jogador apreciar o jogo e divertir-se ao aprender a jogar, para além das heurísticas relacionadas com questões de usabilidade, procurámos também identificar problemas ao nível do divertimento (Malone, 1982), que pudessem pôr em causa a satisfação do utilizador/jogador.

2.1.5.2.2 Métodos e técnicas de recolha de dados

Na avaliação heurística foram utilizados dois questionários: um de avaliação da usabilidade e um de avaliação do divertimento. O método de observação, através do recurso à observação direta e ao *Think Aloud* (van Someren *et al.*, 1994) onde a verbalização dos sujeitos enquanto realizam as tarefas fornece dados do que pensam e sentem, foi utilizado como complemento dos questionários, de forma a tornar “visíveis” feedbacks mais espontâneos, vividos em tempo real.

2.1.5.2.3 Descrição

Foram realizadas cinco sessões de testes com peritos distintos. A avaliação da usabilidade durou entre 40 a 60 minutos e a avaliação de divertimento cerca de 30 minutos.

Qualquer uma destas sessões teve início com a apresentação dos objetivos da avaliação, das heurísticas e das respetivas escalas de severidade. De seguida, atribuímos uma tarefa aos avaliadores: jogar o RPG e preencher o respetivo questionário de avaliação (usabilidade e/ou divertimento).

O *Think Aloud* teve a duração de cerca de 40 minutos, tendo sido realizado numa única sessão com um dos peritos. Teve início com a apresentação dos objetivos da avaliação e instruções acerca do desenvolvimento do teste, a que se seguiu a atribuição da tarefa ao avaliador: jogar o RPG e simultaneamente expressar verbalmente os seus pensamentos. Todos os procedimentos foram alvo de gravação áudio e de registos escritos.

2.1.5.2.4 A amostra e o perfil da mesma

Os testes foram realizados com 5 especialistas com uma média de idades de 35 anos, todos eles com formação académica (licenciatura) na área da informática e experiência profissional no ensino da informática e/ou no desenvolvimento/ teste de jogos. Exceção feita a um dos peritos, que possui o 12º ano de escolaridade e tem vasta experiência enquanto jogador de jogos digitais, tendo participado já em vários torneiros.

2.1.5.2.5 Recolha e tratamento dos dados

A construção do questionário de avaliação de usabilidade baseou-se nas dez heurísticas definidas por Nielsen (1993), cada uma delas contendo várias sub-heurísticas consideradas pertinentes para a avaliação do jogo (Tabela 1).

Heurísticas de Nielsen	
1- Visibilidade do estado do sistema	O sistema deve manter o utilizador informado sobre o que está a acontecer, através de feedback adequado; a resposta do sistema deve ser imediata (para tempos superiores a 10 seg o sistema deve possuir indicadores de progresso)
	O jogador é sempre capaz de identificar a sua pontuação/status no jogo? São utilizados sons para fornecer feedback significativo? A cada comando do jogador existe uma resposta do sistema?
2- Semelhança entre o sistema e o mundo real	A linguagem colocada no interface deve conter termos e conceitos familiares ao utilizador; devem-se seguir as orientações do mundo real, em que a informação apareça numa ordem natural e lógica

Heurísticas de Nielsen	
	<p>O jogo está de acordo com os padrões dos jogos digitais de forma a minimizar a curva de aprendizagem?</p> <p>O jogador é envolvido no jogo de forma fácil?</p> <p>O jogo utiliza linguagem familiar ao utilizador (palavras, frases, conceitos familiares)?</p>
3- Controlo e liberdade	<p>O utilizador deve poder desfazer e refazer ações, retornar ao estado anterior ou seguir adiante o utilizador; deve poder interromper processos e sair sempre que quiser</p>
	<p>Os movimentos do personagem são facilmente controláveis?</p> <p>É permitido sair do jogo em qualquer altura?</p> <p>É permitido pausar o jogo em qualquer altura?</p> <p>É fácil encontrar as opções disponíveis?</p>
4- Consistência e Padrões	<p>Devem-se seguir as convenções em termos de linguagem e ter o mesmo critério ao longo de todo o sistema; o utilizador não deve ter que pensar se palavras diferentes significam a mesma coisa</p>
	<p>A interface do jogo é consistente (controlo, cor, tipografia, menu e design de diálogo)?</p> <p>O método de movimentação do cursor e das teclas ao longo do jogo é consistente?</p>
5- Prevenção de erros	<p>Melhor que boas mensagens de erro é ter um projeto que previne a ocorrência de problemas; avisar quando processos de risco estiverem a ser realizados; todos os processos devem ser fáceis de serem concluídos</p>
	<p>O jogo previne erros antes de eles acontecerem, através de mensagens de aviso como "Tem certeza de que quer sair?"</p>
6- Reconhece ao invés de lembrar	<p>Devem-se tornar os objetos, ações e opções visíveis pois o utilizador não deve ter que (re)lembrar informação; instruções para o uso do sistema devem estar visíveis ou facilmente recuperáveis sempre que necessário</p>
	<p>As sugestões (mensagens) estão corretamente posicionadas?</p>
7- Flexibilidade e eficiência de utilização	<p>O sistema deve oferecer mais do que uma maneira do utilizador cumprir a tarefa; o sistema deve poder atender tanto a utilizadores novatos como experientes</p>
	<p>O interface é simples o suficiente para o jogador iniciado?</p> <p>O jogador avançado pode usar atalhos para melhorar o seu desempenho?</p>
8- Estética e desenho minimalista	<p>Os diálogos não devem conter informação irrelevante; textos concisos, demasiado texto e imagens distraem o utilizador layout limpo e simples</p>
	<p>O interface é visualmente agradável?</p> <p>O menu encaixa no estilo gráfico?</p> <p>O menu não interfere com a área onde ocorre o jogo?</p>
10- Ajuda e documentação	<p>O sistema deve poder ser usado sem documentação de ajuda; as informações necessárias para completar a tarefa devem estar disponíveis a cada momento; a ajuda deve ser fácil de aceder e focada na tarefa do utilizador, relacionar passos concretos a serem dados, não muito longa</p>
	<p>O jogo é apresentado por meio de um introdução interessante e esclarecedora?</p> <p>A função "ajuda" é visível?</p>

Heurísticas de Nielsen
<p>O interface de ajuda é consistente com os interfaces do protótipo que o suporta? A informação de ajuda é objetiva (o que eu posso fazer no jogo)? São fornecidos pequenos itens de ajuda no decorrer do jogo, como por exemplo, personagens que aparecem à medida que o jogo evolui, com dicas e informações dos próximos passos a seguir? Os utilizadores podem facilmente comutar entre a ajuda e o jogo e recomeçar o jogo onde tinham parado?</p>

Tabela 1 – Heurísticas de Nielsen adaptadas a jogos

Não incluímos, no entanto, a heurística número 9 (auxílio para o utilizador reconhecer, diagnosticar e recuperar dos erros), por, segundo Shneiderman (1997) não se aplicar aos jogos.

A cada especialista foi solicitado que indicasse as sub-heurísticas que não estavam a ser respeitadas, descrevendo o problema. Na descrição do problema, para além de um relato completo do mesmo, o especialista pôde apresentar soluções para o resolver. A cada problema identificado foi atribuído o respetivo grau de severidade, sendo 0 considerado sem importância e 4 catastrófico.

A construção do questionário de avaliação de divertimento foi feita de igual modo, baseando-se, no entanto, nas 8 heurísticas de Malone (1982), subdivididas em 3 categorias: desafio, fantasia e curiosidade (Tabela 2).

Heurísticas de Malone	
Desafio	
1	São claros os objetivos a serem atingidos em todas as etapas do jogo?
2	Existe informação, sempre que necessário, do quão distante estás de cumprir os objetivos?
3	O nível de dificuldade é desafiador ao longo do jogo?
Fantasia	
4	O jogo apela às tuas emoções?
5	Todos os elementos do jogo são fáceis de reconhecer? (escudo, a magia, as vidas...)
Curiosidade	
6	Compreendes, de forma direta e instantânea, como funcionam os elementos do jogo?
7	O jogo é divertido?

Tabela 2 – Heurísticas de Malone adaptadas ao jogo RPG

A implementação deste questionário seguiu os mesmos passos acima descritos, embora o grau de severidade estivesse escalonado entre 1 e 5, correspondendo 1 a muito pobre e 5 a muito bom.

Os dados recolhidos através dos questionários de avaliação heurística foram agrupados em diferentes temáticas e analisados comparativamente de acordo com as questões relacionadas com a avaliação da usabilidade ou divertimento. De seguida, foi feito um cruzamento da informação recolhida através dos questionários com os registos provenientes do método de observação direta/ *Think Aloud*.

2.1.5.2.6 Síntese dos resultados e recomendações

Através da avaliação heurística foi possível corrigir diversos problemas e obter várias sugestões para melhorar a usabilidade do jogo antes deste ser testado pelos utilizadores.

Em termos de usabilidade, os peritos identificaram problemas nas heurísticas n.ºs 1, 3, 5, 6, 7, 8, 10. Classificaram de problemas catastróficos (grau de severidade 4) alguns “bugs” que o jogo apresentava, assim como a falta de ajuda ao utilizador. Com o grau de severidade 1 e 2 (cosmético e simples) foram ainda identificados problemas que se prendiam com questões relacionadas com a interface do jogo e escassez de feedbacks sonoros.

Relativamente à avaliação de divertimento, os especialistas identificaram problemas nas heurísticas n.ºs 1, 2, 3, 4, 6, 7 e 8. Classificaram como muito pobre ou pobre (grau de severidade 1 e 2) questões relacionadas com a movimentação do personagem principal e problemas/bugs e falta de informações nalguns desafios. A escassez de efeitos sonoros foi também apontada como um problema, embora de grau de severidade 3 (ok).

O método de observação/ *Think Aloud* veio reforçar os problemas identificados na avaliação heurística e introduzir outros novos como, por exemplo, a facilidade excessiva do desafio final (matar o monstro) que poderia conduzir à desmotivação do jogador. Atendendo a que, segundo Kieras (2006), um jogo não será divertido a menos que possua algum tipo de desafio envolvido, algo difícil de aprender e solucionar, esta foi considerada uma informação valiosa.

Foram retificados, em conformidade com as sugestões propostas, todos os problemas identificados pelos peritos, à exceção daqueles que não iam ao encontro dos objetivos deste

protótipo cuja finalidade será servir de exemplo para um projeto que o público-alvo virá a conceber.

2.1.5.3 Avaliação com utilizadores

2.1.5.3.1 Objetivos

Após a retificação do protótipo, procedemos à realização de testes de avaliação com utilizadores semelhantes aos utilizadores finais.

Tal como referem vários autores, este é dos testes mais importantes, uma vez que permite obter informação direta e objetiva sobre como o jogador joga o jogo e que problemas enfrenta (Nokia, 2006; Laitinen, 2005; Nielsen, 1993).

Pretendemos, portanto, detetar problemas de usabilidade do jogo, do ponto de vista dos utilizadores, e perceber o nível de divertimento que o jogo lhes proporcionava.

2.1.5.3.2 Métodos e Técnicas de Recolha de Dados

Dada a especificidade do protótipo, para uma melhor observação e pelo interesse de poder haver interatividade com o participante, no sentido de perceber as suas impressões sobre o interface, o nível de envolvimento que o jogo proporciona e a jogabilidade, o teste decorreu com o moderador sentado ao lado do participante (*sit-by*) e sob a forma de “*play test*” (Rubin & Chisnell, 2008).

À semelhança das fases anteriores, recorreremos aos métodos de inquérito e de observação. No primeiro aplicamos a técnica de questionário, onde realizamos um questionário para analisar a usabilidade do jogo e outro para analisar o grau de divertimento proporcionado (o mesmo utilizado na avaliação com peritos). No segundo, utilizamos a técnica do *Software Logging* para registar toda a ação dos participantes no jogo, do *Think Aloud* para obter dados do que sentiam e pensavam enquanto jogavam e de observação direta para obter informação não verbalizada, tais como expressões faciais (Rubin & Chisnell, 2008; Nielsen, 1993).

2.1.5.3.3 Descrição do teste

Os testes decorreram numa sala reservada para o efeito com o equipamento necessário, um computador com o software *Scratch* e com um software de captura de imagem e som do computador e de som do microfone (SMRecorder).

Para que as avaliações fossem independentes entre si e não houvesse contaminações, os testes decorreram apenas na presença do monitor e do participante, tendo também havido o cuidado de não permitir que os participantes se encontrassem entre o término de um teste e o início do outro (Nielsen, 1990).

O moderador teve o papel de ir registando o comportamento e os comentários dos utilizadores, respondendo a alguma dúvida que surgisse, intervindo o menos possível. Tendo observado dúvidas comuns nos diferentes participantes, houve necessidade de antecipar alguma ajuda como as teclas para mover a personagem ou o avanço da introdução. (Nielsen, 1993; Rubin & Chisnell, 2008).

Segundo as orientações de Nielsen (1993) e de Rubin & Chisnell (2008), o teste decorreu em quatro fases distintas:

- Na preparação, foi verificada, horas antes, a funcionalidade do protótipo e do software utilizado para a técnica de *Software Logging* e o registo do *Think Aloud (SMRecorder)*, bem como a disponibilidade dos questionários necessários;
- Na introdução foi dado a cada sujeito o questionário de caracterização do seu perfil de utilizador (Dumas & Redish, 1999; Rubin & Chisnell, 2008) após o que se procedeu à explicação do objetivo do teste e dos procedimentos que lhe estavam associados como a gravação do que fosse dizendo e realizado no computador. Salvaguardaram-se aspetos éticos tais como: o anonimato do sujeito; a sua participação voluntária no teste; a informação de que o foco do teste consistia no protótipo e não no próprio jogador ou na sua performance.
- Na fase de teste pediu-se ao participante que procurasse chegar ao fim do jogo, sendo essa a sua principal tarefa, sem ter número mínimo ou máximo de tentativas, ultrapassando os diferentes obstáculos que lhe iam aparecendo, podendo parar quando pretendesse. Solicitou-se que fosse verbalizando os seus pensamentos relativamente a tudo o que achasse pertinente relativamente ao jogo. Quando o participante concluiu o

objetivo ou disse não querer continuar a jogar, foi pedido que preenchesse o questionário de usabilidade ou de divertimento.

- Na fase final, “*debriefing*”, passou-se um questionário *SUS (System Usability Scale)*, que mede o grau de satisfação do participante sobre o jogo, e agradeceu-se a sua participação no teste (Rubin & Chisnell, 2008; Dumas & Redish, 1999).

Os testes tiveram uma duração aproximada de 35 minutos.

2.1.5.3.4 A amostra e o perfil da mesma

À semelhança das características do público-alvo (Dumas & Redish, 1999; Nielsen, 1993; Rubin & Chisnell, 2008), os testes foram realizados com rapazes e raparigas, alunos de uma escola do Porto e outra de Coimbra, com uma média de 17 anos, num total de 8 participantes, como defendido por Rubin & Chisnell (2008). Todos têm conhecimentos de informática e acesso ao computador e à Internet. Metade costuma jogar jogos RPG e um participante não costuma jogar jogos digitais. A maioria joga diariamente, quatro passam menos de duas horas a jogar por semana, e três entre duas e quatro horas.

2.1.5.3.5 Síntese dos resultados e recomendações

Após análise dos dados recolhidos, foi possível retirar várias conclusões.

Quanto à usabilidade, todos os participantes acharam que todas as informações do jogo eram apresentadas de forma clara e objetiva, consideraram o cenário do jogo bem estruturado, mas mostraram não gostar dos controlos da personagem. Neste sentido, quatro participantes admitiram ter tido dificuldade em coordenar a personagem. Cinco acharam que o jogo era fácil de aprender porém três desistiram antes de conseguir chegar ao final do jogo. Três participantes acharam que o jogo nem sempre era atrativo. Foram ainda detetados alguns “*bugs*” no jogo.

Todos os participantes foram unânimes no que diz respeito à primeira impressão e à impressão final tendo-a classificado de bom.

No que diz respeito ao divertimento, traduzido através das heurísticas de Malone, aquela que obteve pior classificação foi a nº 6, tendo sido apontado como problema o não

funcionamento do escudo provocando frustração no utilizador. De seguida, as heurísticas nº 2 e 5 obtiveram a classificação “pobre” devido ao facto do utilizador não saber o quão distante está do objetivo final e da energia contida pela personagem principal não ser clara. As heurísticas nº 3 e 7 foram classificadas com “bom”. Foi também possível constatar que a energia que o utilizador dispõe é, na maioria dos casos, insuficiente para terminar o jogo, causando alguma frustração no utilizador.

Percebeu-se que alguns problemas de divertimento identificados prendem-se muito a questões de usabilidade, o que nos leva a concluir que a usabilidade está intrinsecamente ligada ao divertimento (Gurgel *et al.* 2006; Federoff, 2002; Kieras, 2006; Laitinen, 2005).

Adicionalmente ao que já foi referido, no que diz respeito aos dados obtidos da observação direta, do *Software Logging* e do *Think Aloud* percebeu-se que a média de tentativas a que os participantes recorriam era de 4; todos os utilizadores demonstraram dificuldades na passagem dos remoinhos, a maioria demorou algum tempo a iniciar o jogo demonstrando dificuldade na aprendizagem dos comandos; o tempo médio para um participante concluir o objetivo do jogo era de 5 minutos e 30 segundos.

O grau médio de satisfação obtido através do questionário *SUS* foi de 76 numa escala de 0 a 100.

No que concerne a recomendações, os participantes sugeriram que a opção de saltar a introdução fosse para um ponto mais avançado da mesma de modo a agilizar o recomeço do jogo; que as teclas fossem alteradas para outras mais usuais nos jogos e próximas entre si; que houvesse uma barra de energia da personagem e outra que permitisse perceber quanto faltava para o fim; que houvesse hipótese da personagem conseguir mais energia.

2.1.5.4 Conclusão

Na era do acesso facilitado a todo o tipo de recursos, a diversificação de materiais pedagógicos é uma estratégia imprescindível no processo de ensino-aprendizagem. Motivar os alunos implica pensar novos conceitos pedagógicos, novas metodologias e novos recursos. Contudo, não importa apenas que os recursos sejam uma novidade mas que proporcionem aprendizagem.

O estudo de desenvolvimento do protótipo permitiu verificar a relevância das avaliações de usabilidade na evolução de um qualquer protótipo. Para além da avaliação da usabilidade,

verificou-se que a avaliação do divertimento trouxe significativas melhorias ao protótipo na medida em que nos permitiu, enquanto equipa de desenvolvimento, entender de que forma é que as características e funcionalidades do jogo podem afetar o nível de diversão e, conseqüentemente, de satisfação no utilizador.

No final foram identificados na avaliação com os utilizadores ainda alguns erros de usabilidade. Por isso, o protótipo foi alvo de nova intervenção para correção dos erros detetados, de forma a ser utilizado como ferramenta pedagógica no âmbito da minha intervenção de estágio.

3 DESENVOLVIMENTO E AVALIAÇÃO DA INTERVENÇÃO

3.1 Fase de implementação

O principal objectivo da minha intervenção foi ensinar todos os alunos da turma do 12º B a programar. Para isso utilizei uma abordagem diferente da do professor orientador cooperante: em vez de usar a linguagem de programação Pascal usei o *Scratch*. O meu objectivo foi que todos os alunos aprendessem a programar e que entendessem os conceitos de abstracção e raciocínio lógico, condições básicas para a programação (França *et al.* 2010). Nesta turma, durante a fase de observação ao longo do primeiro período, consegui identificar um grupo grande de alunos que já se encontrava completamente à vontade com a programação. No entanto existiam alunos que não conseguiam programar de forma autónoma. Ao introduzir o *Scratch* pretendi manter os alunos, com bons conhecimentos, igualmente motivados e motivar para a programação aqueles alunos que têm mais dificuldades e, talvez por isso, mais desmotivados.

Esta fase do trabalho teve início no segundo período e terminou no final do terceiro período em Junho. Como já referi atrás no ponto 2.1.3 a implementação do projeto dividiu-se em duas partes, numa primeira parte ensinei aos alunos o ambiente de programação *Scratch* e no final pedi aos mesmo que desenvolvessem um jogo individual que demonstrasse a aquisição desses conhecimentos. Numa segunda parte, que teve início no terceiro período, foi pedido aos alunos que em grupo desenvolvessem um jogo recorrendo ao ambiente de programação *Scratch*. O protótipo do jogo RPG por mim desenvolvido serviria de fonte de informação sobre a programação e de recurso quanto às potencialidades do *Scratch*. De forma colaborativa os alunos trabalharam em grupo num jogo do tipo RPG, que foi desenvolvido em articulação com a turma do 12º E. Os alunos da turma E foram responsáveis pelo desenvolvimento de cenários, sons, *sprites* e guião do jogo. A turma B foi responsável pela parte da programação do RPG, a partir de todos os materiais multimédia desenvolvidos pela turma E.

O horário da turma do 12º B, à disciplina de aplicações informáticas B, era às terças e quintas-feiras das 10:00 às 11:30, no entanto devido ao facto de ter de assistir às aulas da parte curricular do Mestrado às terças-feiras de manhã a minha intervenção nas nove primeiras aulas apenas se fazia às quintas-feiras de oito em oito dias, as aulas de terça-feira eram dadas pelo professor orientador cooperante. Todas as quartas-feiras reunia com o professor orientador

cooperante, para saber como tinha corrido a aula de terça-feira e planificar em conjunto a aula seguinte de quinta-feira. A partir do início do terceiro período já passei a intervir às terças e quintas-feiras. O facto de nas primeiras nove aulas apenas estar de oito em oito dias com os alunos, constituiu uma das limitações ao meu estudo, pois apesar de ter conseguido um bom entrosamento com os alunos, não me era possível estabelecer uma relação professor-aluno de igual cumplicidade à do professor orientador cooperante, que estava sempre presente.

3.1.1 Primeira aula

Nesta primeira aula de implementação do meu projecto, iniciei a aula com uma proposta bastante ousada, os alunos desenvolverem o seu primeiro jogo no *Scratch*. Após uma breve explicação dos objectivos do programa e suas potencialidades, pedi aos alunos que com o auxílio de um pequeno guião (disponível em [aula 1](#)), criassem os seus projectos de uma versão simplificada do jogo do *Pac-man*. Adoptei esta estratégia de forma a motivar os alunos e mostrar como é fácil e excitante programar no *Scratch*. A grande vantagem do *Scratch*, é que os alunos podem verificar de imediato o resultado das suas instruções de código, algo que não acontecia antes com o PASCAL, neste ambiente de programação os alunos primeiro tinham de escrever todo o programa e só no final após a compilação sem erros é que era possível verificar o resultado final do seu trabalho.

Como se tratava da primeira aula, e a maior parte dos alunos teve pela primeira vez contacto com o *Scratch* (na fase de observação durante a entrevista coletiva do tipo de *Focus Group* apenas dois alunos mencionaram ter tido contacto prévio como o *Scratch*), utilizei o método de instrução directa e execução por tarefas para exemplificar alguns passos no *Scratch*, nomeadamente a criação de *sprites* e palcos, ao mesmo tempo os alunos iam resolvendo o exercício no lugar. Em seguida, desloquei-me por entre os alunos dando apoio individualizado sempre que necessário. À medida que se ia aproximando o final da aula notei que alguns alunos já estavam bastante adiantados e a maioria estava bem lançada no desenvolvimento do jogo do *Pac-man*, no entanto a conclusão do mesmo teria de ser feita na aula seguinte de terça-feira, dada pelo professor orientador cooperante.

No final da aula, fiz uma breve síntese e informei os alunos que teriam de concluir o jogo na próxima aula. Comuniquei ainda que estavam disponíveis na plataforma *Moodle*, vários tutoriais sobre o *Scratch* que lhes poderiam ser úteis para a conclusão do jogo do *Pac-man*.

Na aula seguinte, leccionada pelo professor orientador cooperante, os alunos concluíram os projectos e enviaram-nos para a plataforma *Moodle*, de forma a eu poder avaliar os trabalhos. Na reunião de quarta-feira com o professor orientador cooperante, este informou-me que todos os alunos conseguiram concluir os projectos e que todos enviaram os trabalhos para a plataforma, no entanto uma vez que só temos doze computadores para 26 alunos e apesar de 5 alunos trazerem os seus portáteis para as aulas, existem 8 grupos de alunos de dois que trabalham sempre em grupo, pelo que é difícil concluir se ambos os alunos conseguiram desenvolver as competências pretendidas com a tarefa proposta. Mais tarde em casa analisei um por um todos os trabalhos desenvolvidos pelos alunos, de um total de 26 alunos, obtive 10 trabalhos individuais e 8 trabalhos de grupo de dois alunos. Constatei que todos os trabalhos tinham conseguido atingir os requisitos mínimos tendo os trabalhos de dois alunos superado esses requisitos através da implementação das melhorias propostas, nomeadamente a introdução de variáveis para as vidas e pontuação, inimigos espalhados pelo labirinto e bónus que apareciam de forma aleatória. Na turma havia um grupo de alunos que elevava bastante a fasquia, adoravam programar e sentiam-se bastante motivados para o fazer, investindo largas horas no desenvolvimento das duas versões do *Pac-man* melhoradas.

3.1.2 Segunda aula

Comecei a segunda aula por fazer a interligação com os conteúdos da aula anterior, apesar de todos os alunos terem conseguido desenvolver pelo menos uma versão simplificada do jogo do *Scratch*, e pelo facto de os alunos que tinham mais dificuldades na área da programação durante o primeiro período, terem trabalhado em grupos de dois alunos, dediquei os 15 minutos iniciais para esclarecer algumas dúvidas e incertezas que os alunos possam ter tido no desenvolvimento dos seus trabalhos. Noto que os alunos se mostram bastante interessados e chamam os colegas para verem as suas versões do jogo. De forma a ultrapassar a dificuldade em perceber se todos os alunos estão a conseguir acompanhar os trabalhos, peço aos alunos que tinham trabalhado em grupos de dois que desenvolvam em casa o jogo de forma individual e que posteriormente o enviem para a plataforma *Moodle*.

Dei início à apresentação dos conteúdos previstos para a segunda aula, utilizando como recurso uma apresentação em PowerPoint (PPT) (disponível em [aula 2](#)), onde apresentei as condições da realização do projecto individual, que continha os objectivos e os requisitos que o

projecto individual em *Scratch* deveria cumprir, assim como os critérios de avaliação e prazos de entrega. Os alunos colocaram algumas dúvidas relativamente à realização de um relatório pois não estavam habituados com a elaboração de relatórios e mostraram algum receio com a data da entrega do trabalho, pois o tempo poderia não ser suficiente para concluírem o projecto, como mais tarde se verificou. Em relação aos seus medos sobre a realização dos relatórios tentei mostrar-lhes a importância dos mesmos, nomeadamente para aqueles que mais tarde pretendiam continuar os seus estudos na Faculdade, e que, se tivessem qualquer tipo de dificuldade eu estaria presente para os ajudar a tirar qualquer dúvida, através do e-mail ou Facebook (ferramenta muito utilizada pelos alunos). No entanto, nenhum aluno o fez.

De seguida, apresentei oralmente através do método expositivo os conceitos de listas do *Scratch* muito semelhantes aos da estrutura de *arrays* do PASCAL, com a ajuda do quadro desenhei algumas listas de dados e questioneei os alunos sobre os conteúdos dessas listas. Através deste exemplo concreto e possibilitando a visualização das listas, a maioria dos alunos esteve atenta à minha apresentação e demonstrou ter compreendido os conceitos apresentados, pois apresentou respostas correctas às minhas questões. Fiquei bastante contente com este facto pois os conceitos de listas e variáveis são os mais abstractos da programação e os mais difíceis de compreender.

De seguida, apresentei uma ficha de trabalho prática (disponível no PPT anteriormente mencionado) com quatro exercícios práticos, informei os alunos que pretendia que os dois primeiros fossem realizados na aula e que os dois últimos fossem realizados na aula seguinte dada pelo professor Filipe. Uma das razões para ter apresentado todos os exercícios e não somente aqueles que pretendia que realizassem na aula deveu-se ao facto de na turma existirem dois grupos de alunos distintos, uns mais avançados e com bastante prática a programar que facilmente resolviam os exercícios propostos e outro grupo com mais dificuldades que necessitava de mais apoio e auxílio da minha parte. Desta forma não era raro verificar que alguns alunos já tinham realizado os exercícios propostos para a aula e já estavam a realizar os exercícios da aula seguinte. Assim que terminavam a ficha prática tinham a nossa autorização (minha e do professor orientador cooperante) para trabalhar nos projectos individuais.

À medida que os alunos iam realizando os exercícios práticos, desloquei-me por entre a turma dando apoio individualizado sempre que solicitado. Os alunos com mais conhecimentos conseguiram resolver de forma autónoma os exercícios, enquanto que os alunos com mais dificuldades iam solicitando a minha ajuda no esclarecimento de algumas dúvidas. De acordo

como o método a que me tinha proposto implementar para o ensino da programação, nunca apresentava a solução aos alunos, mas sim tentava direccioná-los no sentido de eles próprios tentarem descobrir a solução, remetendo para exercícios anteriores ou exemplos apresentados, muitas vezes para a página do *Scratch* na internet para tentarem encontrarem exemplos semelhantes. Sempre que achava necessário introduzia novos conceitos teóricos através do método expositivo.

Este tipo de exercícios não são tão interessantes como foi o jogo do *Pac-man*, no entanto é importante que os alunos adquiram conhecimentos sobre manipulação de listas de dados, a única estrutura de dados do *Scratch*.

No final da aula pedi aos alunos para enviarem os exercícios resolvidos para a *Moodle*, fiz uma breve síntese e comuniquei à turma que na aula seguinte teriam de concluir a ficha de exercícios nº 2.

Na reunião seguinte que tive com o professor orientador cooperante, este informou-me que alguns alunos tiveram grandes dificuldades na resolução dos exercícios três e quatro, pelo que em conjunto decidimos que deveria continuar na aula seguinte com exercícios sobre manipulação de listas.

Mais tarde em casa, avalei os exercícios da ficha de trabalho nº 2 realizada durante as aulas. Todos os alunos conseguiram realizar os exercícios 1 e 2, razão que talvez se ficou a dever à ajuda que tinha dado aos alunos com mais dificuldades durante a aula, no entanto os exercícios três e quatro (de grau de dificuldade mais elevado) não tinham sido completamente resolvidos por oito alunos, que são aqueles que tinham sido identificados como alunos com mais dificuldades com a programação. No entanto fiquei contente com os resultados obtidos, pois como já anteriormente referi estes são os conceitos mais abstractos e difíceis da programação, e apesar de os exercícios 3 e 4 não estarem completamente resolvidos (pelos 8 alunos referidos), alguns raciocínios estavam correctos faltando apenas alguns pormenores de programação. De referir, que os alunos mais motivados e empenhados conseguiram resolver os exercícios 3 e 4 na sua plenitude apresentando algoritmos e formas originais e diferentes de resolver os problemas apresentados.

3.1.3 Terceira aula

A terceira aula começou por uma interligação com a aula anterior através da correção dos exercícios três e quatro da ficha de trabalho nº 2. Durante a correção referi com especial atenção as partes onde os alunos mais falharam, procurando esclarecer todos os alunos. Assim, foi feito um apanhado dos conceitos das duas aulas anteriores e orientei-os no sentido dos conteúdos previstos para a terceira aula.

Através do método expositivo, recorri a uma apresentação em PPT (disponível em [aula 3](#)), identifiquei as diferentes instruções de manipulação de listas, tendo apresentado em seguida a Ficha de Exercícios nº3 (PPT anterior). Pedi aos alunos para realizarem as duas primeiras alíneas do exercício, ficando as restantes para a aula seguinte leccionada pelo orientador cooperante. À medida que os alunos iam resolvendo os exercícios, desloquei-me por entre eles, dando apoio individualizado sempre que solicitado. À semelhança das aulas anteriores a maior parte dos alunos conseguiu resolver os exercícios apresentados, que já apresentavam um grau de complexidade e dificuldade elevado, no entanto quatro dos alunos identificados com dificuldades apenas conseguiram resolver os exercícios com as minhas orientações ou então recorrendo à ajuda dos colegas. Notei que os alunos colaboravam bastante entre eles e sempre que solicitados ajudavam os colegas com mais dificuldades.

No final da aula pedi aos alunos para enviarem os exercícios resolvidos para a *Moodle*, fiz uma breve síntese e comuniquei à turma que na aula seguinte teriam de concluir a ficha de exercícios nº 3.

Na reunião seguinte que tive com o orientador cooperante, este informou-me que os alunos conseguiram na sua maioria realizar os exercícios propostos na ficha de trabalho nº 3, não tendo percecionado durante a sua aula dificuldades maiores por parte dos alunos.

Mais tarde em casa, avalei os exercícios da ficha de trabalho nº 3 realizada durante as aulas. Dos vinte e seis alunos, dezanove conseguiram realizar na sua totalidade os exercícios propostos, desses dezanove encontro 10 alunos com maneiras diferentes de resolver os exercícios pelo que noto que esses alunos resolveram o exercício de forma autónoma desenvolvendo um raciocínio e algoritmo próprio. Dos nove restantes alunos noto algumas semelhanças na resolução, que se poderá dever à ajuda entre colegas. De salientar que no mapa da sala de aula estes nove alunos estão próximos uns dos outros, fazendo alguns par entre eles (mesmo PC). Dos restantes sete alunos, aqueles que demonstraram mais

dificuldades, um deles não realizou qualquer alínea (durante o ano este aluno mostrou-se sempre desmotivado e foram frequentes as chamadas de atenção durante as aulas para que trabalhasse), os restantes conseguiram realizar todas as alíneas de forma incompleta, pelo que faltavam alguns pormenores de programação para que os seus algoritmos funcionassem.

3.1.4 Quarta aula (supervisionada)

A quarta aula no âmbito do meu projecto começou com um breve resumo sobre manipulação de listas e suas instruções de controlo, com especial ênfase nos aspectos que os alunos mais haviam falhado na ficha de trabalho realizada na aula anterior. Foram resolvidas no quadro através do método de instrução directa algumas dúvidas levantadas pelos alunos.

De seguida, dei início aos conteúdos planeados para esta aula, instruções de movimento e caneta do *Scratch*, utilizei o método expositivo, recorrendo a uma apresentação em PPT (disponível em [aula 4](#)), para apresentar o sistema de coordenadas cartesianas. Inquiri os alunos para identificarem cinco pontos de coordenadas e sem dificuldade todos os alunos o fizeram, de forma a ter a certeza que todos os alunos estavam a compreender os conceitos apresentados, fui inquirindo os alunos, para identificar diversos pontos no ecrã do computador, à medida que me deslocava por entre a turma. Posteriormente, recorrendo à mesma apresentação em PPT, apresentei as instruções de movimento do *Scratch* e de forma resumida disse para que servia cada uma delas. Para me certificar que os conteúdos estavam a ser compreendidos por todos os alunos, apresentei um exercício em que os alunos tinham de desenhar uma seta e executar duas instruções, indicadas por mim. Perguntei aos alunos o resultado, e todos disseram que uma das instruções fazia rodar a seta no sentido dos ponteiros do relógio e a outra instrução fazia rodar no sentido contrário, fiquei contente com o facto de todos os alunos demonstrarem ter compreendido esta instrução, importante para execução do exercício seguinte. Da mesma maneira que tinha feito para as coordenadas cartesianas, volto a deslocar-me por entre os alunos de forma a ter a certeza que todos perceberam a utilidade das instruções apresentadas. Procedi de igual forma para o exercício que pedia para desenhar um triângulo e um quadrado. Ao deslocar-me por entre a turma notei que alguns deles para além de já terem desenhado o triângulo e o quadrado, já estavam a tentar desenhar casas com janelas! A maior parte dos alunos da turma B estava sempre muito motivada e com vontade de ir mais além. Dei tempo suficiente a que todos os alunos realizassem o exercício e no final para ter a certeza que todos

entenderam os conceitos apresentados, através do método de instrução direta, resolvi o exercício no quadro.

De forma a consolidar os conhecimentos adquiridos nesta aula apresentei o desafio de desenvolverem um algoritmo que permitia desenhar um polígono com o número de lados indicado pelo utilizador. Dei algumas dicas para a realização do exercício mas à luz da metodologia de ensino a que me propus, deixei os alunos resolverem o exercício por si de forma a descobrirem e desenvolverem o raciocínio lógico-abstracto. Fui-me deslocando por entre a turma, à medida que ia tirando algumas dúvidas e orientando os alunos com mais dificuldades, no sentido de descobrirem por eles próprios como se resolvia o exercício. Na sua maioria os alunos foram conseguindo resolver o exercício, apesar de alguns (aqueles com mais dificuldades) terem precisado da minha ajuda. À medida que a aula se aproximava do final, apercebi-me que os alunos mais avançados já tinham terminado o algoritmo, desta forma apresentei um segundo exercício de recurso (bola Picasso, ver no PPT anterior). Informei os alunos que estavam mais atrasados que resolveriam o exercício da bola Picasso na aula seguinte dada pelo professor orientador cooperante.

No final não tive tempo para fazer a síntese da aula e a interligação para a aula seguinte, pois deixei-me levar pelo entusiasmo dos alunos e perdi a noção do tempo. Mas após a saída dos alunos juntamente com o professor orientador cooperante decidimos que na aula seguinte os alunos que resolveram os dois algoritmos previstos para esta aula, iriam trabalhar nos projetos individuais e os restantes concluiriam os exercícios da ficha de trabalho nº 4 e posteriormente também iriam trabalhar nos seus projetos individuais.

Na reunião seguinte que tive com o professor orientador cooperante, este informou-me que todos os alunos conseguiram desenvolver os algoritmos de desenho, se bem que alguns alunos com mais dificuldades com a sua ajuda e a colaboração dos colegas. Os alunos já se encontram a trabalhar nos projetos individuais e já começam a surgir trabalhos muito interessantes.

Mais tarde em casa, avalei os exercícios da ficha de trabalho nº 4 realizada durante as aulas. Todos os alunos conseguiram desenvolver os algoritmos de desenho propostos, apesar de existirem formas bastante originais e diferentes de resolução, os alunos com mais dificuldades continuavam a apresentar soluções muito semelhantes entre si o que me sugeria que alguns alunos com mais dificuldades ainda não conseguiam programar de forma autónoma.

3.1.5 Quinta aula

A quinta aula no âmbito do meu projeto começou por um breve resumo dos exercícios anteriores, que apesar de terem sido resolvidos por todos, necessitavam de ver clarificados alguns pormenores na resolução, que ainda levantava certas dúvidas em alguns alunos.

Uma vez que os alunos já estavam a trabalhar nos seus projetos individuais, apresentei um exercício (disponível em [aula 5](#)) em que os alunos tinham de aplicar os conhecimentos adquiridos sobre manipulação de listas, no desenvolvimento de um pequeno jogo. O exercício para além de ser uma excelente forma para consolidarem os conhecimentos sobre manipulação de listas, constituía uma boa base para o desenvolvimento dos seus projetos individuais. Notava-se que os alunos andam motivados e muito deles já andam a desenvolver os seus projetos individuais, pediam-me opinião sobre os seus projetos e muitos deles já começavam a apresentar dúvidas concretas no que toca à programação em *Scratch* dos seus jogos.

Nesta aula tive alguma dificuldade em ter todos os alunos focados no exercício apresentado, pois como já referi muitos deles estavam mais preocupados com os seus jogos que consideravam mais importantes e avançados do que o exercício proposto. No entanto consegui demonstrar a importância do exercício e a utilidade do mesmo na resolução de algumas dúvidas que me estavam a colocar, pelo que no final consegui que todos avançassem na resolução do mesmo.

No final da aula pedi aos alunos para enviarem os jogos desenvolvidos para a *Moodle*, fiz uma breve síntese e comuniquei à turma que na aula seguinte teriam de concluir o jogo e continuar a trabalhar nos projetos individuais.

Na reunião seguinte que tive com o professor orientador cooperante, este informou-me que os alunos conseguiram concluir o jogo proposto na ficha de trabalho nº 5, não tendo percecionado durante a sua aula dificuldades maiores por parte dos alunos. Os alunos continuavam a trabalhar nos seus projetos individuais e o professor titular conseguia acompanhá-los sem dificuldades de maior, esclarecendo os alunos sempre que os mesmos lhe levantavam dúvidas. O facto de o professor orientador cooperante se sentir bastante à vontade na área da programação e ter desde o início demonstrado bastante interesse e inteirando-se plenamente de como funcionava o *Scratch*, foi uma mais-valia para o sucesso deste trabalho, caso contrário todo o projeto poderia ficar em causa. Volto a agradecer ao orientador cooperante, toda a ajuda, colaboração e disponibilidade durante a realização do meu estágio na ESNB.

Mais tarde em casa, avaliei os exercícios da ficha de trabalho nº 5 realizada durante as aulas. Todos os alunos conseguiram desenvolver um jogo respondendo aos requisitos pré-estabelecidos, na forma como já começavam a desenvolver os seus trabalhos já notava mais autonomia nos alunos com mais dificuldades com exceção de um aluno que apresentava um exercício completamente igual ao do vizinho do lado. Por várias vezes tinha alertado este aluno para a sua falta de motivação mas todos os meus esforços para conseguir que este aluno trabalhasse na sala de aula até ao momento se tinham revelado infrutíferos.

3.1.6 Sexta aula (supervisionada)

Na sexta aula do meu projeto os alunos chegaram bastante alterados à sala de aula, tinham tido teste de matemática, que na opinião deles tinha sido bastante difícil e que a notas não iriam ser as melhores, como já tinha referido na contextualização da turma, os alunos do 12º B eram muito preocupados com as suas médias e na sua maioria estavam interessados em seguir para Universidade. Notei nesta aula que estavam muito abalados! Foram necessários os 15 minutos iniciais da aula para que todos chegassem e se comessem a acalmar. De certa forma este incidente condicionou a participação na aula de alguns alunos, que no seu estado normal participavam bastante e colaboravam nas aulas.

Assim, tentei acalmar os alunos falando um pouco com eles, quando já estavam todos sentados e mais calmos dei início à aula. Comecei por fazer a interligação com a aula anterior, analisando alguns dos exercícios resolvidos, dando especial destaque à resolução do exercício por parte de um aluno, que demonstrou bastante originalidade na sua resolução, criando um jogo com elevada jogabilidade. Dada a qualidade do trabalho realizado, resolvi partilhá-lo com a restante turma, apresentando-o como um bom exemplo de resolução do exercício da ficha prática nº 5.

De seguida, dei início aos conteúdos planeados para a sexta aula, comecei por dialogar com os alunos interrogando-os “porque é necessário ordenar os dados?”, uma aluna referiu o exemplo de bases de dados de alunos nas escolas, outro aluno apresentou o exemplo das páginas amarelas (imagem apresentada no PPT, disponível em [aula 6](#), à qual não me tinha referido). Os alunos estavam cientes da necessidade de ordenar dados, pelo que continuei com uma pequena exposição sobre o assunto apresentando uma definição científica por parte de um autor do que é ordenar.

Posteriormente, comecei por apresentar o algoritmo de ordenação por seleção, de forma a os alunos entenderem melhor, como se processa este método de ordenação fiz uma traçagem do algoritmo recorrendo ao quadro e caneta, com alguns valores projetados. À medida que fui traçando o algoritmo, fui interpelando os alunos e recebendo *feedback* de que estes estavam a perceber de como se processava a traçagem do mesmo. Na parte do Pseudocódigo para posterior implementação no *Scratch*, notei que os alunos estavam com mais dificuldade em perceber como se processava o mesmo, quando os interpelei, para perceber se os mesmos estavam a compreender, notei que uma aluna não estava muito convencida, pelo que decidi chamá-la ao quadro para exemplificar aos colegas como se processava a troca de valores utilizando uma variável auxiliar. No final apercebi-me que a aluna tinha compreendido e mais tarde concretizei essa minha percepção ao ver que a mesma estava no bom caminho na implementação do algoritmo no *Scratch*.

Em seguida, lancei o desafio de os alunos implementarem o Algoritmo de seleção no *Scratch*. Para isso forneci algumas dicas, pois estava ciente da dificuldade do exercício em causa. Movimentei-me por entre a turma e constatei que os alunos, com mais dificuldade na programação, estavam muito longe de conseguir implementar o algoritmo, e até ao final da aula pouco progressos fizeram nesse sentido. No entanto verifiquei que alguns alunos estavam bem encaminhados. Os alunos que demonstravam mais dificuldades no que toca à programação estavam a tentar adaptar um exercício anteriormente resolvido sobre manipulação de listas, porém estavam bastante longe de aproximarem-se da resolução do algoritmo de ordenação.

Nesta aula notei que um par de alunos que trabalham no mesmo PC, funcionava mal, pois quando um dos alunos estava sozinho (por motivo de o colega ter faltado) conseguia trabalhar melhor do que na companhia do colega, que pelo facto de faltar bastante estava mais atrasado em termos de conhecimentos na área da programação. Tive o cuidado de nas aulas posteriores os separar e nos trabalhos de grupo não os colocar no mesmo grupo de trabalho. À medida que a aula foi decorrendo, fui interpelado por vários alunos e notei que estavam interessados em resolver o algoritmo, fui dando resposta às suas questões devido ao grau de dificuldade do exercício, a movimentação na sala de aula foi constante, e comecei a ficar cansado pois não era fácil dar aulas deste tipo a 26 alunos.

Quando faltavam 15 minutos para o final da aula notei que alguns alunos estavam muito próximos da resolução mas no entanto ainda faltava um ou outro pormenor para o algoritmo funcionar, pelo que decidi apresentar a resolução do algoritmo no quadro. À medida que o ia

fazendo ia alertando alguns alunos para os problemas que estavam afectar a implementação dos seus algoritmos. Notei que fizeram de imediato os ajustes necessários. Em seguida passei novamente pelos alunos e pelo menos cinco alunos conseguiram colocar o algoritmo a funcionar, no entanto, não tive tempo de verificar se os restantes alunos tinham conseguido concluir o exercício, pelo que pedi que enviassem os seus algoritmos de ordenação para a *Moodle* para posterior correcção.

No final fiz uma breve síntese da aula e informei os alunos que na aula seguinte iriam implementar o algoritmo de ordenação num jogo, demonstrando a sua utilidade para apresentação da pontuação dos jogadores por ordem decrescente.

Na reunião seguinte que tive com o orientador cooperante, este informou-me que os alunos conseguiram implementar o algoritmo de ordenação no jogo desenvolvido na ficha de trabalho nº 5, no entanto referiu que a maior parte utilizou um algoritmo que já estava a funcionar e que tinha sido desenvolvido por um dos alunos mais avançados, não tendo percepcionado durante a sua aula dificuldades maiores por parte dos alunos. Os alunos continuavam a trabalhar nos seus projectos individuais e o orientador cooperante conseguiu acompanhá-los sem dificuldades de maior, esclarecendo os alunos sempre que os mesmos lhe levantavam dúvidas.

Mais tarde verifiquei que somente 12 alunos tinham conseguido implementar o algoritmo de ordenação, os restantes alunos apresentaram soluções aproximadas mas que não funcionavam corretamente (os valores não apareciam ordenados), no entanto estes resultados não me deixavam preocupado porque implementar um algoritmo de ordenação não é fácil e são assuntos que serão abordados mais tarde na faculdade, para os alunos que pretendam seguir os estudos na área da programação. A minha intenção com este exercício foi mais uma vez desenvolver o raciocínio lógico-abstracto e não propriamente a sua implementação, no entanto fiquei bastante contente por 12 alunos o terem conseguido. A implementação do algoritmo no jogo realizada no decorrer da aula leccionada pelo professor orientador cooperante tinha sido feita por todos, uma vez que todos foram capazes de pegar no algoritmo desenvolvido por um colega e adaptá-lo para o seu jogo. Esta realidade, deixou-me bastante motivado pois o facto de os alunos conseguirem adaptar o código dos colegas é também bastante importante para a programação.

3.1.7 Sétima aula

A sétima aula no âmbito do meu projeto começou por um breve resumo sobre a implementação do algoritmo de ordenação por seleção em jogos, que apesar de ter sido resolvidos por todos, necessitava de ver clarificados alguns pormenores na resolução, que ainda levantava certas dúvidas em alguns alunos.

Com o objectivo de apoiar os alunos no desenvolvimento dos seus projetos individuais e introduzindo o tema dos trabalhos de grupo a desenvolver no terceiro período, fiz uma breve introdução, recorrendo ao método expositivo através de uma apresentação em PPT (disponível em [aula 7](#)), sobre os RPG. Explorei o tema dos jogos com os alunos, tema no qual todos se sentiam à vontade, e em seguida, apresentei um exercício para desenvolvimento de um pequeno RPG.

Todos os alunos da turma começaram de imediato a resolver o exercício, e demonstraram bastante entusiasmo quando encontravam situações que podiam ser adaptadas para os seus projetos individuais. De salientar que os alunos realizavam este tipo de atividades (desenvolvimento de jogos) com mais interesse do que os exercícios sobre manipulação de listas.

No final fiz uma breve síntese da aula e informei os alunos que na aula seguinte iriam terminar o exercício do RPG, uma vez que o tempo não tinha sido suficiente para o concluírem.

Na reunião seguinte que tive com o orientador cooperante, este informou-me que os alunos conseguiram resolver sem grandes problemas o exercício sobre o desenvolvimento de um pequeno RPG, proposto na sétima aula. Em conjunto e uma vez que estávamos dependentes dos conteúdos multimédia desenvolvidos pela turma do 12º E, decidimos que o trabalho de grupo seria desenvolvido no terceiro período e que as próximas aulas seriam dedicadas ao desenvolvimento dos projetos individuais, que por sinal estavam a ficar muito interessantes, com trabalhos com elevada qualidade por parte de alguns alunos. O professor orientador cooperante apresentou ainda algumas dúvidas que tinham sido colocadas pelos alunos relativas aos seus projetos individuais, mas que não tinha tido tempo de esclarecer na sua aula.

Mais tarde verifiquei que todos os alunos tinham conseguido desenvolver o pequeno RPG respondendo a todos os pré-requisitos estabelecidos.

3.1.8 Oitava aula

Na oitava aula no âmbito do meu projeto, comecei por fazer um breve resumo sobre o desenvolvimento de um pequeno RPG, que apesar de todos os alunos terem sido capazes de desenvolver, necessitava de ver clarificados alguns pormenores aos quais os alunos não deram muita importância mas que eram fundamentais no desenvolvimento dos seus projetos individuais. Em seguida informei os alunos que os trabalhos de grupo transitarão para o terceiro período e que as próximas aulas seriam exclusivamente dedicadas ao desenvolvimento dos seus projetos individuais. Os alunos ficaram bastante aliviados, por saberem que iriam ter até ao final do período tempo suficiente para desenvolver os seus projetos.

No sentido de os apoiar novamente no desenvolvimento dos seus projectos individuais e procurando esclarecer algumas dúvidas que tinham sido colocadas ao professor titular, como anteriormente referi, nesta aula apresentei o desafio para desenvolverem um pequeno jogo de *Ping-Pong* (disponível em [aula 8](#))

Todos os alunos da turma começaram de imediato a resolver o exercício, e novamente demonstraram bastante entusiasmo quando encontravam situações que podiam ser adaptadas para os seus projetos individuais.

No final fiz uma breve síntese da aula e informei os alunos que na aula seguinte iriam terminar o exercício do jogo de *Ping-Pong*, uma vez que o tempo não tinha sido suficiente para o concluírem nesta aula.

Na reunião seguinte que tive com o orientador cooperante, este informou-me que os alunos conseguiram resolver sem grandes problemas o jogo de *Ping-Pong*.

Mais tarde verifiquei que todos os alunos tinham conseguido desenvolver o jogo de *Ping-Pong* respondendo a todos os pré-requisitos estabelecidos, existindo inclusive uma versão muito original do jogo que decidi partilhar na aula seguinte com a turma.

3.1.9 Nona aula à décima terceira aula

Iniciei a nona aula do meu projeto de intervenção, chamando ao quadro o aluno que tinha desenvolvido a versão mais original do jogo de *Ping-Pong*. Pedi ao aluno que mostrasse aos colegas como se jogava o seu jogo e que mostrasse os aspectos mais relevantes no seu

desenvolvimento. Os colegas mostraram-se bastante interessados e quiseram saber mais sobre alguns pormenores de programação.

A partir desta aula passei a dedicar a totalidade da aula para o desenvolvimento dos projetos individuais e esclarecimento de dúvidas. Fui verificando os progressos feitos por cada aluno e conversando com a turma para perceber o que faltava fazer e as dificuldades que estavam a sentir. Muitos alunos já tinham os seus projetos bastante avançados e notava que os alunos desenvolviam os trabalhos em casa aproveitando as aulas para tirar dúvidas. Em todas as aulas fui solicitado para auxiliar em pequenos problemas e dificuldades com que os alunos se iam deparando. Procurava sempre esclarecer os alunos orientando-os no sentido de eles próprios encontrarem uma solução, recorrendo a exercícios anteriormente resolvidos ou fornecendo exemplos semelhantes, mas nunca a solução.

A partir desta nona aula também comecei a intervir às terças e quintas-feiras pelo que deixou de haver aquele interregno que de certa forma prejudicava a minha intervenção, pois os alunos alternavam de professor de aula para aula, não havendo uma continuidade de raciocínio no esclarecimento de dúvidas, uma vez que tanto eu como o professor Filipe tínhamos formas próprias de programar e de desenvolver os algoritmos.

No final da décima primeira aula, dei por concluídos os trabalhos pelo que informei os alunos que deveriam enviar os seus projetos acompanhados de um relatório para a plataforma *Moodle* para posterior correção e avaliação.

Na penúltima aula do segundo período (décima segunda do meu projeto de intervenção), iniciei a apresentação dos trabalhos individuais por parte de cada aluno, solicitando a atenção da restante turma. Enquanto os alunos apresentavam os seus projetos, fui avaliando a qualidade das apresentações (no ponto 3.2.1 deste relatório são apresentados os resultados). No final das apresentações, durante a décima terceira aula do meu projeto de intervenção, coloquei algumas questões relativamente ao processo de desenvolvimento dos diferentes projetos, nomeadamente: se tinham gostado de aprender a trabalhar com o *Scratch*; quais as maiores dificuldades sentidas; o que haviam gostado mais no projeto; o que mudariam, entre outras. Procurei sempre ouvir todos os alunos, incentivando aqueles que revelavam maior timidez. Em seguida salientei os pontos fortes e fracos de cada projeto, tentando sempre mostrar como poderiam ter resolvido as falhas identificadas. As apresentações correram muito bem e toda a turma colaborou, discutindo e criticando de forma construtiva os trabalhos uns dos outros.

A última meia hora da aula foi dirigida pelo professor orientador cooperante que procedeu à autoavaliação por parte dos alunos, sobre a nota do segundo período.

3.1.10 Décima quarta aula

No 3º Período de aulas, iniciei a minha intervenção no dia 16 de Abril com a décima quarta aula do meu projeto de intervenção. As aulas anteriores foram asseguradas pelo professor orientador cooperante, que esteve a apresentar matéria relativa ao módulo Multimédia que fazia parte do programa da disciplina de Aplicações Informáticas B, no âmbito deste módulo o professor titular propôs um trabalho de projeto individual de desenvolvimento de um vídeo, que deveria ser concluído até final do terceiro período.

O objectivo da minha intervenção no terceiro período foi verificar se todos os alunos tinham aprendido a programar em *Scratch* e se eram capazes de trabalhar em grupo e entre turmas. Para isso teriam que desenvolver um jogo RPG em *Scratch*, de forma colaborativa e cooperativa, trabalhando em grupo dentro da turma B e entre turmas com a turma do 12º E, turma acompanhada pelas minhas colegas estagiárias Fátima Oliveira e Cátia Silva, no âmbito dos seus projetos de intervenção. A turma E era responsável pelo desenvolvimento do guião do jogo, cenários, sons e personagens do jogo (*sprites*), trabalho que já se vinha a desenvolver desde o segundo período mas que ainda não estava terminado. A turma do 12º B seria responsável pela programação no *Scratch* das diferentes cenas do jogo.

No interesse do trabalho desenvolvido pela minha colega Fátima Oliveira, tentei estimular ao máximo para que os meus alunos utilizassem as ferramentas de trabalho colaborativo e cooperativo disponibilizadas, *Facebook* e *Dropbox*, de forma a que a comunicação entre as turmas fosse a melhor possível, no entanto a minha maior preocupação focava-se no facto de os alunos programarem bem, de forma estruturada e com boa comunicação entre os grupos para que no final fosse possível obter um produto final, jogo RPG. Em termos de programação, não via problemas de maior pois alguns alunos tinham desenvolvido trabalhos individuais bem mais complexos do que aquilo que se pretendia com o jogo RPG. A grande dificuldade centrava-se na integração de todas as cenas, num produto final. Para o bom progresso do projeto, e de acordo com a metodologia de desenvolvimento referida, apresentei um protótipo de um jogo RPG, devidamente testado em termos de usabilidade, que serviu como azimute, direccionando os alunos no bom sentido.

Nesta aula fez-se a escolha dos grupos e atribuição das cenas. Para escolher os grupos foram nomeados 5 líderes de projeto. Os líderes foram selecionados devido ao seu espírito de liderança, organização e capacidade de ajuda com os colegas. De forma a evitar que os grupos fossem escolhidos de acordo com as amizades e para garantir um grupo o mais heterogéneo possível, foi tirada à sorte a ordem pela qual os líderes dos grupos escolheriam os seus membros e alternadamente cada líder escolheu um elemento para o seu grupo, até se perfazerem cinco grupos. As cenas do guião (em anexo) foram atribuídas segundo as minhas intenções, atendendo às capacidades dos elementos de cada grupo. A Tabela 3 apresenta os diferentes grupos criados e quais as cenas do guião atribuídas.

Grupo	Nº de elementos	Cenas do Guião
Grupo 1	5	Cena 3 - Biblioteca
Grupo 2	5	Cena 4 - Cantina
Grupo 3	5	Cena 5 – Laboratório de Química Cena 6 – Cenário de sucesso Cena 7 – Cenário de insucesso
Grupo 4	6	Cena 0 – Introdução Cena 1 – Entrada da ESNB Movimento do Personagem Principal
Grupo 5	5	Cena 2 – Passagem para biblioteca

Tabela 3 - Grupos de trabalho e cenas do guião atribuídas

3.1.11 Décima quinta aula

No início da décima quinta aula fiz um breve resumo, sobre os grupos de trabalho e cenas atribuídas.

Em seguida, na presença da colega estagiária Fátima Oliveira, apresentámos o *Dropbox* aos alunos, mostrando onde se encontravam os conteúdos multimédia, imagens, sons e vídeos. Apresentou-se também o *teaser* do jogo, vídeo desenvolvido pela turma E de forma a motivar e integrar a turma do 12º B no espírito de trabalho entre turmas.

Alertamos os alunos para o facto de poderem utilizar o *Facebook* de forma a interagir com a turma E, solicitando aos colegas todos os materiais que necessitassem para desenvolver a programação das suas cenas.

Nesta aula os alunos exploraram os diferentes materiais e todos os grupos já começaram a trabalhar com o *Scratch*, verificando como ficavam os cenários e *sprites*.

A Professora Fátima apresentou a disponibilidade por parte da turma E em trabalhar com a turma B e dar resposta a todos os seus pedidos.

No final fiz uma síntese dos conteúdos da aula e informei os alunos que na aula seguinte continuaríamos a trabalhar nos projectos de grupo/turma .

3.1.12 Décima sexta aula

A décima sexta aula do meu projeto de intervenção, começou com um breve resumo da aula anterior, onde referi novamente as ferramentas de trabalho colaborativo disponíveis, fazendo no final um breve ponto da situação de cada um dos grupos de trabalho.

Notei os alunos um pouco alterados e desmotivados, pelo que abrimos então uma discussão para troca de ideias para ver o que os andava a incomodar. Uma aluna argumentou que a qualidade do material multimédia, nomeadamente as imagens estavam com pouca qualidade, pelo que juntamente com outro aluno se disponibilizaram para retocar as imagens, referiu ainda que durante o *brainstorm*, realizado na turma E no final do primeiro período, para o qual alguns alunos da turma B tinha sido convidados, tinham deixado várias ideias e nenhuma das suas ideias tinha sido implementada em termos de guião do jogo. Todos os alunos manifestaram que a turma B deveria ter sido escutada durante essa fase, e que sentiam nesse momento que a turma B não estava representada no jogo. "*Destruir zombies com uma régua...não é uma boa ideia!*", referiu uma aluna (aluna A), tendo a restante turma começado a rir-se em uníssono. Outro aluno, estava preocupado com a integração das diferentes cenas, pelo que sugeri que todos os líderes de projeto se deveriam reunir na aula e definirem em conjunto as regras do jogo em termos de programação. Outro aluno, voltou novamente a queixar-se da qualidade das imagens, traços pretos em todos os personagens com que o seu grupo tinha de trabalhar. Notei uma certa resistência na turma do 12º B em trabalhar neste projeto entre turmas, muitos deles estavam mais interessados em trabalhar no seu projeto

individual multimédia (vídeo), do que propriamente desenvolver o projeto *Zschool* (nome dado pela turma E ao jogo RPG) no *Scratch*.

Nesta altura da minha intervenção, estava com alguma dificuldade para que pegassem no projecto *Zschool* de forma produtiva. Aproveitei a sugestão dada anteriormente por um aluno e sugeri que o líderes se reunissem e vissem o que era necessário alterar em termos de conteúdos desenvolvidos pela turma E, que definissem as variáveis comuns e a forma como deveriam ser feitas as interligações entre as diferentes cenas. Os restantes elementos do grupo continuavam a trabalhar nos seus projetos, alguns no *Zschool*, outros no vídeo.

Neste momento refleti que deveríamos ter envolvido mais a turma B na construção do guião do jogo, o facto de este ter sido desenvolvido unicamente pela turma E não tinha sido uma boa estratégia, pois agora estava com dificuldade em motivar os meus alunos para o *Zschool*, um projecto com o qual não se identificavam.

Incentivei os alunos a colocarem os seus comentários, de forma construtiva, no *Facebook* e pedir aos colegas da turma E que alterassem os cenários e personagens que achavam que deviam ser alterados.

No final fiz uma síntese dos conteúdos da aula e informei os alunos que na aula seguinte continuaríamos a trabalhar nos projetos de grupo/turma .

3.1.13 Décima sétima aula à décima oitava aula

Nestas duas aulas continuei com os trabalhos de grupo, os líderes de projeto voltaram a reunir-se de forma a estabelecer novas regras de programação, nomeadamente na forma como se iria movimentar o personagem principal no palco do jogo. De acordo com o guião elaborado pela turma E, o personagem principal deveria saltar, mas segundo os alunos da turma B se o personagem se movimentasse da forma como estava definido, em termos de implementação no *Scratch* não era possível colocar o personagem a saltar e mesmo tempo a deslocar-se nos dois eixos de coordenadas (XY), os saltos só eram permitidos se se tratasse de um jogo de plataformas. Os alunos solicitaram o meu auxílio com este problema e entre todos chegamos à conclusão que o melhor era o personagem não saltar. Em termos de jogabilidade não afectava o que estava estipulado no guião, pelo que dei conhecimento à minha colega de estágio desta decisão em termos de implementação no *Scratch* tomada pela turma.

Notei nestas duas aulas que todos os grupos se encontram motivados e todos se encontram a trabalhar no *Zschool*.

O grupo 3, na cena do laboratório de química, estava com dificuldades em criar o pequeno jogo que dava acesso ao antídoto. Depois de algumas dicas e trocas de ideias, os alunos conseguiram entre eles chegar à conclusão daquilo que pretendiam desenvolver, pelo que a partir desse momento começaram a desenvolver o algoritmo.

No final das aulas fiz um breve resumo de como estavam a correr os trabalhos e informei os alunos que nas aulas seguintes continuariam a desenvolver os seus trabalhos de grupo. A partir destas aulas comecei a pedir aos alunos para no final da aula, guardarem os seus trabalhos numa pasta da *Dropbox* de forma a poder em casa acompanhar e verificar como estavam a decorrer os trabalhos.

3.1.14 Décima nona aula (supervisionada)

Na décima nona aula da minha intervenção, comecei por fazer a interligação com a aula anterior, fazendo um breve resumo sobre as dificuldades, identificadas em casa, no desenvolvimento dos trabalhos de grupo.

Em seguida, dei início aos conteúdos planeados para esta aula, recorrendo ao método expositivo e a um PPT (disponível em [aula 19](#)) apresentei um protótipo de um jogo desenvolvido em *Scratch*, devidamente testado no que toca à usabilidade do mesmo (ver ponto 2.1.4). Através do protótipo tentei implementar uma metodologia de desenvolvimento, em que há medida que os alunos iam desenvolvendo os seus trabalhos de grupo pretendia-se que recorressem ao protótipo sempre que se deparassem com alguma dificuldade ou dúvida. Assim o protótipo serviria como um farol que orientaria os alunos no bom caminho para a conclusão dos seus trabalhos.

Em seguida, apresentei algumas soluções, que o protótipo poderia ajudar a resolver, para algumas dificuldades referidas no início da aula.

Posteriormente, pedi aos alunos que explorassem o protótipo à medida que continuavam com o desenvolvimento dos seus trabalhos de grupo. À medida que me desloquei por entre a turma, fiquei contente em verificar que os alunos estavam a explorar o protótipo e a identificar situações em que poderiam recorrer à ajuda do mesmo. Fui inquirindo os diferentes grupos para ver de que forma o protótipo lhes poderia ser útil. Os alunos do grupo 4, ao qual tinha sido

atribuídas as cenas 0,1 e 8 do guião, mostraram-se bastante entusiasmados com o facto de o protótipo ser útil para ajudar resolver os problemas que estavam a ter com a parte em que tinham de saltar a introdução do jogo, com o movimento do personagem principal e movimento da reguada que tirava energia aos zombies. Os alunos estavam bastante motivados e entre eles começaram a dividir tarefas e a alterar os seus programas no *Scratch*. Os alunos do grupo 5 (cena 2), que estavam com dificuldade em fazer o movimento dos zombies, a implementar as fronteiras nos cenários e efetuar as transições entre palcos, mostraram-se bastante motivados e referiram ter encontrado no protótipo o algoritmo que os ajudaria a ultrapassar as suas dificuldades, pelo que continuaram compenetrados no desenvolvimento dos seus códigos. Os alunos do grupo 1 (cena 3), mostraram interesse em explorar o protótipo, no entanto consideravam o seu trabalho bastante encaminhado e a grande dificuldade que estavam a ter era com a transição entre palcos, pelo que iriam explorar o protótipo para ver se conseguiam ultrapassar essa dificuldade. O grupo 2 (cena 4), também estava bastante avançado no seu trabalho, no entanto mostraram interesse no protótipo para os ajudar com o lançamento dos objetos. O grupo 3 (cenas 5,6 e 7) não via no protótipo qualquer tipo de ajuda para o trabalho que estavam a desenvolver no entanto consideravam-no importante para os orientar quando tivessem que fazer a junção de todas as cenas e interligação das mesmas. O facto de também estarem a ver um produto final ajudava a motivá-los para concluírem o seu trabalho.

No final da aula fiquei bastante contente por ter introduzido o protótipo e senti que foi um factor de motivação para todos os alunos, tornando-os ainda mais interessados em concluir o *Zscholl*.

Terminei a aula com uma breve síntese e comuniquei à turma que na aula seguinte iriam continuar a desenvolver os seus trabalhos de grupo.

3.1.15 Vigésima à vigésima terceira aula

Na vigésima aula da minha intervenção a colega Fátima esteve presente de forma a verificar as necessidades em termos de alterações de cenários e personagens pretendidas pelos meus alunos, para além de colocarem os seus comentários no *Facebook*, estimei os alunos a dar conhecimento diretamente à colega de forma a agilizar os processos pois já não faltam muitos dias para o término das aulas.

Durante estas aulas decorreram os trabalhos de grupo e verifiquei que alguns alunos recorriam ao protótipo várias vezes ao longo das aulas (na avaliação dos trabalhos de grupo no ponto 3.2.2 irei avaliar a extensão da utilização do protótipo).

No decorrer das aulas, a implementação das diferentes cenas do guião já começavam a ficar concluídas, pelo que os alunos começaram a parte complicada do projeto, que era a integração das diferentes cenas num jogo final. Os alunos demonstravam alguma dificuldade mas notei entre os grupos uma grande ajuda. Era normal no decorrer destas aulas ver elementos de um grupo a levantar-se e ir auxiliar outro grupo explicando como desenvolveram a sua programação de forma a poderem integrar as diferentes cenas. Também era frequente ver os alunos a detectar um erro de programação e recorrer ao protótipo para encontrar a solução. Foram raras as vezes em que foi solicitada a minha ajuda.

Na vigésima terceira aula demos por concluído em termos de programação o *Zschool*. Nesta aula a colega Fátima esteve presente de forma a verificar como tinha ficado o projeto final. De forma a se poder concluir o jogo tiveram que ser eliminadas algumas falas pois as mesmas não estavam de acordo com a programação desenvolvida. O ideal, teria sido haver mais duas semanas de aulas para se poder testar devidamente o jogo e efetuar todas as alterações necessárias em termos de sons e correção de imagens. Tal não foi possível, no entanto, notei que os alunos estavam bastante contentes com a versão Beta obtida.

3.1.16 Vigésima quarta aula

Neste vigésima quarta aula da minha intervenção realizei uma entrevista coletiva do tipo *Focus Group*, para avaliação final do projeto por parte dos alunos da turma B.

No ponto 3.2.3 são apresentadas as impressões por parte dos alunos durante este *Focus Group* final.

3.1.17 Vigésima quinta aula

Na vigésima quinta e última aula da minha intervenção aproveitei o início da aula para refletir com os alunos sobre a forma como tinha decorrido o meu projeto de intervenção, atendo ao facto que esta tinha sido a primeira vez que tinha leccionado programação a alunos do 12º ano com o *Scratch*, fiquei bastante contente com o *feedback* dado pelos alunos, pois foi notório

que gostaram de trabalhar nos diferentes projetos e que no final do ano lectivo muitos se sentiam motivados para ingressar em cursos na área das TI e aprender mais sobre programação. A restante aula foi dirigida pelo professor orientador cooperante, com a auto-avaliação dos alunos.

3.2 Fase de avaliação e apresentação de resultados

3.2.1 Avaliação e apresentação de resultados dos trabalhos individuais

Como já referi, numa primeira fase da minha intervenção propus aos alunos o desenvolvimento de um jogo individual desenvolvido em *Scratch* que deveria cumprir um determinado número de requisitos. Este jogo fazia parte da avaliação dos alunos no segundo período, aspecto que foi acordado e definido com o professor orientador cooperante.

Foram analisados um total de 26 jogos relativos aos trabalhos individuais desenvolvidos ao longo do segundo período. De salientar que não avalei apenas o produto final, mas também o processo de desenvolvimento do mesmo, nomeadamente o interesse e empenho dos alunos no decorrer das aulas, assim como o cumprimento dos requisitos, cumprimento dos prazos de entrega, criatividade, e qualidade do código desenvolvido.

Nos estudos de Maloney et. Al (2008), que analisaram 536 projetos desenvolvidos em *Scratch* num clube, os autores chegaram à conclusão que 111 desses projetos não poderiam ser considerados programação uma vez que não continham nenhum dos setes conceitos de programação definidos por eles, a saber: (1) interação com utilizador (uso do teclado ou rato para introdução de dados); (2) ciclos; (3) declarações condicionais; (4) comunicação e sincronização; (5) lógica Booleana; (6) variáveis e (7) números aleatórios. Nos restantes 425 projetos, os autores verificaram a existência de pelo menos um dos sete conceitos de programação anteriormente referidos.

Desta forma, procedi de igual modo na análise aos projetos individuais dos alunos, para determinar se os alunos aprenderam, de facto, conceitos de programação com o *Scratch*.

3.2.1.1 Projeto *Harlem Shake*

O aluno que desenvolveu este projeto revelava imensas dificuldades na área da programação informática. Estava identificado como um aluno pouco motivado para a programação (fase de observação) e com pouca vontade de aprender (dados do Plano de Turma). Não tinha interesse em continuar os estudos a nível superior e foi necessário na sala de aula estar sempre a chama-lo a atenção para que trabalhasse.

Harlem Shake	
Comunicação e sincronização	3
Ciclos	16
Declarações condicionais	0
Interacção com utilizador	0
Lógica Booleana	0
Variáveis	0
Números aleatórios	0

Estrutura	
Nº de cenários do palco	1
Nº de <i>sprites</i>	7

Tabelas 4 e 5 – Conceitos de programação e estrutura do projecto *Harlem Shake*.

Como se pode concluir, da análise da Tabela 4, o aluno que desenvolveu este projeto, ao qual deu o nome de *Harlem Shake*, evidencia que apenas dois dos sete conceitos de programação analisados foram implementados no desenvolvimento do seu trabalho. Este prima pela originalidade demonstrada, e carácter cómico associado à apresentação multimédia. Através da análise da Tabela 5, podemos verificar que o projeto é bastante simples, envolvendo apenas um cenário de palco e 7 *sprites*. Na Figura 10 apresenta-se a interface gráfica que também se revela bastante simples.



Figura 10 - Interface do Projecto *Harlem Shake*

3.2.1.2 Projeto Jogo de Carros

A aluna que desenvolveu este jogo revelava algumas dificuldades e pouca autonomia. Anotei, ao longo da minha intervenção, que a aluna era ajudada muitas vezes pelos colegas e tinha vergonha de pedir ajuda ao professor de forma a não evidenciar as lacunas que tinha na programação. No entanto, como podemos constatar através da análise da Tabela 6, o seu projeto conseguiu evidenciar a quase totalidade dos conceitos de programação estudados com exceção da utilização de instruções que evidenciassem o uso de números aleatórios. No entanto, o projeto, um jogo de carros, de fraca jogabilidade e muito pouco original, no que toca à estrutura (Tabela 7), envolve 9 cenários e 28 *sprites*. A Figura 11 apresenta a interface do jogo, muito simples e pouco elaborada.

Jogo de carros	
Comunicação e sincronização	171
Ciclos	55
Declarações condicionais	41
Interacção com utilizador	12
Lógica Booleana	41
Variáveis	6
Números aleatórios	0

Estrutura	
Nº de cenários do palco	9
Nº de <i>sprites</i>	28

Tabelas 6 e 7 – Conceitos de programação e estrutura do projecto Jogo de Carros.

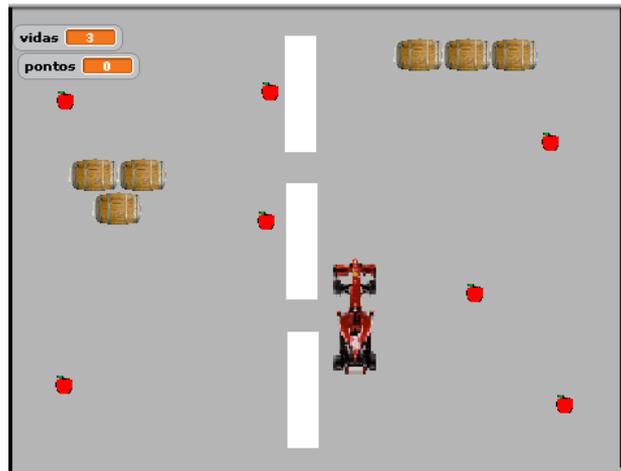


Figura 11 - Interface do Projecto Jogo de Carros

3.2.1.3 Projeto O Jogo Mais Difícil do Mundo

Jogo bastante agradável e viciante pelo grau de dificuldade que apresenta. Apesar de ser muito difícil de jogar a sua jogabilidade é elevada, pois o jogador quer sempre jogar mais para poder ultrapassar os desafios que se lhe deparam pela frente (Figura 12). Este jogo foi desenvolvido por um aluno com bons conhecimentos na área da programação e conseguiu demonstrá-los com a execução deste trabalho. Apesar de não ter utilizado instruções que evidenciassem o conceito de números aleatórios (Tabela 8), tal deveu-se ao facto de não serem necessários estas instruções neste tipo de jogo, e não vejo onde poderiam ter sido utilizadas. O jogo está bem dimensionado e apresenta uma estrutura de 4 cenários de palco e 54 *sprites* (Tabela 9).

O Jogo mais difícil do Mundo	
Comunicação e sincronização	165
Ciclos	42
Declarações condicionais	14
Interacção com utilizador	6
Lógica Booleana	14
Variáveis	5
Números aleatórios	0

Estrutura	
Nº de cenários do palco	4
Nº de <i>Sprites</i>	54

Tabelas 8 e 9 – Conceitos de programação e estrutura do projecto Jogo mais difícil do Mundo.

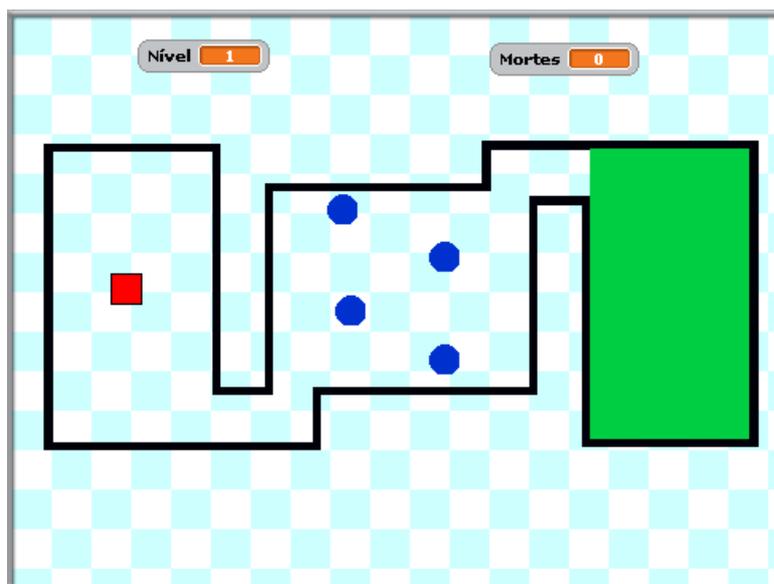


Figura 12 – Interface do Jogo mais difícil do Mundo

3.2.1.4 Projecto *Snake*

Jogo muito colado ao original, revelando pouca originalidade. No entanto o aluno revela bons conhecimentos na área da programação e conseguiu demonstrá-los na execução deste trabalho. Como se pode verificar da análise da Tabela 10, o aluno evidenciou a totalidade dos conceitos de programação estudados e soube utilizá-los nos momentos e alturas corretas do jogo. Este projeto apresenta uma estrutura de 7 cenários de palco e 10 *sprites* (Tabela 11). A interface gráfica é simples e agradável, demonstrando cuidados ao nível estético (Figura 13). Identifiquei (e informei o aluno) alguns *bugs* que fazem com que por vezes o jogo bloqueie. No entanto o aluno nunca chegou a apresentar uma versão melhorada do mesmo.

Snake	
Comunicação e sincronização	45
Ciclos	4
Declarações condicionais	8
Interação com utilizador	12
Lógica Booleana	8
Variáveis	5
Números aleatórios	4

Estrutura	
Nº de cenários do palco	7
Nº de <i>sprites</i>	10

Tabelas 10 e 11 – Conceitos de programação e estrutura do projeto *Snake*.

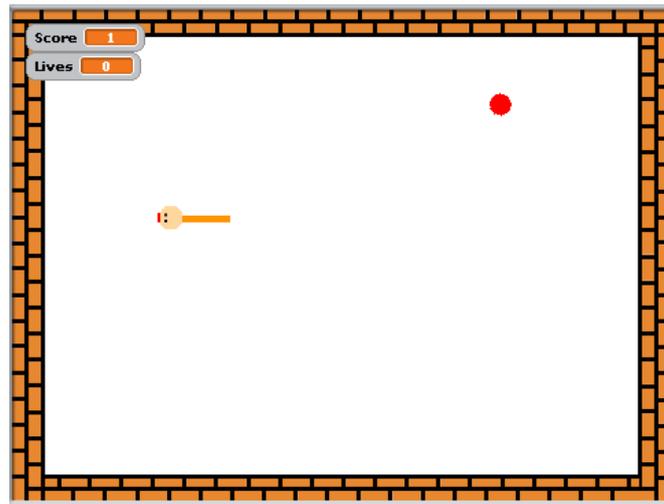


Figura 13 – Interface do Jogo *Snake*

3.2.1.5 Projeto Jogo do *Noddy*

Jogo desenvolvido por uma aluna com grandes dificuldades na área da programação (identificada durante a fase de observação com falta de motivação e interesse pela programação, fazendo parte do grupo de alunos que só vieram para a disciplina porque no ano anterior lhes tinham dito que era fácil e podiam tirar boas notas). Apesar das suas limitações a aluna esforçou-se e conseguiu criar um pequeno jogo de forma a evidenciar a totalidade dos conceitos de programação estudados. O jogo (Figura 14) consiste em ultrapassar um percurso sem tocar nas paredes. Esta aluna durante a fase de observação nas aulas de Pascal não era capaz de programar de forma autónoma, socorrendo-se da ajuda e trabalho dos colegas para poder resolver os exercícios. Com o *Scratch*, consegui que a aluna de forma autónoma desenvolvesse o seu projeto, que apesar de ser simples contemplou 6 dos 7 requisitos de Maloney (Tabela 12). Não utilizou comandos de números aleatórios que poderiam ter sido introduzidos neste jogo aumentando a sua jogabilidade e interesse, pois ao fim de jogar o jogo não nos sentimos motivados a voltar a jogá-lo, pois é bastante monótono. No entanto a aluna conseguiu programar, ou seja, desenvolver o raciocínio lógico-abstrato, condição fundamental da programação. O projeto apresenta uma estrutura de 10 cenários de palco e 6 *sprites* (Tabela 13).

Jogo do <i>Noddy</i>	
Comunicação e sincronização	62
Ciclos	3
Declarações condicionais	14
Interacção com utilizador	8
Lógica Booleana	14
Variáveis	2
Números aleatórios	0

Estrutura	
Nº de cenários do palco	10
Nº de <i>sprites</i>	6

Tabelas 12 e 13 – Conceitos de programação e estrutura do projecto Jogo do *Noddy*.



Figura 14 – Interface do Jogo do *Noddy*

3.2.1.6 Projeto Pika Thunder

Jogo bastante divertido e aliciante para os jogadores da faixa etária a que se destina (crianças entre os 6 a 10 anos - a aluna desenvolveu o jogo para a sua irmã mais nova de 7 anos que ia testando o jogo à medida que ia sendo desenvolvido). Durante a fase de observação tinha identificado esta aluna como fazendo parte do grupo de alunos com dificuldades na área da programação, no entanto no segundo período, conseguiu superar as suas lacunas e mostrou-se bastante motivada no desenvolvimento deste jogo. O projecto revela que foram investidas

largas horas no seu desenvolvimento, e há medida que o vamos jogando, torna-se mais complexo e interessante. A aluna evidenciou todos os conceitos de programação estudados e soube aplica-los nos momentos e altura devidos (ver Tabela 14). O jogo está bem dimensionado e apresenta uma estrutura de 4 cenários de palco e 54 *sprites* (Tabela 15). A interface é simples mas cuidada, sendo adequada à faixa etária a que se destina, como podemos verificar através da Figura 15.

Jogo <i>Pika Thunder</i>	
Comunicação e sincronização	305
Ciclos	31
Declarações condicionais	82
Interacção com utilizador	21
Lógica Booleana	82
Variáveis	8
Números aleatórios	21

Estrutura	
Nº de cenários do palco	11
Nº de <i>sprites</i>	25

Tabelas 14 e 15 – Conceitos de programação e estrutura do projecto *Pika Thunder*.



Figura 15 – Interface do Jogo *Pika Thunder*

3.2.1.7 Projeto Jogo do Elementos

Este jogo juntamente com o projecto RPG, que vamos analisar mais à frente no ponto 3.2.1.25, são os dois jogos mais complexos em termos de cenários e número de *sprites* (Tabela 17) revelando ao mesmo tempo originalidade e cuidados com a programação. A aluna que desenvolveu este jogo investiu largas horas do seu tempo para conseguir concluir este projecto, que se revelou bastante agradável, muito original e com elevada jogabilidade. A aluna teve o cuidado de criar códigos de atalho que permitem recomeçar o jogo na posição em que o jogador perdeu, para além de revelar uma característica muito original que é o personagem principal ir mudando ao longo do jogo, adquirindo características dos quatro elementos, terra, ar, fogo e água. O jogo apresenta um enredo original como mostra a Figura 16. Com este trabalho a aluna conseguiu evidenciar os seus bons conhecimentos na área da programação tendo evidenciado 6 dos 7 requisitos de Maloney (Tabela 16). A aluna não utilizou comandos de números aleatórios que poderiam ainda aumentar mais a jogabilidade, introduzindo alguma aleatoriedade aos movimentos dos inimigos.

Jogo dos Elementos	
Comunicação e sincronização	482
Ciclos	232
Declarações condicionais	155
Interacção com utilizador	45
Lógica Booleana	155
Variáveis	19
Números aleatórios	0

Estrutura	
Nº de cenários do palco	30
Nº de <i>sprites</i>	67

Tabelas 16 e 17 – Conceitos de programação e estrutura do projecto Jogo dos Elementos.



Figura 16 – Interface do Jogo do Elementos

3.2.1.8 Projeto *Go Ask Alice*

Jogo bastante divertido e aliciante para os jogadores da faixa etária a que se destina (crianças entre os 6 a 10 anos). A aluna, à semelhança da aluna que desenvolveu o Jogo do *Noddy*, estava identificada, durante a fase de observação, como uma aluna com grandes dificuldades na área da programação. No entanto, conseguiu superar as suas dificuldades, demonstrou interesse (a aluna no segundo período começou a trazer o seu portátil para as aulas, que não foi prenda de Natal) e mostrou-se bastante motivada no desenvolvimento deste jogo. A aluna evidenciou todos os conceitos de programação estudados e soube aplica-los nos momentos e altura devidos (Tabela 18). O jogo apresenta uma estrutura de 8 cenários de palco e 53 *sprites* (Tabela 19) e a sua interface apresenta-se cuidada e adequada à faixa etária a que se destina (Figura 17).

<i>Go Ask Alice</i>	
Comunicação e sincronização	175
Ciclos	41
Declarações condicionais	51
Interacção com utilizador	5
Lógica Booleana	51
Variáveis	11
Números aleatórios	21

Estrutura	
Nº de cenários do palco	8
Nº de <i>sprites</i>	53

Tabelas 18 e 19 – Conceitos de programação e estrutura do projecto *Go Ask Alice*.



Figura 17 – Interface do jogo *Go ask Alice*

3.2.1.9 Projeto *Snake Five*

O aluno que desenvolveu este projecto, apresentou o mesmo tipo de jogo que o jogo *Snake*, apresentado no ponto 3.2.1.4, mas numa versão muito mais original. O jogo tem cinco versões com 5 diferentes objectivos e cinco formas diferentes de jogar este popular jogo, o aluno intitulou o seu jogo como o *Snake 5 em 1*, daí o nome *Snake Five*.

Com este projecto o aluno evidenciou os sete conceitos de Maloney (Tabela 20), revelando que se sente há vontade com os conceitos lógico-abstratos que a programação exige. O jogo apresenta uma estrutura de 8 palcos e 31 *sprites* (Tabela 21) e uma interface bastante simples e pouco cuidada (Figura 18), mas que resulta dada a elevada jogabilidade e originalidade do jogo.

<i>Snake Five</i>	
Comunicação e sincronização	140
Ciclos	24
Declarações condicionais	53
Interacção com utilizador	17
Lógica Booleana	53
Variáveis	13
Números aleatórios	30

Estrutura	
Nº de cenários do palco	8
Nº de <i>Sprites</i>	31

Tabelas 20 e 21 – Conceitos de programação e estrutura do projecto *Go Ask Alice*.

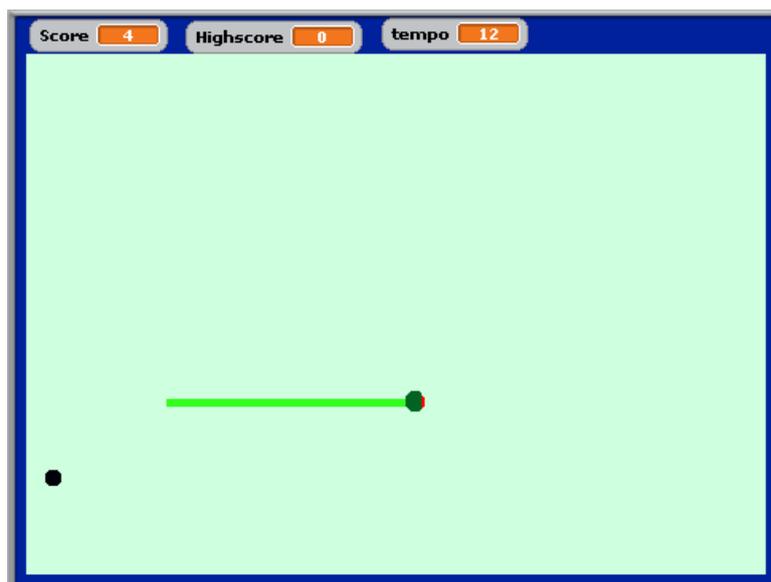


Figura 18 – Interface do Jogo *Snake Five*

3.2.1.10 Projeto *Titanic Beat Ups*

O aluno que desenvolveu este projecto é bastante interessado e gosta de programar no entanto tem grande dificuldade em estabelecer barreiras. Com este trabalho pretendia criar um jogo bastante complexo, no entanto ao programar não conseguiu implementar aquilo que tinha idealizado. Criou um jogo muito confuso com fraca jogabilidade. Desenvolveu muito código, se bem que muitos *scripts* são replicações de código de *sprite* para *sprite*, com muitas variáveis, tendo-se revelado no final um projeto muito difícil de analisar. Apesar de todos os pontos negativos o aluno conseguiu demonstrar que adquiriu os conceitos básicos de programação definidos por Maloney (Tabela 22). O seu projeto apresenta uma estrutura de 9 palcos e 35 *sprites* (Tabela 23). Não demonstrou grandes cuidados com o aspecto gráfico tendo criado uma interface simples e também um pouco confusa (Figura 19).

<i>Titanic Beat Ups</i>	
Comunicação e sincronização	233
Ciclos	141
Declarações condicionais	541
Interacção com utilizador	34
Lógica Booleana	544
Variáveis	148
Números aleatórios	9

Estrutura	
Nº de cenários do palco	9
Nº de <i>Sprites</i>	35

Tabelas 22 e 23 – Conceitos de programação e estrutura do projecto *Titanic Beat Ups*.



Figura 19 – Interface do Jogo *Titanic Beat Ups*

3.2.1.11 Projeto Jogo de Voleibol

A aluna que desenvolveu este projeto, à semelhança da aluna referida no ponto 3.2.1.2, era uma aluna que revelava algumas dificuldades com a programação. Apesar de ser ajudada pelos colegas a aluna demonstrou interesse e quis desenvolver o projeto por si própria. Como podemos constatar através da análise da Tabela 24, a aluna conseguiu evidenciar a totalidade dos conceitos de programação estudados. O projeto desenvolvido foi um jogo simples com fraca jogabilidade, em que o movimento da bola não estava harmonioso, no entanto dadas as suas limitações na área da programação, acabou por desenvolver um algoritmo simples e de certa forma eficaz, que permitiu resolver o problema que tinha pela frente.

A aluna conseguiu superar as suas dificuldades, demonstrou interesse e mostrou-se bastante motivada no desenvolvimento deste jogo, apresentando várias dúvidas ao longo da

minha intervenção no segundo período. O jogo apresenta uma estrutura de 5 cenários de palco e 15 *sprites* (Tabela 25) e a sua interface apresenta-se simples e podia ter tido mais cuidada (Figura 20).

Jogo de Voleibol	
Comunicação e sincronização	50
Ciclos	14
Declarações condicionais	24
Interacção com utilizador	10
Lógica Booleana	24
Variáveis	5
Números aleatórios	7

Estrutura	
Nº de cenários do palco	5
Nº de <i>Sprites</i>	15

Tabelas 24 e 25 – Conceitos de programação e estrutura do projecto Jogo de Voleibol.

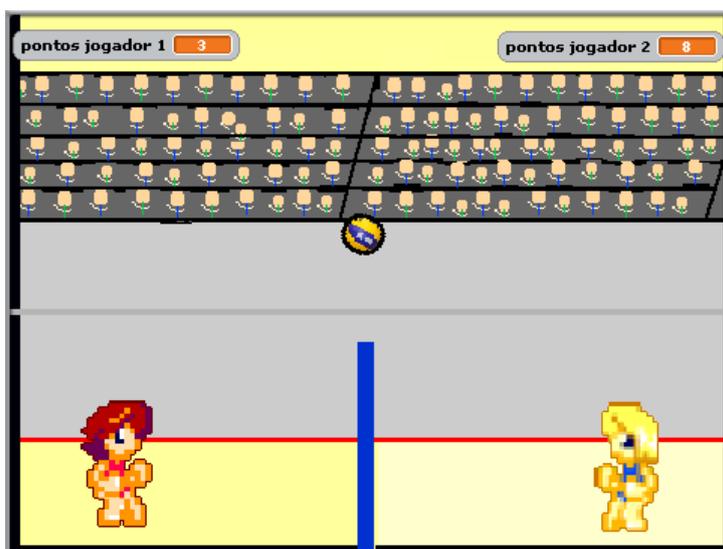


Figura 20 – Interface do Jogo de Voleibol

3.2.1.12 Projeto *Tetris*

Jogo desenvolvido por um dos melhores alunos da turma no que toca à programação. Peca pela pouca originalidade e acredito que este aluno, dadas as suas capacidades podia fazer muito melhor. No entanto no que toca à programação apresenta um código extremamente bem

estruturado e cuidado. Revela algoritmos bastante interessantes e originais. A forma como resolveu os problemas que se depararam pela frente demonstra que o aluno domina na plenitude todos os conceitos de programação estudados, como o demonstra a Tabela 26. Desenvolveu um jogo com uma estrutura de 5 cenários de palco e 16 *sprites* (Tabela 27), apresentando uma interface gráfica simples e eficaz (Figura 21). Este jogo foi o único projeto que conseguiu colocar a funcionar perfeitamente uma lista com as 10 melhores pontuações. Todos os alunos implementaram este requisito, mas somente este aluno conseguiu que funcionasse a 100%.

<i>Tetris</i>	
Comunicação e sincronização	86
Ciclos	13
Declarações condicionais	92
Interacção com utilizador	43
Lógica Booleana	95
Variáveis	25
Números aleatórios	2

Complexidade	
Nº de cenários do palco	5
Nº de <i>Sprites</i>	16

Tabelas 26 e 27 – Conceitos de programação e estrutura do projecto *Tetris*.

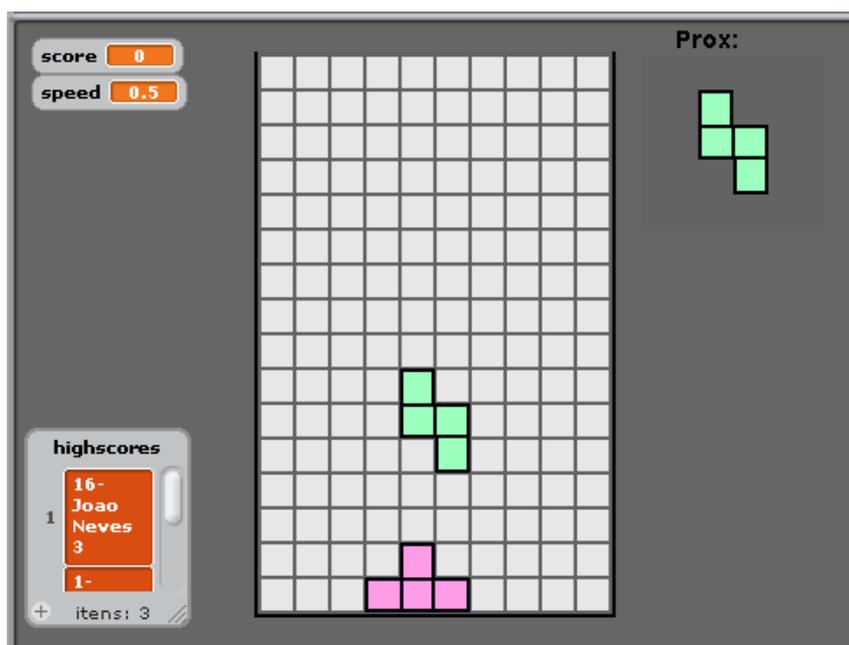


Figura 21 – Interface do Jogo *Tetris*

3.2.1.13 Projeto *Super Slime Soccer*

Este jogo é um *remix* (versão modificada de um projeto partilhado no site do *Scratch* internacional²) de um jogo publicado. No entanto o aluno teve a capacidade de adaptar o código e dar-lhe o seu cunho pessoal. O aluno que desenvolveu este projeto tem bons conhecimentos na área da programação e conseguiu evidenciá-los na realização deste jogo (Tabela 28). O jogo apresenta uma estrutura de 3 cenários de palco e 30 *sprites* (Tabela 29). A interface é simples no entanto funcional e agradável (Figura 22).

O *remix* de jogos é uma prática usual dentro da comunidade de programadores em *Scratch*, inclusive a prática de *remixing* é encorajada pelos criadores do *Scrath*, fazendo referência ao carácter mais social da aplicação, segundo Resnich et al (2009) o conceito de partilha encontra-se embutido na interface da aplicação através de um botão de partilha, o Site internacional do *Scratch* tem servido como uma ferramenta de trabalho colaborativo. Os seus membros estão constantemente a pedir emprestado imagens e programas, adaptando-os ou construindo a partir de ideias pré-existentes. Mais de 15% dos projetos existente no site do *Scratch* são *remixes* de outros projetos que já existiam publicados.

<i>Super Slime soccer</i>	
Comunicação e sincronização	158
Ciclos	42
Declarações condicionais	110
Interacção com utilizador	12
Lógica Booleana	112
Variáveis	16
Números aleatórios	1

Complexidade	
Nº de cenários do palco	3
Nº de <i>Sprites</i>	30

Tabelas 28 e 29 – Conceitos de programação e estrutura do projecto *Super Slime Soccer*.

² <http://scratch.mit.edu/projects/2931898/>

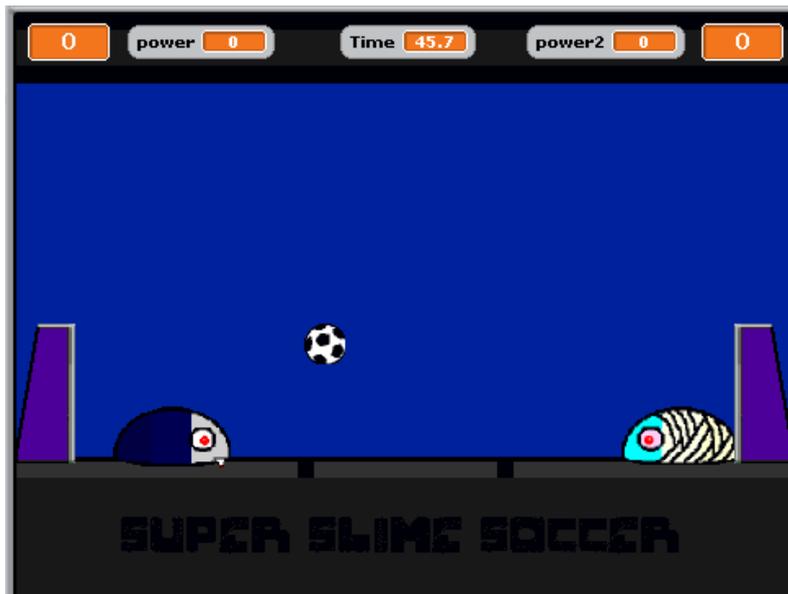


Figura 22 – Interface do Jogo *Super Slime Soccer*

3.2.1.14 Projeto Tiro ao Alvo

Este projeto foi desenvolvido por um aluno com bons conhecimentos na área da programação, o aluno demonstrou criatividade e desenvolveu um jogo com boa jogabilidade. Como se pode verificar da análise da Tabela 30, o aluno evidenciou a totalidade dos conceitos de programação estudados e soube utilizá-los nos momentos e alturas corretas do jogo. Este projeto apresenta uma estrutura de três palcos e doze *sprites* (Tabela 31). A interface gráfica é simples e agradável, demonstrando cuidados ao nível estético (Figura 23).

Tiro ao Alvo	
Comunicação e sincronização	31
Ciclos	13
Declarações condicionais	12
Interacção com utilizador	3
Lógica Booleana	29
Variáveis	7
Números aleatórios	1

Estrutura	
Nº de cenários do palco	3
Nº de <i>Sprites</i>	12

Tabelas 30 e 31 – Conceitos de programação e estrutura do projecto Tiro ao Alvo.



Figura 23 – Interface do Jogo Tiro ao Alvo

3.2.1.15 Projeto *Portal*

À semelhança do projeto apresentado no ponto 3.2.1.13, este jogo é um *remix* de outro jogo que se encontra publicado no site internacional do *Scratch*, no entanto o aluno teve a capacidade de adaptar o código e dar-lhe o seu cunho pessoal. O aluno que desenvolveu este projeto tem bons conhecimentos na área da programação e conseguiu evidenciá-los na realização deste jogo, como se pode verificar através da análise da Tabela 32. O jogo apresenta uma estrutura de 15 cenários de palco e 33 *sprites*, (Tabela 33) mostrando que foram investidas largas horas no desenvolvimento deste jogo. No que toca à parte gráfica o aluno criou os próprios cenários, que apesar de simples são eficientes para o tipo de jogo em causa (Figura 24).

³ <http://scratch.mit.edu/projects/12378597/>

<i>Portal</i>	
Comunicação e sincronização	235
Ciclos	77
Declarações condicionais	166
Interacção com utilizador	29
Lógica Booleana	176
Variáveis	39
Números aleatórios	0

Estrutura	
Nº de cenários do palco	15
Nº de <i>Sprites</i>	33

Tabelas 32 e 33 – Conceitos de programação e estrutura do projecto *Portal*.

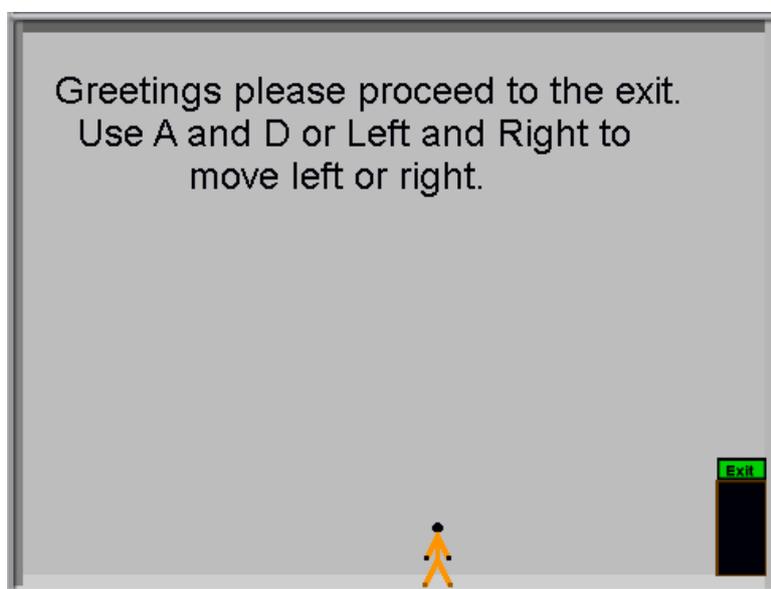


Figura 24 – Interface do Jogo *Portal*

3.2.1.16 Projeto Death Stars! Run!

Jogo bastante criativo e com excelente jogabilidade. O aluno que desenvolveu este jogo foi identificado na fase de observação como um aluno com algumas dificuldades na área da programação, com o Pascal o aluno não tinha sido capaz de desenvolver o raciocínio lógico/abstracto que a programação tanto exige. Na minha opinião este foi o aluno que mais ganhou com a introdução do *Scratch*, tendo-se notado uma grande evolução no aluno em termos de aprendizagem dos conceitos de programação (Tabela 32). No entanto, neste jogo faria todo o sentido introduzir os comandos de números aleatórios para que as estrelas da morte se

movessem de forma aleatória sempre que o jogo fosse iniciado. Apesar deste pormenor de programação o aluno criou um excelente jogo com 20 cenários de palco e 32 *sprites* (Tabela 33), bastante original e muito cuidado no que toca ao aspecto gráfico (Figura 19).

Death Stars! Run!	
Comunicação e sincronização	212
Ciclos	45
Declarações condicionais	46
Interacção com utilizador	4
Lógica Booleana	46
Variáveis	2
Números aleatórios	0

Estrutura	
Nº de cenários do palco	20
Nº de <i>Sprites</i>	32

Tabelas 34 e 35 – Conceitos de programação e estrutura do projecto *Death Stars! Run!*.

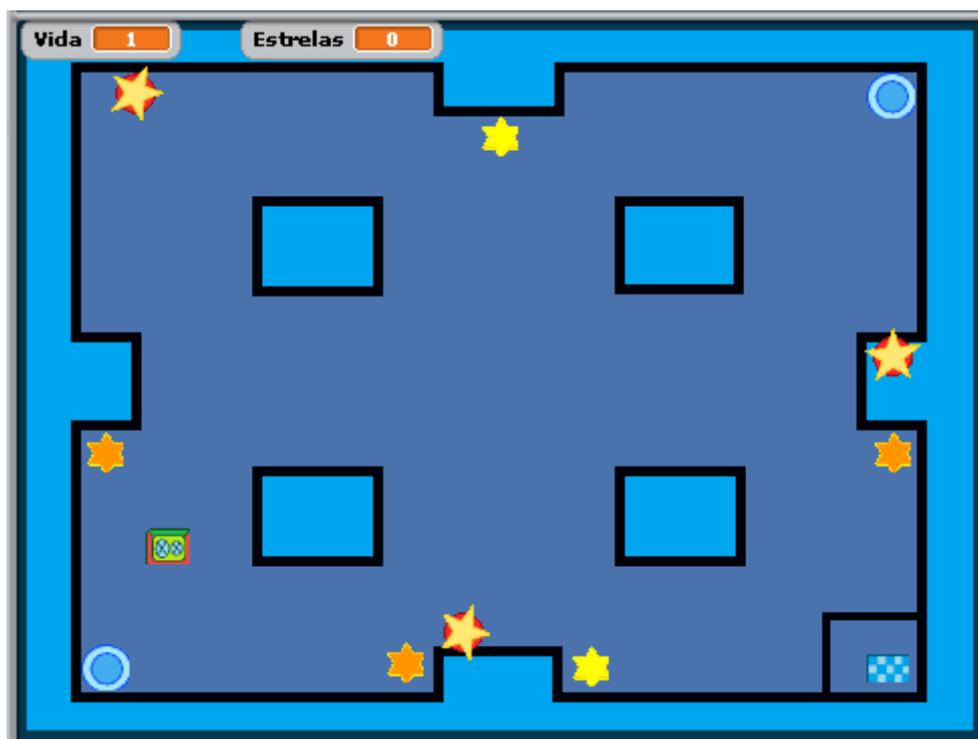


Figura 25 – Interface do Jogo *Death Stars! Run!*

3.2.1.17 Projeto Jogo de Carros

Este jogo foi desenvolvido por um aluno com bons conhecimentos na área da programação, o aluno demonstrou criatividade e desenvolveu um jogo com boa jogabilidade. Como se pode verificar da análise da Tabela 36, o aluno evidenciou a totalidade dos conceitos de programação estudados e soube utilizá-los nos momentos e alturas corretas do jogo. Este projeto apresenta uma estrutura de 13 palcos e 16 *sprites* (Tabela 37). O facto de não utilizar comandos de números aleatórios, não compromete a qualidade do projeto pois não faz sentido em algum momento do jogo utilizar estes comandos. A interface gráfica é simples e agradável, demonstrando cuidados ao nível estético (Figura 26).

Jogo de Carros	
Comunicação e sincronização	178
Ciclos	40
Declarações condicionais	154
Interacção com utilizador	20
Lógica Booleana	162
Variáveis	26
Números aleatórios	0

Estrutura	
Nº de cenários do palco	13
Nº de <i>Sprites</i>	16

Tabelas 36 e 37 – Conceitos de programação e estrutura do projecto Jogo de Carros.



Figura 26 – Interface Jogo de Carros

3.2.1.18 Projeto Super Space Mário

Jogo bastante original e com excelente jogabilidade. O aluno que desenvolveu este projeto tem muito bons conhecimentos na área da programação e conseguiu evidenciá-los como se pode verificar através da análise da Tabela 38, o aluno conseguiu criar um jogo muito bom com excelente jogabilidade. O facto de não ter utilizado comandos de números aleatórios, não comprometeu a qualidade do projeto pois não fazia sentido em algum momento do jogo utilizar estes comandos. O jogo apresenta uma estrutura de 3 cenários de palco e 21 *sprites* (Tabela 39) com uma interface gráfica simples mas eficaz para o jogo em causa (Figura 27).

<i>Super Space Mario</i>	
Comunicação e sincronização	69
Ciclos	7
Declarações condicionais	22
Interacção com utilizador	6
Lógica Booleana	22
Variáveis	7
Números aleatórios	0

Estrutura	
Nº de cenários do palco	3
Nº de <i>Sprites</i>	21

Tabelas 38 e 39 – Conceitos de programação e estrutura do projeto Jogo de Carros.



Figura 27 – Interface do Jogo *Super Space Mario*

3.2.1.19 Projeto Jogo da Princesas

Jogo bastante simples, desenvolvido por uma aluna com algumas dificuldades na área da programação. Esta aluna estava identificada como uma aluna com falta de motivação e interesse pela programação, À semelhança da aluna mencionada no ponto 3.2.1.8 fazia parte do grupo de alunas que só vieram para esta disciplina porque no ano anterior lhes tinham dito que era fácil e podiam tirar boas notas. Apesar das suas limitações a aluna esforçou-se e conseguiu criar um jogo de forma a evidenciar a totalidade dos conceitos de programação estudados (Tabela 40). À medida que se vai evoluindo no jogo este torna-se mais interessante e complexo, apresentando uma estrutura de 7 cenários de palco e 15 *sprites* (Tabela 41). A interface é simples e pouco cuidada como se pode verificar através da Figura 28.

Jogo das Princesas	
Comunicação e sincronização	45
Ciclos	9
Declarações condicionais	24
Interacção com utilizador	15
Lógica Booleana	27
Variáveis	5
Números aleatórios	2

Estrutura	
Nº de cenários do palco	7
Nº de <i>Sprites</i>	15

Tabelas 40 e 41 – Conceitos de programação e estrutura do projecto Jogo das Princesas.

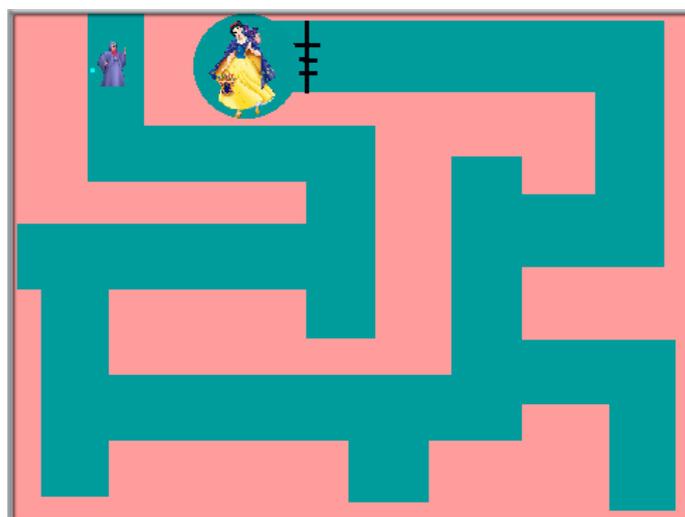


Figura 28 – Interface do Jogo das Princesas

3.2.1.20 Projeto Portal Still Alive

Este jogo é um *remix* de outro jogo que se encontra publicado no site internacional do *Scratch*, no que toca à parte gráfica (Figura 29) o aluno dá os créditos ao autor do projeto mencionado anteriormente no entanto o aluno desenvolveu todo o código sem ter feito qualquer tipo de adaptação de código pré-existente. O aluno que desenvolveu este projeto tem muito bons conhecimentos na área da programação e conseguiu evidenciá-los na realização deste jogo (Tabela 40). Através da análise das Tabelas 32 e 42, podemos verificar que o aluno com um menor número de comandos conseguiu criar o mesmo jogo que o aluno que tinha criado o jogo *Portal* analisado no ponto 3.2.1.15 deste relatório. O facto de não utilizar comandos de números aleatórios, não compromete a qualidade do projeto pois não faz sentido em algum momento do jogo utilizar este tipo de comandos. O jogo apresenta uma estrutura de 19 cenários de palco e 15 *sprites* (Tabela 43).

<i>Portal Still Alive</i>	
Comunicação e sincronização	110
Ciclos	25
Declarações condicionais	82
Interacção com utilizador	9
Lógica Booleana	85
Variáveis	8
Números aleatórios	0

Estrutura	
Nº de cenários do palco	19
Nº de <i>Sprites</i>	15

Tabelas 42 e 43 – Conceitos de programação e estrutura do projeto *Portal Still Alive*.

⁴ <http://scratch.mit.edu/projects/647875/>

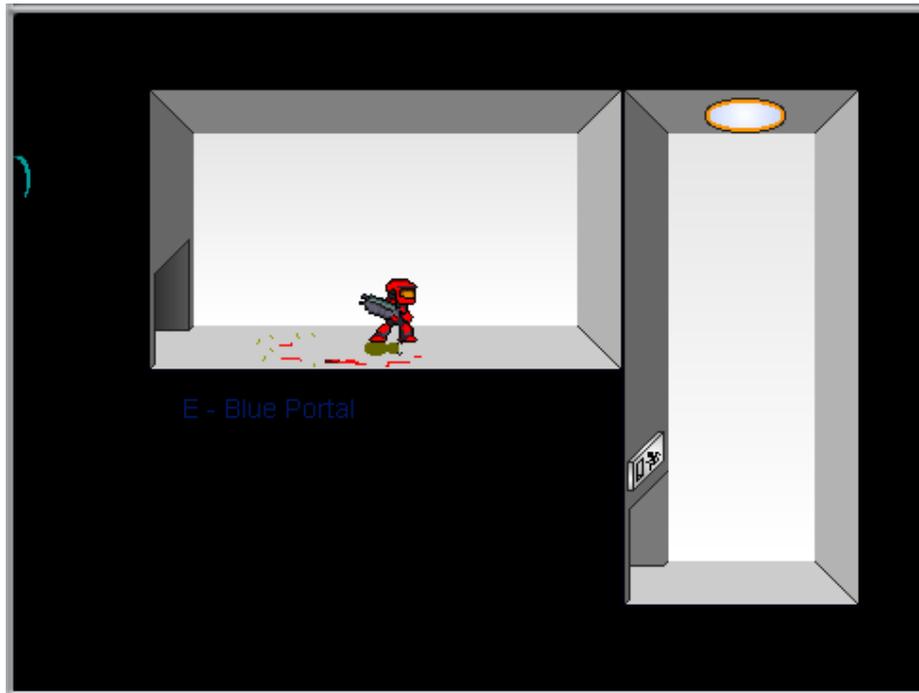


Figura 29 – Interface do Jogo *Portal Still Alive*

3.2.1.21 Projeto Dragon Ball Arena

Jogo simples, desenvolvido por um aluno com grandes dificuldades na área da programação. Este aluno estava identificado (fase de observação) como um aluno com falta de motivação e interesse pela programação. Apesar das suas limitações o aluno esforçou-se, e com a ajuda dos colegas conseguiu criar um jogo de forma a evidenciar a totalidade dos conceitos de programação estudados (Tabela 44). O facto de não utilizar comandos de números aleatórios, não compromete a qualidade do projeto pois não faz sentido em algum momento do jogo utilizar estes comandos pois trata-se de um jogo de simulação de luta entre dois oponentes. O jogo apresenta uma estrutura de 5 cenários de palco e 20 *sprites* (Tabela 45). A interface é simples mas eficaz para o tipo de jogo em causa (Figura 30).

<i>Dragon Ball Arena</i>	
Comunicação e sincronização	113
Ciclos	9
Declarações condicionais	162
Interacção com utilizador	92
Lógica Booleana	162
Variáveis	8
Números aleatórios	0

Estrutura	
Nº de cenários do palco	5
Nº de <i>Sprites</i>	20

Tabelas 44 e 45 – Conceitos de programação e estrutura do projecto *Portal Still Alive*.



Figura 30 – Interface do Jogo *Dragon Ball Arena*

3.2.1.22 Projeto *Time Breaker*

Este projeto foi desenvolvido por um aluno com bons conhecimentos na área da programação, o aluno demonstrou criatividade e desenvolveu um jogo com boa jogabilidade. Como se pode verificar da análise da Tabela 46, o aluno evidenciou a totalidade dos conceitos de programação estudados e soube utilizá-los nos momentos e alturas corretas do jogo. Este projeto apresenta uma estrutura de 5 palcos e 35 *sprites* (Tabela 47). A interface gráfica é simples e agradável, demonstrando cuidados ao nível estético (Figura 31).

<i>Time Breaker</i>	
Comunicação e sincronização	65
Ciclos	58
Declarações condicionais	55
Interacção com utilizador	9
Lógica Booleana	55
Variáveis	2
Números aleatórios	3

Estrutura	
Nº de cenários do palco	5
Nº de <i>Sprites</i>	35

Tabelas 46 e 47 – Conceitos de programação e estrutura do projeto *Time Breaker*.

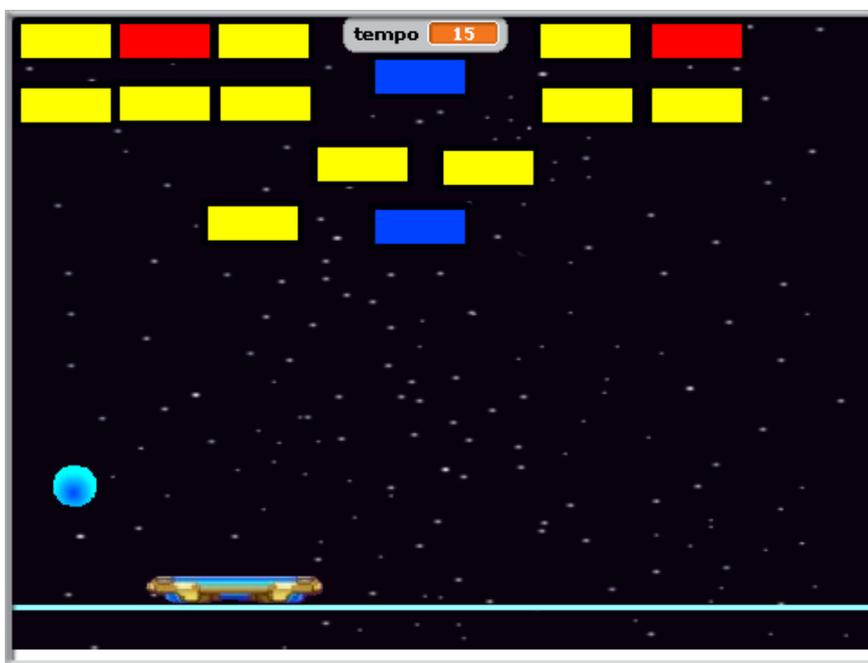


Figura 31 – Interface do Jogo *Time Breaker*

3.2.1.23 Projeto Pokémon

Este jogo foi desenvolvido por um aluno com bons conhecimentos na área da programação, o aluno demonstrou criatividade e desenvolveu um jogo com boa jogabilidade. Como se pode verificar da análise da Tabela 48, o aluno evidenciou a totalidade dos conceitos de programação estudados e soube utilizá-los nos momentos e alturas corretas do jogo. Este projeto apresenta uma estrutura de 7 palcos e 12 *sprites* (Tabela 49). A interface gráfica é simples e agradável, demonstrando cuidados ao nível estético (Figura 32).

<i>Pokémon</i>	
Comunicação e sincronização	42
Ciclos	23
Declarações condicionais	52
Interacção com utilizador	39
Lógica Booleana	52
Variáveis	6
Números aleatórios	0

Estrutura	
Nº de cenários do palco	7
Nº de <i>Sprites</i>	12

Tabelas 48 e 49 – Conceitos de programação e estrutura do projeto *Pokémon*.



Figura 32 - Interface do Jogo *Pokémon*

3.2.1.24 Projeto *Dragon Quest*

Jogo simples mas com uma jogabilidade agradável. O aluno com este projeto conseguiu evidenciar todos os conceitos de programação estudados (Tabela 50). Este aluno tem bons conhecimentos na área da programação e tem capacidade para fazer mais. O aluno investiu poucas horas neste projeto e se tivesse trabalhado mais poderia ter criado um projeto muito mais aliciante. O jogo começa de forma bastante simples, mas à medida que vai decorrendo torna-se um pouco mais complexo, apresenta uma estrutura de 12 cenários de palco e 25 *sprites* (Tabela 51). A interface apresenta-se muito pouco cuidada revelando o pouco trabalho do aluno no desenvolvimento deste projeto (Figura 33).

<i>Dragon Quest</i>	
Comunicação e sincronização	141
Ciclos	21
Declarações condicionais	64
Interacção com utilizador	16
Lógica Booleana	68
Variáveis	3
Números aleatórios	2

Estrutura	
Nº de cenários do palco	12
Nº de <i>Sprites</i>	25

Tabelas 50 e 51 – Conceitos de programação e estrutura do projeto *Dragon Quest*.

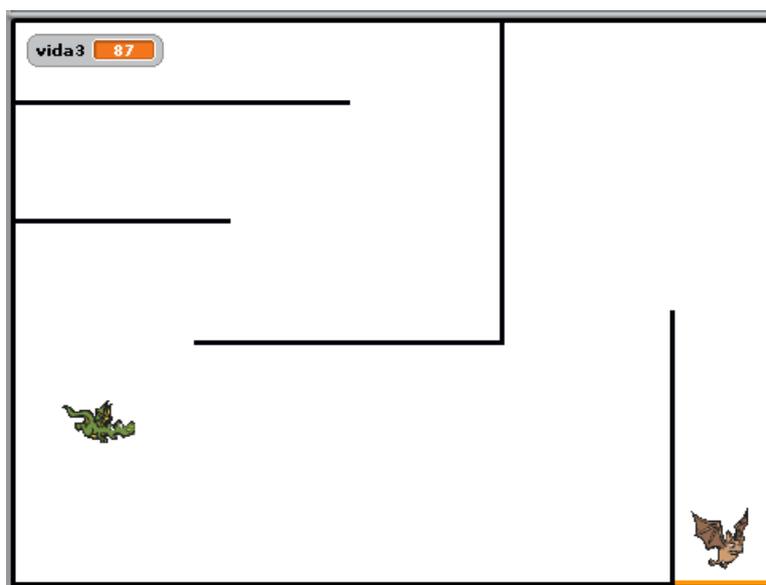


Figura 33 – Interface do Jogo *Dragon Quest*

3.2.1.25 Projeto RPG

Como já tinha sido referido no ponto 3.2.1.7, este jogo juntamente com o projeto Jogo dos elementos, é um dos dois jogos mais complexos em termos gráficos revelando ao mesmo tempo originalidade e cuidados com a programação. O aluno que desenvolveu este jogo investiu largas horas do seu tempo para conseguir concluir este projeto, que se revelou bastante agradável, muito original e com elevada jogabilidade. O aluno apresenta bons conhecimentos na área da programação e conseguiu evidenciá-los com a execução deste jogo (Tabela 52). O jogo apresenta uma estrutura de 10 cenários e 124 *sprites* (Tabela 53). A interface gráfica apresenta-se cuidada tendo alguns cenários sido adaptados de jogos de computador pré-existentis (Figura 34).

RPG	
Comunicação e sincronização	591
Ciclos	38
Declarações condicionais	133
Interacção com utilizador	16
Lógica Booleana	138
Variáveis	6
Números aleatórios	6

Estrutura	
Nº de cenários do palco	10
Nº de <i>Sprites</i>	124

Tabelas 52 e 53 – Conceitos de programação e estrutura do projecto RPG.



Figura 34 – Interface do Jogo RPG

3.2.1.26 Projeto Super-homem no espaço

Jogo bastante original e divertido, desenvolvido por uma aluna que durante a fase de observação apresentava algumas dificuldades na área da programação. Apesar das suas limitações a aluna esforçou-se e conseguiu criar um jogo de forma a evidenciar a totalidade dos conceitos de programação estudados (Tabela 54). À medida que se vai evoluindo no jogo este torna-se mais interessante e complexo, apresentando uma estrutura de 39 cenários de palco e 15 *sprites* (Tabela 55). A aluna investiu largas horas no desenvolvimento deste projeto apresentando uma interface gráfica maioritariamente desenvolvida por si (Figura 35).

Super-homem no espaço	
Comunicação e sincronização	168
Ciclos	65
Declarações condicionais	461
Interacção com utilizador	6
Lógica Booleana	462
Variáveis	18
Números aleatórios	430

Estrutura	
Nº de cenários do palco	39
Nº de <i>Sprites</i>	15

Tabelas 54 e 55 – Conceitos de programação e estrutura do projeto Super-homem no espaço.



Figura 35 – Interface do Jogo Super-homem no espaço

3.2.2 Avaliação e apresentação dos resultados dos trabalhos de grupo

Como já tinha sido referido, foi proposto aos alunos o desenvolvimento de um jogo colectivo para toda a turma, em que cada grupo iria ser responsável pela programação no *Scratch* de determinadas cenas de um guião desenvolvido pela turma do 12º E. Este projeto coletivo interdisciplinar fazia parte da avaliação do terceiro período, facto acordado e definido com o professor orientador cooperante. Na avaliação dos trabalhos de grupo, não avaliei unicamente o produto final desenvolvido pelo grupo, mas também todo o processo de desenvolvimento, através dos meus registos de reflexão ao longo da minha intervenção, tendo especial ênfase no interesse e empenho individual dos alunos, a colaboração entre os diferentes alunos e grupos. No final optou-se por atribuir uma nota de avaliação igual para todos os alunos do grupo, uma vez que para a avaliação do terceiro período também existia uma componente individual resultante do desenvolvimento dos vídeos do módulo multimédia.

O desenvolvimento dos trabalhos de grupo suportou-se no meu protótipo de um jogo previamente testado em termos de usabilidade. Nesta fase de avaliação dos trabalhos de grupo, avaliámos a extensão da utilização do protótipo no desenvolvimento dos trabalhos de grupo. De referir que os grupos não foram penalizados ou valorizados por se terem baseado mais ou menos no protótipo, mas sob o ponto de vista investigativo deste trabalho é pertinente avaliar até que ponto foi utilizado o protótipo no desenvolvimento dos trabalhos. É também importante salientar que a componente gráfica não foi alvo de avaliação, pois como já foi referido anteriormente todo o material multimédia tinha sido desenvolvido pela turma do 12º E.

3.2.2.1 Grupo 1

Os cinco elementos deste grupo de trabalho, três rapazes e duas raparigas, demonstraram sempre bastante interesse no desenvolvimento do seu projeto de grupo. Era um grupo de alunos bastante autónomo e raramente foi necessária a minha intervenção. Faziam parte deste grupo alunos com muito bons conhecimentos de programação, daí o facto de raramente necessitarem da minha ajuda. Este grupo de trabalho foi responsável por desenvolver a programação da cena 3 do guião do jogo, em que o personagem principal tinha que percorrer a biblioteca, matar os zombies que lhe aparecessem pela frente, tendo como objectivo reunir as folhas do livro onde constava a fórmula secreta com a cura para o vírus que transformava

humanos em zombies. A Figura 36 apresenta a interface da cena do jogo desenvolvida por este grupo de trabalho.



Figura 36 – Cena da biblioteca

Os alunos conseguiram implementar todos os requisitos apresentados no guião da cena, através de código bem estruturado e que facilmente foi integrado no jogo compilado com todas as cenas. Dados os seus muito bons conhecimentos na área de programação, recorreram ao protótipo de forma a orientarem a estrutura do seu trabalho, tal foi observado ao longo da minha intervenção nas aulas e através das conversas que íamos tendo. No entanto, em termos de implementação e programação os alunos foram capazes de desenvolver os seus próprios algoritmos e *scripts* de programação tal como se pode verificar através da Figura 37, que compara dois *scripts* relativos ao momento de retirar energia ao personagem principal (PP) quando toca no adversário.

Protótipo



Grupo 1 - Cena 3

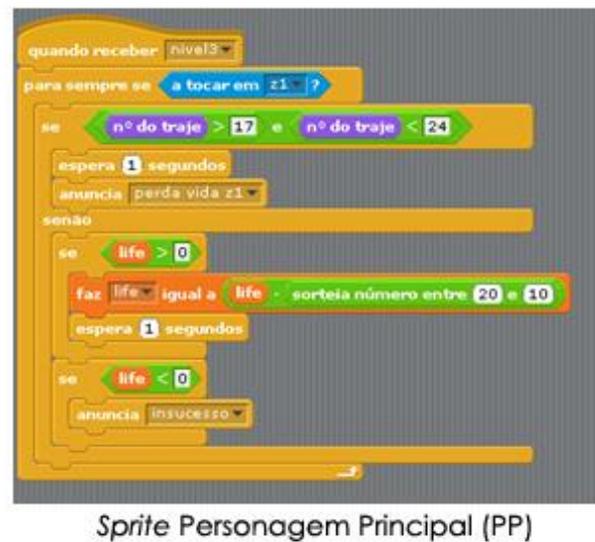


Figura 37 – *Scripts* do momento em que é retirada energia ao PP

No protótipo o *script* que retira energia ao personagem principal é sempre ao nível dos *sprites* adversários, no caso da Figura 37 refere-se a um *script* do fogo lançado por Molgror (adversário) que sempre que toca no Druida (PP) retira uma unidade de energia (variável Energia). Quando o Druida fica sem energia é anunciado o fim do jogo que é implementado ao nível dos *scripts* do PP. A retirada de energia aos adversários é implementada ao nível do *sprite* magia lançada pelo druida que sempre que toca no adversário retira energia ao mesmo. No caso do trabalho desenvolvido pelo grupo 1, os *scripts* que desempenham as mesmas três tarefas anteriores, são todos implementados ao nível do personagem principal, assim, sempre que este toca nos adversários (no caso da cena 3, são três zombies designados por z1, z2 e z3), se estiver a tocar com a régua (trajes do personagem principal entre 18 e 23) quem perde energia é o zombie, caso contrário quem perde energia (variável *life*) de forma aleatória entre 10 e 20 unidades é o personagem principal do jogo. Quando a energia do personagem principal for inferior a zero termina o jogo. Desta forma podemos concluir que o protótipo foi uma ferramenta útil para o grupo de desenvolvimento e que este foi capaz de dar o seu cunho pessoal ao código implementado.

3.2.2.2 Grupo 2

Este grupo de trabalho era constituído por cinco alunos, quatro rapazes e uma rapariga, que apesar de interessados pela programação por vezes dispersavam nas aulas e não se focavam no trabalho que tinham a desenvolver, sendo por vezes necessária a minha intervenção para que se focassem mais no trabalho de grupo e menos em outros projectos paralelos. Faziam parte deste grupo alunos com bons conhecimentos de programação, pelo que poucas foram as vezes que solicitaram o meu auxílio. Este grupo de trabalho foi responsável por desenvolver a programação da cena 4 do guião do jogo, em que o personagem principal tinha chegado à cantina, onde tinha que matar a funcionária que se tinha transformado em zombie. Pela frente tinha a tarefa de se desviar dos objectos que a funcionária atirava, que retiravam energia (pratos e bombas) e apanhar aqueles davam energia extra (bananas e brócolos) assim como as armas (facas e garfos) para as poder arremessar à funcionária, retirando-lhe energia e mantando-a, tendo como objectivo obter o ingrediente secreto que permitia sintetizar no laboratório o antídoto. A Figura 38 apresenta a interface da cena do jogo desenvolvida por este grupo de trabalho.

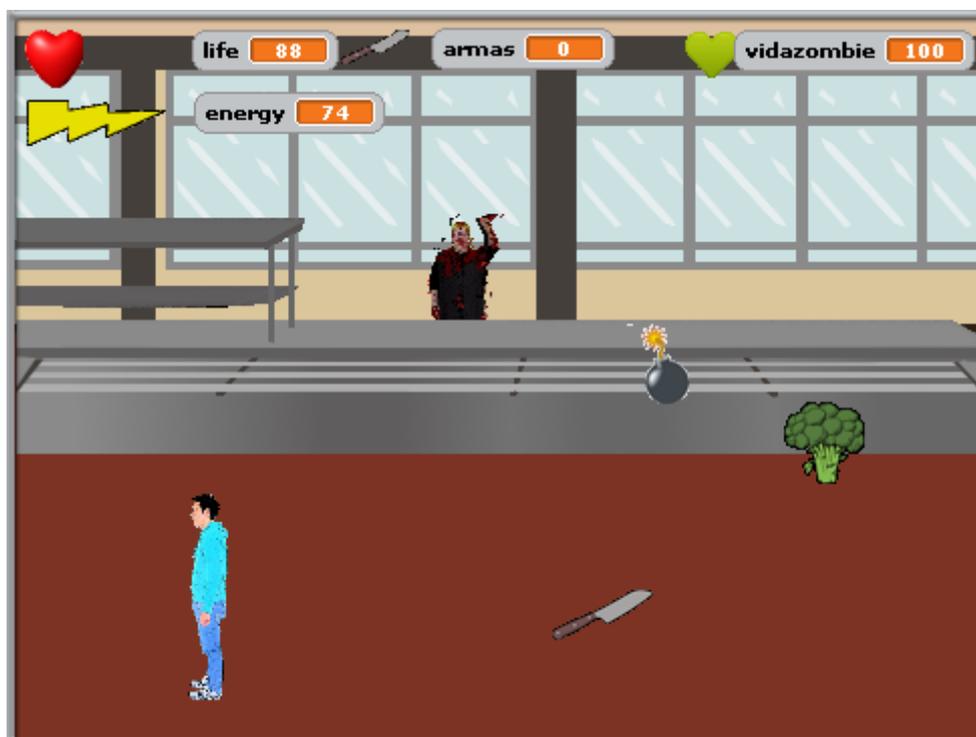


Figura 38 – Cena da cantina

Os alunos conseguiram implementar todos os requisitos apresentados no guião da cena, através de código bem estruturado e que facilmente foi integrado no jogo compilado com todas as cenas. Durante a minha fase de intervenção e através das conversas que ia tendo com os alunos deste grupo, apercebi-me que a maior utilidade do protótipo centrava-se no facto de permitir aos alunos terem uma visão de um jogo completo e permitir a estes estruturar o seu código para que mais tarde fosse fácil fazer a integração da sua cena no projeto global *Zschool*. Os alunos também mostraram especial interesse pelo protótipo no momento em que estavam a desenvolver o lançamento das bombas, pela funcionária zombie, e que atingiam o personagem principal retirando-lhe energia. No protótipo tinham descoberto uma situação semelhante quando o monstro lançava bolas de fogo que atingiam o druida retirando-lhe energia. A Figura 39 apresenta a comparação dos *scripts* relativos a esta situação.

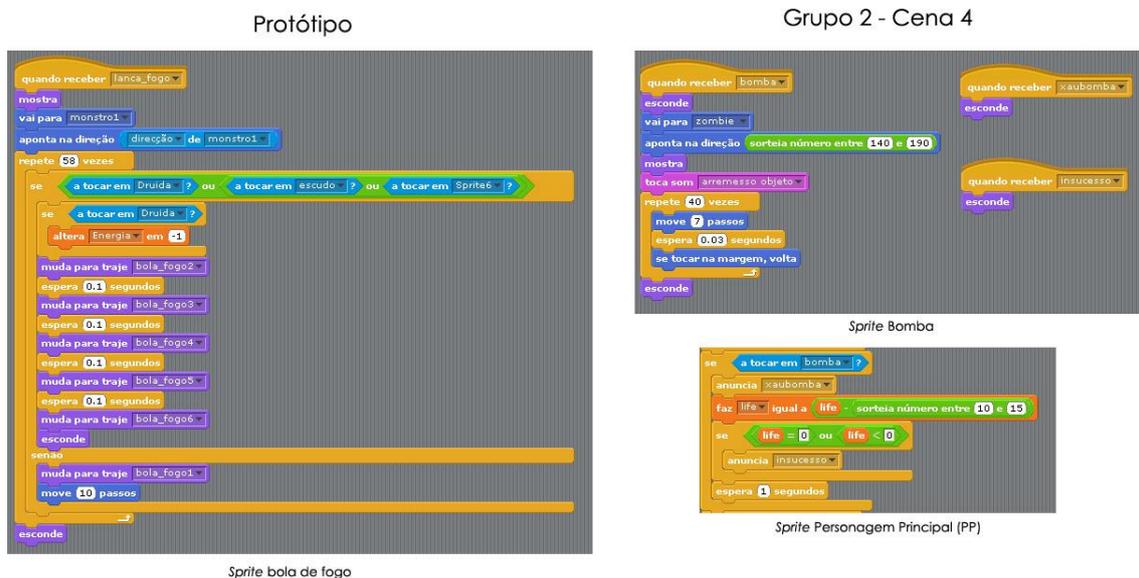


Figura 39 – *Scripts* do momento em que são lançadas bombas ao PP

Tanto no protótipo como na implementação por parte dos alunos, o movimento do elemento que retira energia, no caso do protótipo o *sprite* bola de fogo e no caso da cena da cantina o *sprite* bomba, é feito recorrendo a um ciclo que repete um determinado número de passos para que o elemento atravessasse todo o ecrã até embater no PP ou nas fronteiras do jogo. A implementação por parte do grupo 2, foi baseada no protótipo, no entanto os alunos deram o seu cunho pessoal ao fazerem o objecto apontar numa direção aleatória, de forma a descrever movimentos diagonais, enquanto que no protótipo a direção era sempre a mesma (horizontal), e quem se movia era o Sprite (Monstro) que lançava a bola de fogo. Outra situação interessante foi

o facto de os alunos implementarem o momento em que é retirada energia ao PP ao nível dos *scripts* do *sprite* PP, enquanto no protótipo esse momento é implementado ao nível do *sprite* bola de fogo. Mais uma vez, podemos concluir que o protótipo foi uma ferramenta útil para o grupo de desenvolvimento e que este foi capaz de dar o seu cunho pessoal ao código implementado.

3.2.2.3 Grupo 3

Este grupo de trabalho era composto por cinco alunos, três rapazes e duas raparigas, os conhecimentos na área da programação dos elementos do grupo variavam entre conhecimentos intermédios até avançados, pelo que foi agradável notar a colaboração dos alunos com mais conhecimentos ajudando aqueles que não se sentiam tão à vontade com programação. De salientar que apesar de haver uma diferença grande de conhecimentos entre os alunos, todos estavam interessados no desenvolvimento do seu projeto de grupo. Este grupo de trabalho estava responsável pelo desenvolvimento de três cenas: cena 5, laboratório de química, onde o personagem principal tinha de acertar na sequência correta de ingredientes do antídoto de forma a o poder sintetizar; cena 6, sucesso, cena final do jogo quando o PP conseguia alcançar o antídoto e curar todos os zombies que tinham invadido a ESNB, apresentando os créditos do jogo listando todos os intervenientes no desenvolvimento do projeto *Z-School*, desde alunos da turma E e B, a professores estagiários e professor orientador cooperante; cena 7, insucesso, sempre que o PP ficava sem energia era anunciada a cena de insucesso (*GAME OVER*), os zombies saíram vitoriosos e não foi possível salvar a ESNB. As Figuras 40 e 41 apresentam as interfaces das diferentes cenas desenvolvidas pelo grupo 3.



Figura 40 – Cena do laboratório de química



Figura 41 - Cenas de sucesso e insucesso

Os alunos conseguiram implementar todos os requisitos apresentados no guião das cenas, através de código bem estruturado e que facilmente foi integrado no jogo final. Durante a minha fase de intervenção e através das conversas que ia tendo com os alunos deste grupo, apercebi-me que a maior utilidade do protótipo centrava-se no facto de permitir aos alunos terem uma visão de um jogo completo e permitir a estes estruturar o seu código para que mais tarde fosse fácil fazer a integração da sua cena no projeto global jogo *Zschool*. Apesar de os alunos não terem encontrado no protótipo situações semelhantes à que tinham que desenvolver, como

já referi anteriormente, este foi útil para os alunos terem uma ideia global do projeto final motivando-os para que concluíssem o seu trabalho, parte integrante e fundamental do projeto final *Zschool*. Apesar de não se terem baseado no protótipo para desenvolver o seu código é importante referir o algoritmo desenvolvido pelo alunos para o jogo do laboratório de química, que à semelhança do desenvolvimento dos trabalhos individuais, vem comprovar que os alunos foram capazes de desenvolver o raciocínio lógico/abstracto que a programação exige, concluindo assim que os alunos aprenderam a programar com o *Scrath*. A Figura 42 apresenta o *script* do pequeno jogo do laboratório de química.

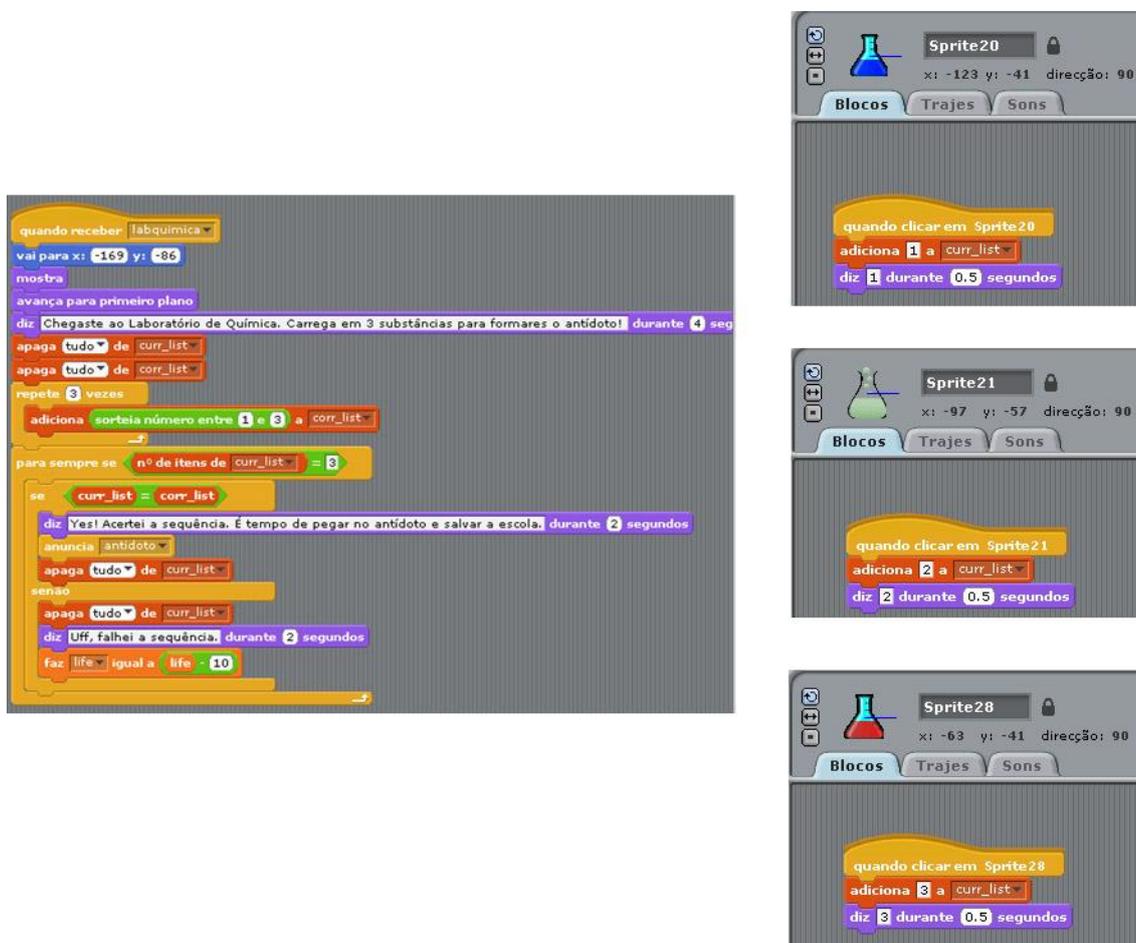


Figura 42 – *Script* do algoritmo do jogo do laboratório de química

Como podemos verificar da análise deste *script*, foram implementadas as sete premissas de Maloney analisadas para os jogos individuais: comunicação e sincronização através dos comandos “Quando receber” e “anuncia”; ciclos através dos ciclos “repete 3 vezes” e “para sempre se”; declarações condicionais através dos comandos “se””senão”; interação com utilizador através do comando “Quando clicar em”; lógica booleana através da comparação “curr_list = corr_list”;

variáveis através das listas de variáveis “*corr_list*” e “*curr_list*” assim como a variável “*life*”; e, números aleatórios através do comando “*sorteia número entre 1 e 3*”. Para além de terem utilizado todos os comandos referidos nos estudos de Maloney, os alunos criaram um algoritmo bastante interessante que passo a explicar. Sempre que é iniciado o jogo do laboratório de química é criada uma sequência correcta de ingredientes, através do ciclo “*repete 3 vezes*”. Em cada uma das passagens pelo ciclo é adicionado um valor aleatório entre 1 a 3 à lista “*corr_list*”. No final temos então a nossa sequência correcta pela qual devem ser seleccionados os ingredientes que se encontram em cima da bancada. De cada vez que um dos ingredientes é clicado, *sprites* 20, 21 e 28, é adicionado o seu código de ingrediente à lista que está a ser criada “*curr_list*”, assim que tiverem sido clicados três ingredientes, situação que está constantemente a ser verificada através do ciclo condicional “*para sempre se n° de itens de curr_list=3*”, vai-se verificando se a sequência seleccionada é igual à sequência correcta através da comparação “*curr_list=corr_list*”, se sim é dada uma mensagem de sucesso e aparece o antídoto, senão é dada uma mensagem de insucesso e é retirada vida ao PP, iniciando-se novamente todo o processo de selecção dos ingredientes.

3.2.2.4 Grupo 4

O grupo 4 de desenvolvimento, constituído por quatro rapazes e 2 raparigas, era o grupo com maior número de elementos, pelo que foi atribuído a este grupo de trabalho uma carga maior de trabalho. À semelhança do grupo 3, também neste grupo tínhamos alunos com diferentes níveis de conhecimento na área da programação, desde o elementar ao mais avançado. Este grupo de trabalho estava responsável pelo desenvolvimento de duas cenas do guião do jogo assim como todos os comandos de movimento do personagem principal. Tinham que desenvolver a cena 0, relativa à introdução do jogo, com o menu e informação sobre como jogar o *Zschool*, cena 1, relativa à entrada da ESNB, onde é feito o enquadramento do jogador no enredo do jogo e toda a parte de programação de movimentos do personagem principal, nomeadamente os movimentos para esquerda, direita, cima e baixo assim como as reguadas para a esquerda e direita. As Figuras 43 e 44, apresentam as interfaces da introdução e entrada da ESNB, respectivamente.

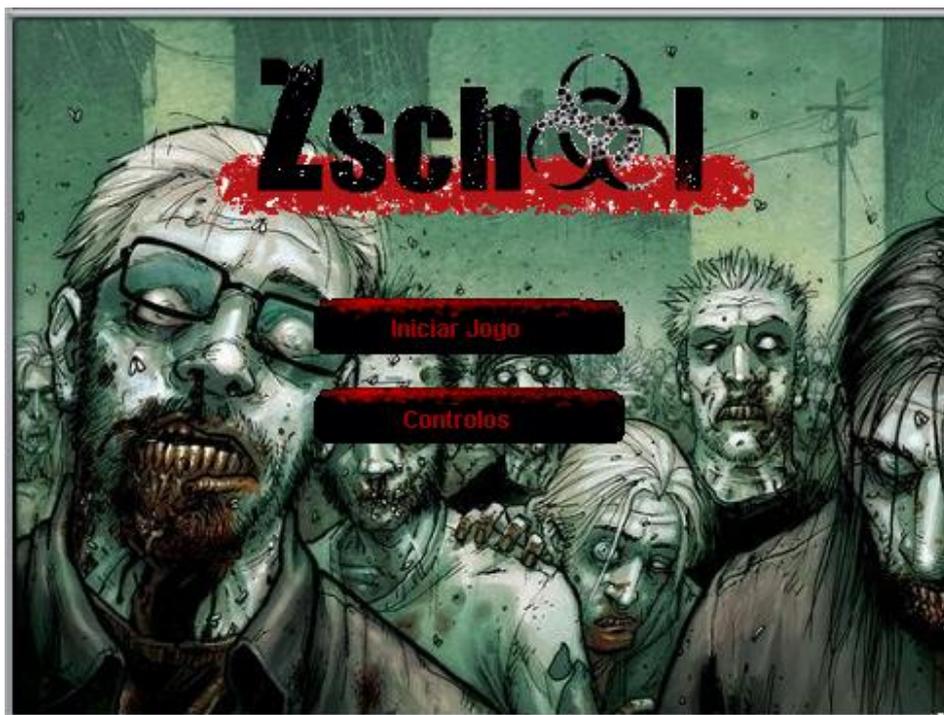


Figura 43 – Cena da introdução

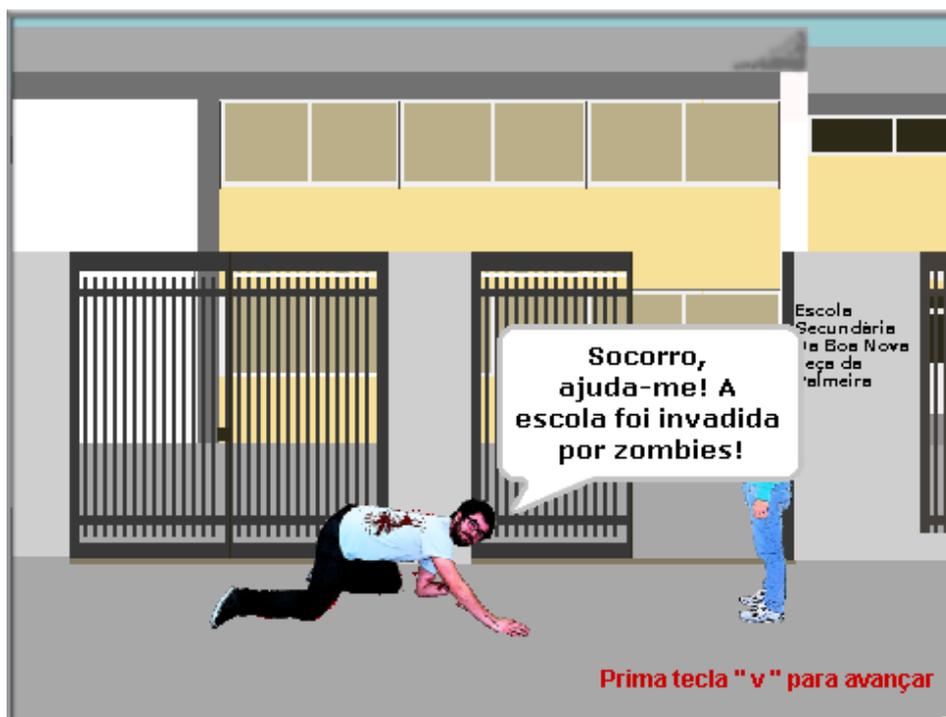


Figura 44 – Cena da entrada da ESBN

Os alunos foram capazes de implementar todas as situações do guião elaborado pela turma do 12º E, no entanto foram feitas algumas alterações no que toca ao movimento do personagem principal, pois como já foi anteriormente referido, não era possível colocar o PP a saltar uma vez que o jogo não se tratava de um jogo de plataformas mas sim de aventura em

que o PP se deslocava nos dois eixos, X e Y. Durante a fase intervenção foi muito bom verificar que este foi o grupo que mais recorreu ao protótipo, não só porque encontraram grandes semelhanças no que toca ao movimento do PP, mas também na forma como iriam resolver a situação de avançar a cena da entrada da ESNB, em que era feita toda uma narrativa para enquadramento no jogo, que deveria ser avançado, caso o jogador assim o pretendesse. As figuras 45 e 46 apresentam as semelhanças com o protótipo encontradas no movimento do PP, assim como a forma como se avança a cena da entrada da ESNB.

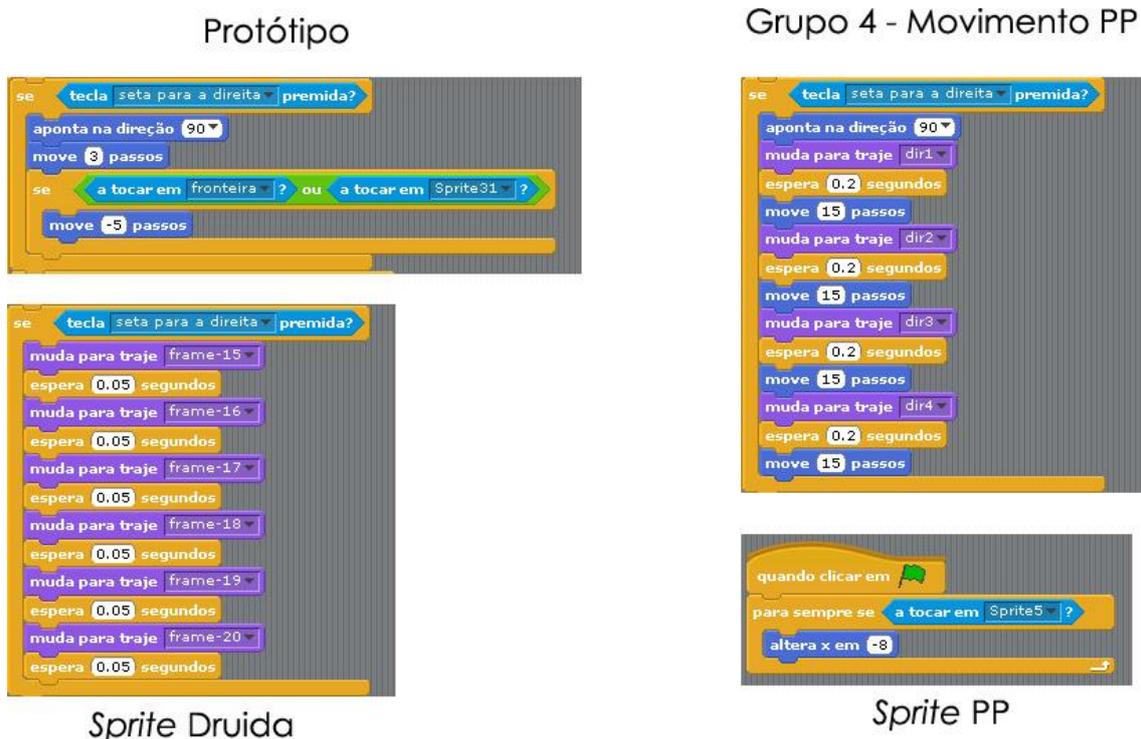


Figura 45 – *Scripts* do movimento do personagem principal (PP)

A Figura 45 apresenta o movimento do personagem principal, tanto no protótipo como na implementação desenvolvida pelo grupo de trabalho 4. São apresentados os *scripts* com o movimento para a direita, que são exatamente iguais aos restantes movimentos, esquerda, acima ou baixo, apenas alterando a direção. Como se pode verificar existem grandes semelhanças entre os *scripts* implementados com os alunos com os do protótipo, no entanto os alunos, foram capazes de dar o seu cunho pessoal e implementar um algoritmo diferente que segundo eles seria mais eficaz. Enquanto no protótipo temos dois scripts a funcionar em paralelo, uma para o movimento e outro para a mudança dos trajes, para dar a sensação dos passos, em que a condição de paragem (sempre que se toca numa fronteira) está inserida no

script de movimento, na implementação dos alunos temos apenas um ciclo que ao mesmo tempo faz o movimento e a mudança dos trajes, sendo a condição de paragem, sempre que se toca nas fronteiras um *script* paralelo.

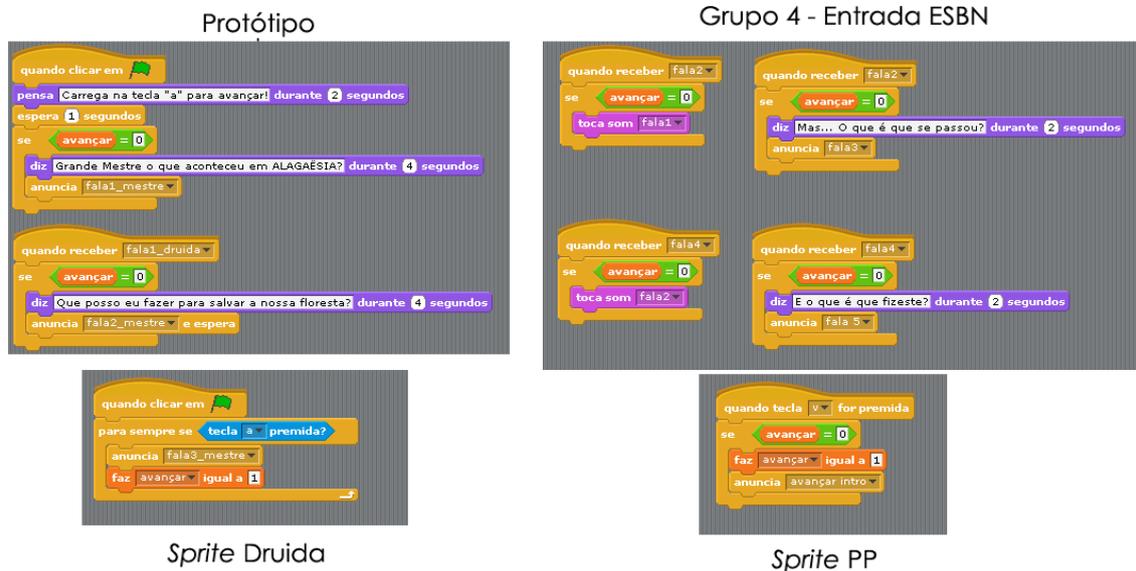


Figura 46 – *Scripts* do momento em que se avançam as falas da entrada da ESNB

Mais uma vez é notória a influência do protótipo no desenvolvimento dos scripts por parte dos alunos, sendo no entanto possível verificar que os alunos conseguiram desenvolver o seu próprio código. No caso do protótipo a cena inicial das falas entre o druida e o mestre são avançadas sempre que se carrega na tecla “a”, para isso existe um ciclo para sempre que está constantemente a verificar se a tecla é pressionada. Os alunos conseguiram implementar uma solução ainda mais inteligente, pois não há necessidade de estarmos a gastar recursos do computador a verificar constantemente se uma tecla é pressionada. Na implementação dos alunos estes resolveram o problema através de um comando que verifica se a tecla “v” foi pressionada. No caso de ainda não ter sido pressionada, a variável “avançar” toma o valor 0 (falso), ao ser pressionada a tecla “v” a variável “avançar” toma o valor 1 (verdadeiro) e anunciam aos restantes *sprites* envolvidos na cena que podem avançar as falas.

À semelhança dos restantes grupos, mais uma vez é notória a utilidade do protótipo no desenvolvimento dos trabalhos, não só como um meio para orientar o trabalho e integração dos diferentes mini projetos no projeto global final, mas também no desenvolvimento do próprio código, sendo notória as semelhanças encontradas no código desenvolvido com o código do

protótipo, com a particularidade de implementarem a meu ver código mais limpo e eficaz, como se pode verificar com a situação apresentada na Figura 46.

3.2.2.5 Grupo 5

Este grupo de trabalho, constituído por cinco alunos, quatro rapazes e uma rapariga, tinha na sua formação o aluno com pior desempenho na área da programação da turma. Não sei se este facto poderá ter constituído uma situação de destabilização dentro do grupo mas o que é certo, é que este grupo de trabalho foi aquele que mais dificuldades teve em implementar os requisitos do guião do jogo para a cena que lhes tinha sido atribuída, a passagem para a biblioteca. Nesta cena o personagem principal deveria percorrer vários palcos relativos a fachada principal da ESNB, e há medida que o ia fazendo tinha que matar 3 zombies, assim que o fizesse tinha caminho livre para poder entrar na biblioteca. Apesar de os alunos encontrarem no protótipo uma situação semelhante, em que eram feitas transições de palco para palco (à semelhança do que fez o grupo 1 na cena da biblioteca), no final este grupo de trabalho não foi capaz de implementar esta situação e tiveram que restringir o seu miniprojecto a um só palco. A Figura 47 apresenta a interface gráfica da cena passagem para a biblioteca.

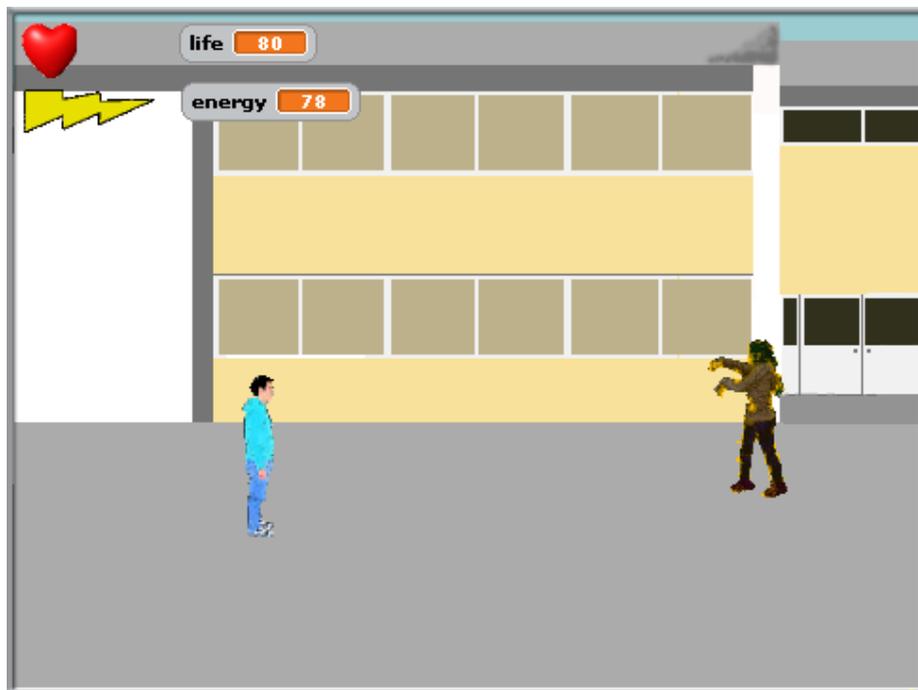


Figura 47 – Cena da passagem para a biblioteca

Para além de não terem sido capazes de implementar todos os requisitos do guião do jogo e terem simplificado a cena, também tiveram grandes dificuldades com a situação em que era retirada energia ao PP sempre que era atacado por um zombie. Para conseguirem ultrapassar esse problema solicitaram a ajuda dos colegas do grupo 1 que tinham implementado uma situação semelhante. Com a colaboração dos colegas foram capazes de adaptar o código desenvolvido pelo grupo 1 para o seu projeto. A figura 48 apresenta o *script* implementado para a situação atrás referida.

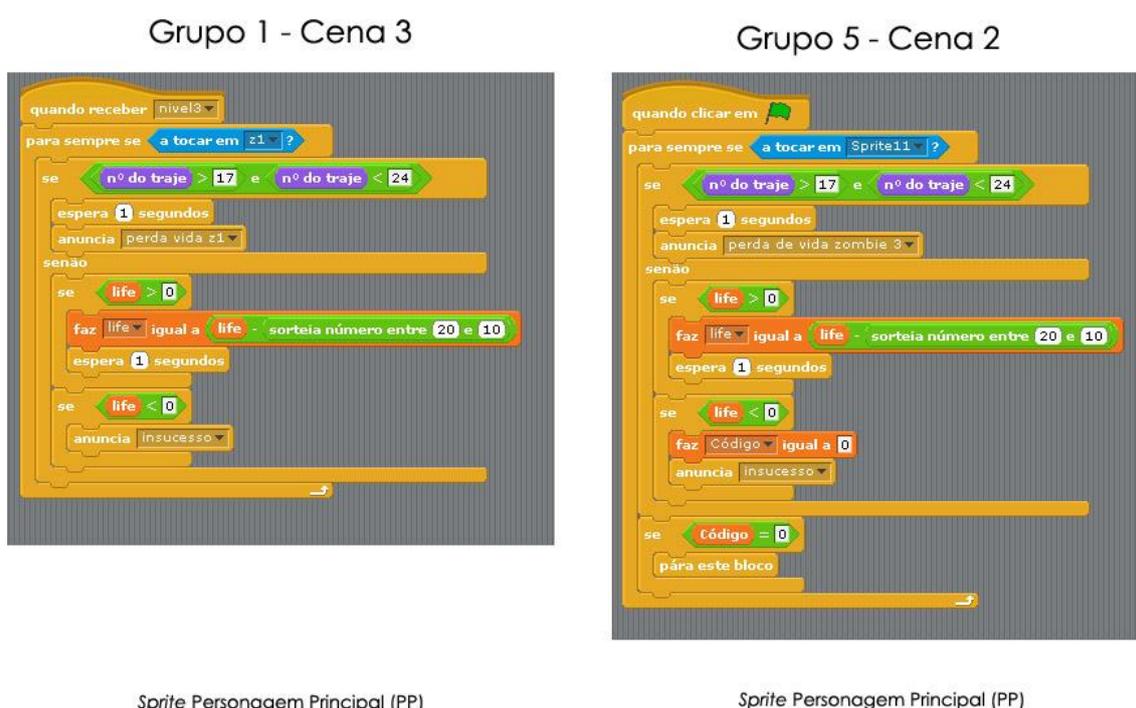


Figura 48 – *Scripts* do momento em que é retirada energia ao PP na cena 2

Como se pode verificar através da análise da Figura 48, os códigos são praticamente iguais. No entanto o grupo 5 teve que adaptar o *script* para que este funcionasse no seu projeto. Para isso tivera que criar uma variável “código” que funcionava como semáforo de forma a indicar no caso de o PP perder toda a sua vida que o jogo tinha terminado e todos os *scripts* deveriam parar de ser executados.

Dadas as dificuldades deste grupo de trabalho, e apesar de terem recorrido bastante ao protótipo, o código que desenvolveram não o demonstra, no entanto através dos meus registos de reflexão feitos ao longo da minha intervenção posso referir que “*Os alunos do grupo 5 (cena 2), que estavam com dificuldade em fazer o movimento dos zombies, a implementar as*

fronteiras nos cenários e efetuar as transições entre palcos, mostraram-se bastante motivados e referiram ter encontrado no protótipo o algoritmo que os ajudaria a ultrapassar as suas dificuldades” (dia 07/05/2013 pelas 11:00) concluindo desta forma que até mesmo para este grupo de trabalho a introdução do protótipo nas minhas aulas foi uma mais-valia para o sucesso e conclusão do *Zschool*.

3.2.3 Entrevista colectiva do tipo *Focus Group* final

A fim de aferir as percepções e opiniões dos alunos sobre o processo de intervenção realizei, juntamente com as minhas colegas estagiárias que acompanharam a turma do 12º E, uma entrevista do tipo *Focus Group* que se realizou no dia 04 junho de 2013, na Escola Secundária da Boa Nova, em dois momentos distintos: o primeiro com alunos da turma B – Ciências e Tecnologias - e o outro com alunos da turma E – Artes Visuais. Optou-se pela separação dos alunos por turma para que a presença de uns não influenciasse ou inibisse a resposta dos outros. Cada uma das entrevistas coletivas teve a duração de aproximadamente 45 minutos e para a sua realização foi necessário um gravador de áudio e um bloco de notas.

Na turma B, chamaram-se para estar presentes os mesmos 12 voluntários do *Focus Group* inicial, tendo-se realizado numa sala independente, na presença de um moderador e de um moderador/assistente.

Na turma E participaram todos os alunos presentes nesse dia (11), tendo também lugar numa sala independente. Para além de um moderador e de um moderador/assistente, desta vez, ao contrário do primeiro *Focus Group*, por opção do próprio, o professor orientador cooperante não esteve presente.

Em ambas as situações foram tidas em conta questões éticas relacionadas com o pedido de autorização para a gravação áudio das entrevistas (autorização previamente concedida). Antes do início da gravação os participantes foram ainda informados dos objetivos gerais da atividade, esclarecendo-se que o objetivo principal se prendia com a recolha generalizada de opiniões enquanto grupo/turma e não com opiniões particulares dos alunos. Salientou-se ainda o facto de o ênfase estar na avaliação por parte dos alunos dos nossos projectos de intervenção no âmbito do estágio curricular.

Após transcrever a entrevista da turma B, objecto do meu estudo, fiz uma análise do conteúdo. Bardin (1979, p. 42) define análise de conteúdo como “*um conjunto de técnicas de*

análise das comunicações visando obter, por procedimentos, sistemáticos e objetivos de descrição do conteúdo das mensagens, indicadores (quantitativos ou não) que permitam a inferência de conhecimentos relativos às condições de produção/recepção (variáveis inferidas) destas mensagens". A finalidade da análise de conteúdo é produzir inferência, trabalhando com vestígios e evidências. Bardin (1979) caracteriza a análise de conteúdo como sendo empírica e, por esse motivo, não pode ser desenvolvida com base num modelo exato. Contudo, para sua operacionalização, devem ser seguidas algumas regras de base, por meio das quais se parte de uma literatura de primeiro plano para atingir um nível mais aprofundado. Assim, o processo de análise de conteúdo consistiu, inicialmente, numa leitura flutuante do texto transcrito e na codificação do material existente: transformação dos dados brutos, por recorte, classificação, agregação e categorização (Bardin, 1979).

Apresento (Tabela 56) as categorias criadas a partir da entrevista coletiva aos alunos e exemplos das evidências que as suportam.

Categoria	Evidência (exemplo)	Frequência
gostaram	<p><i>"Gostei de aprender a utilizar o Scratch, é sempre mais um tipo de programação a conhecer e nunca tinha programado antes jogos."</i> (Aluno B)</p> <p><i>"Adorei criar este jogo, adquiri mais conhecimentos e deu para me divertir um pouco. Comparando com o primeiro período, acho que este foi mais educativo, talvez porque a matéria ou o programa é mais interessante."</i> (Aluna C).</p> <p><i>"Gostei bastante de trabalhar com o Scratch, admito que este jogo foi muito trabalhoso, custou "muitas horas e dores de cabeça". Mas fui motivado pela ideia do fantástico jogo que poderia vir a ser."</i> (Aluno K).</p>	4
divertido	<p><i>"achei a criação deste jogo uma experiência bastante enriquecedora a nível cognitivo e um trabalho divertido"</i> (Aluno L)</p>	2
novo conhecimento	<p><i>"Fazer este jogo obrigou-me a adquirir bastantes conhecimentos sobre como trabalhar com o programa Scratch. Foi muito benéfico para mim e para a minha aprendizagem."</i> (Aluno D)</p> <p><i>"Aprendi bastantes funcionalidades do Scratch que me eram antes desconhecidas"</i> (Aluno F)</p> <p><i>"Este trabalho permitiu-me desenvolver</i></p>	3

	<i>ainda mais algumas bases de programação, puxar pela minha criatividade e melhorar competências na área dos videojogos” (Aluno G)</i>	
dificuldades	<i>“gostei de trabalhar neste projeto, apesar das dificuldades acho que o resultado final foi positivo” (Aluna J) “o mais complicado foi entender a lógica da programação inerente ao Scratch, pois os meus conhecimentos de programação não eram amplos o suficiente” (Aluna M).</i>	2
autonomia	<i>“o mais interessante foi ultrapassar as minhas dificuldades e ter feito tudo pela minha cabeça e pelo meu esforço” (Aluno H)</i>	1

Tabela 56 – Categorias criadas a partir da entrevista do tipo *Focus Group* final

No que toca à planificação do projecto de intervenção, que foi remodelada ao longo da intervenção devido ao facto de um meu projecto estar dependente de conteúdos desenvolvidos pela turma E, os alunos acharam que foi bastante prolongado o tempo em que tiveram que trabalhar com o *Scratch*. Na opinião da maioria dos alunos, o ideal seria termos parado com o *Scratch* no final do segundo período e ter tido tempo para desenvolver o *Zschool* dentro desse período, deixando o terceiro período dedicado exclusivamente para o módulo da multimédia. Em seguida apresento mais algumas conclusões a partir da transcrição da sessão de *Focus Group*. É notório que os alunos gostaram bastante de ter trabalhado com o *Scratch* e de terem tido a oportunidade de desenvolver um jogo, algo que à partida parecia inalcançável:

“Gostei de aprender a utilizar o Scratch, é sempre mais um tipo de programação a conhecer e nunca tinha programado antes jogos.” (Aluno B).

“Adorei criar este jogo, adquirir mais conhecimentos e deu para me divertir um pouco. Comparando com o primeiro período, acho que este foi mais educativo, talvez porque a matéria ou o programa é mais interessante.” (Aluna C).

Os alunos também disseram que aprenderam bastante, que ganharam novos conhecimento de programação, como um aluno que referiu que *“fazer este jogo obrigou-me a adquirir bastantes conhecimentos sobre como trabalhar com o programa Scratch. Foi muito benéfico para mim e para a minha aprendizagem”* (Aluno D), ou outro que disse que *“este trabalho permitiu-me desenvolver ainda mais algumas bases de programação, puxar pela minha criatividade e melhorar competências na área dos videojogos, que é uma área que me agrada e com um amplo mercado mundial.”* (Aluno G), ou ainda *“aprendi bastantes funcionalidades do Scratch que me eram antes desconhecidas.”* (Aluno F), o que os fez ficarem satisfeitos com os

seus progressos: “*Quando terminei o trabalho fiquei orgulhosa pelo que tinha feito, e por este funcionar.*” (Aluna E).

Apesar das dificuldades sentidas, como o relatado por uma aluna quando disse “*gostei de trabalhar neste projeto, apesar das dificuldades acho que o resultado final foi positivo*” (Aluna J), os alunos sentiram-se enriquecidos, como foi referido por um aluno quando disse “*achei a criação deste jogo uma experiência bastante enriquecedora a nível cognitivo e um trabalho divertido*” (Aluno L).

O facto de ter introduzido o *Scratch* e adotar uma metodologia diferente através do desenvolvimento de jogos permitiu também aumentar a autonomia dos alunos, o que pode ser comprovado pela seguinte afirmação “*o mais interessante foi ultrapassar as minhas dificuldades e ter feito tudo pela minha cabeça e pelo meu esforço*” (Aluno H) e aumentar a motivação “*gostei bastante de trabalhar com o Scratch, admito que este jogo foi muito trabalhoso, custou muitas horas e dores de cabeça. Mas fui motivado pela ideia do fantástico jogo que poderia vir a ser.*” (Aluno K).

No final foi bom ouvir no discurso dos alunos que estes conseguiram desenvolver o raciocínio lógico/abstracto, condição fundamental para aprender a programar segundo os autores França *et al.* (2010), Wiendenbeck, 2004; Gomes *et al.*, 2012; e, Ambrósio *et al.*, 2011, de que é exemplo o comentário de uma aluna: “*o mais complicado foi entender a lógica da programação inerente ao Scratch, pois os meus conhecimentos de programação não eram amplos o suficiente*” (Aluna M).

Mesmo assim, houve alunos que referiram preferir os métodos tradicionais de programação, que ouvimos através da afirmação “*ainda assim continuo a preferir programar a escrever as linhas de código e não a arrastar "objetos"!!!*” (Aluno I). Pensamos que isso é muito bom, pois no futuro, para aqueles que ingressarem em áreas profissionais na área da programação informática, serão essas linguagens de programação com que vão ter de trabalhar.

4 CONCLUSÕES, LIMITAÇÕES E RECOMENDAÇÕES

4.1 Conclusões

Retomando o **primeiro objectivo** formulado (motivar os alunos para as aulas de programação informática através da linguagem de programação *Scratch*) considero que o mesmo foi cumprido. Se atendermos à entrevista (*Focus Group*) final, através de algumas das afirmações feitas pelos alunos, nomeadamente “*gostei de aprender a utilizar o Scratch, é sempre mais um tipo de programação a conhecer e nunca tinha programado antes jogos*” (Aluno B); “*gostei de trabalhar neste projeto, apesar das dificuldades acho que o resultado final foi positivo*” (Aluna J); “*achei a criação deste jogo uma experiência bastante enriquecedora a nível cognitivo e um trabalho divertido*” (Aluno L); “*gostei bastante de trabalhar com o Scratch, admito que este jogo foi muito trabalhoso, custou "muitas horas e dores de cabeça". Mas fui motivado pela ideia do fantástico jogo que poderia vir a ser*” (Aluno K), este objectivo foi alcançado. Perante estas afirmações, é lícito afirmar que os alunos se motivaram para a programação com o *Scratch* e que se tornaram conscientes sobre a mais valia do *Scratch* para a conclusão dos seus jogos individuais e de grupo. Durante a fase de observação tinha sido notória a existência de um pequeno grupo de alunos (4 raparigas e 3 rapazes) que não estavam motivados para a programação com o Pascal e não eram capazes de programar de forma autónoma. Durante as aulas mostravam-se desmotivados e socorriam-se do colega de trabalho (todos eles trabalhavam em grupos de dois) para que estes realizassem as fichas de trabalho. Ao introduzir o *Scratch* no segundo período foi notório o aumento da motivação dos alunos para a programação. Facto que se veio a confirmar com os jogos individuais desenvolvidos, para além de que uma das alunas passou a trazer o seu portátil para as aulas para poder programar sozinha. Através dos trabalhos de grupo, foi notória a colaboração entre os alunos, não só dentro do grupo de trabalho, como também entre os próprios grupos. Durante as aulas de trabalho de grupo não eram raras as vezes em que elementos de um grupo de trabalho se levantavam para integrar outro grupo de trabalho e auxiliar na integração das diferentes cenas desenvolvidas. Penso que a estratégia que adoptei durante as minhas aulas, não só durante a fase de exploração do *Scratch*, em que por muita vezes os alunos de forma autónoma iam consultar fóruns sobre o *Scratch* assim como tutoriais no *Youtube*, onde procuravam esclarecer as suas dúvidas e ultrapassar as suas dificuldades assim como a metodologia de desenvolvimento

adoptada, em que os alunos recorriam ao protótipo por mim desenvolvido sempre que tinham dúvidas, adaptando linhas de código, contribuíram para aumentar a autonomia dos alunos.

Relativamente ao **segundo objectivo** (estimular os alunos a desenvolverem os seus próprios jogos a partir da linguagem de programação *Scratch*), o mesmo foi atingido porque cada aluno desenvolveu realmente um jogo em *Scratch*. A avaliação que fiz dos trabalhos mostra o domínio da ferramenta e a diversidade de jogos que conseguiram desenvolver. Mas não só através dos exemplos e exercícios apresentados na primeira fase da minha intervenção. Também através da metodologia de desenvolvimento utilizada, apoiada num protótipo de um jogo RPG, que permitiram estimular os alunos a criar os seus próprios jogos individuais de elevada qualidade assim como um jogo global para toda a turma intitulado *Zschool*. Indo ao encontro das ideias defendidas por vários os autores (Murray, 1999; Turkle, 1997; Overmars, 2004; Paraskeva *et al.*, 2010), que defendem o impacto positivo dos jogos digitais na educação, nomeadamente na forma como os alunos desenvolvem as capacidades de improvisação, raciocínio lógico, resolução de problemas e construção de novas realidades através do desenvolvimento de jogos digitais.

No que se refere ao **terceiro objectivo** (ensinar a programar através da linguagem de programação *Scratch*, desenvolvendo o raciocínio lógico/abstracto que a programação exige) considero que o mesmo também foi atingido, Na realização dos seus projectos individuais, com exceção de um aluno, todos os restantes alunos aplicaram todos ou pelo menos seis dos sete conceitos de programação apontados por Maloney *et al.* (2008), o que mostra que através do *Scratch* os alunos foram capazes de programar, os alunos encontraram soluções para os problemas com que se depararam, através da exploração criativa do ambiente de trabalho. A programação exige pensar de forma lógica (em vez de memorizar) e, conseqüentemente, “*desenvolver o raciocínio lógico e abstracto e também raciocinar mais criativamente*” (Resnick, 2009, p. 60). Através da análise dos trabalhos individuais pude constatar que em todos eles, com excepção do primeiro trabalho, os alunos foram capazes de implementar um sistema de pontuação ou vidas através do uso de variáveis, que são dos conceitos mais abstractos da programação (Resnick, 2009).

Como já foi anteriormente referido por vários autores (França *et al.*, 2010, Wiendenbeck, 2004; Gomes *et al.*, 2012; Ambrósio *et al.*, 2011), o problema principal com a dificuldade na aprendizagem da programação reside principalmente na incapacidade dos alunos resolverem problemas, apresentarem soluções (algoritmos), ou seja, a dificuldade reside na

concepção e formalização de uma solução para um determinado problema através do raciocínio lógico/abstracto e não propriamente na sua codificação. Através do meu projecto de intervenção e recorrendo-me da ferramenta *Scratch* foi possível ajudar os alunos não apenas a compreender o desenvolvimento dos seus programas, mas sobretudo lhes permitir realizar, testar, experimentar, alterar e corrigir os seus próprios programas permitindo compreender a lógica das soluções (algoritmos) desenvolvidos. O sucesso do *Scratch* reside não apenas nas suas funcionalidades e potencialidades, mas essencialmente na forma como interage e comunica com o utilizador. Com este projecto de intervenção apresentei não uma abordagem expositiva dos assuntos a ensinar/aprender, em que o aluno recebe a informação passivamente, mas antes consegui explorar um ambiente que permite ao alunos desenvolver os seus programas com base na experimentação e na prática. O *Scratch* permitiu aos alunos a construção, observação, análise e simulação visual da forma como um dado programa (solução/algoritmo) funciona ou se comporta mediante certas circunstâncias, possibilitando-lhes detectar eventuais erros, corrigi-los e através disso aprender ao seu próprio ritmo.

Relativamente ao **quarto objectivo** (promover a interdisciplinaridade entre a turma de Tecnologias e a turma de Artes), constatou-se no terceiro período que tal foi possível pois no final cada grupo de trabalho conseguiu desenvolver o seu miniprojecto e entre turma conseguiram integrar todos os miniprojectos num projeto final global. Como se pode verificar no ponto 3.2.2 deste relatório, onde são apresentados os projetos de grupo, os alunos foram capazes de superar as suas dificuldades e concluir com sucesso os seus projetos, que através da integração deram origem ao jogo *Zschool*. Nesta fase foi crucial a introdução do protótipo, à luz da metodologia de desenvolvimento, que permitiu aumentar a autonomia dos alunos no desenvolvimento dos seus projectos de grupo, tal pode ser constatado através dos meus registos de reflexão feitos ao longo da minha intervenção: *“Também era frequente ver os alunos a detectar um erro de programação e recorrer ao protótipo para encontrar a solução. Foram raras as vezes em que foi solicitada a minha ajuda”* (nota do dia 28/05/2013 pelas 11:20).

Considero, assim, que cumpro os objetivos que me propus através da adopção de estratégias pedagógicas que foram as que melhor se adequaram, tendo contribuído para o sucesso deste meu projeto de intervenção.

4.2 Limitações

Em relação às **limitações ao estudo** realizado, considero que o facto de se tratar de uma turma com 26 alunos e apenas 13 computadores constituiu uma das grandes limitações. Apesar de, como já tinha referido, alguns alunos trazerem os seus portáteis de casa, mesmo assim havia alunos que trabalhavam aos pares, sendo difícil muitas vezes perceber se ambos estavam a compreender o que lhes era pedido e se ambos estavam a conseguir desenvolver as capacidades necessárias para a programação. Uma fórmula que usei para contornar esta limitação foi através dos projetos individuais, desenvolvidos individualmente pelos alunos, nas aulas e em casa por aqueles alunos mais interessados e motivados.

O facto das oito primeiras aulas terem sido alternadas com o professor orientador cooperante, devido ao facto de termos aulas à terça-feira na Universidade do Minho, também constitui uma das limitações ao estudo, pois era muito difícil manter a coerência de aula para aula e o facto de haver essa interrupção cortava o ritmo das minhas aulas, que eram sempre bastante dinâmicas. O professor titular conseguia acompanhar os alunos, esclarecendo sempre as dúvidas. O professor titular é um especialista na área da programação e mostrou desde o início bastante interesse no *Scratch*. Este facto foi uma mais-valia para o sucesso deste trabalho, caso contrário toda a intervenção poderia ficar em causa, ajudando também a minimizar o impacto desta limitação.

Outra limitação ao trabalho realizado foi o facto de parte do projeto interdisciplinar (desenvolvimento de um jogo global para toda a turma) estar dependente de dois outros projetos de intervenção, desenvolvidos por duas colegas estagiárias que acompanhavam a 12º E de Artes. Como os alunos referiram na entrevista coletiva (*Focus Group*), o *Scratch* deveria ter-se limitado ao segundo período e o trabalho de grupo deveria ter sido realizado no segundo período. Tal não aconteceu porque estávamos dependentes do material multimédia desenvolvido pela turma de Artes que apenas começou a ficar definido no início do terceiro período. Na minha opinião foi muito arriscado realizar um projeto que de certa forma não dependia exclusivamente do meu trabalho e dos meus alunos mas também do trabalho realizado por outros professores e por alunos de outra turma. No entanto, penso que no final acabou tudo por correr bem e considero que foi importante para os alunos da turma do 12º B trabalharem com outros colegas, muito diferentes deles nos interesses, e com ritmos e formas de trabalhar diferentes. No futuro alguns destes alunos terão as suas carreiras profissionais no mundo do desenvolvimento de

software e serão situações semelhantes às do projeto *Zschool* que desenvolveram que se lhes irão deparar pela frente. Com este projecto conseguiu-se demonstrar a importância da interdisciplinaridade ao conjugar competências de duas turmas diferentes, que trabalhando de forma cooperativa, conseguiram completar com sucesso um projecto bastante interessante.

4.3 Recomendações

Recomendações para o futuro prendem-se essencialmente na planificação dos períodos de intervenção para que sejam feitas sem que hajam interrupções permitindo um pleno acompanhamento da turma.

Por fim, gostaria de concluir que esta intervenção permitiu-me desenvolver capacidades pedagógicas e de investigação que espero poder vir a usar na minha vida profissional.

5 REFERÊNCIAS

- Alves, L. (2008). *Relações entre os jogos digitais e aprendizagem: delineando percurso*. Brasil: Educação, Formação & Tecnologias, vol. 1 (2).
- Ambrósio, A., Almeida, L., Macedo, J., Santos, A., & Franco, A. (2011). *Programação de Computadores: Compreender as Dificuldades de Aprendizagem dos Alunos*. Revista Galego-Portuguesa de Psicologia e Educación, Vol. 19, No. 1, pp: 185-197.
- Bardin, L. (1979). *Análise de conteúdo*. Lisboa: Edições 70.
- Bethke, E. (2003). *Game development and production*. Wordware Publishing Inc.
- Bittencourt, J. R., & Giraffa, L. M. (2003). Modelando Ambientes de Aprendizagem Virtuais utilizando Role-Playing Games. *XIV Simpósio Brasileiro de Informática na Educação* (pp. 683 - 692). Porto Alegre: IM/UFRJ.
- Borges, M. A. (2000). *Avaliação de uma metodologia alternativa para a aprendizagem de programação*. Brasil: Faculdade Campo Limpo Paulista.
- Brasil, M. d. (30 de 11 de 2009). *Criação de um Texto Cooperativo – RPG (Role Playing Game)*. Obtido em 18 de 02 de 2013, de Portal do Professor: <http://portaldoprofessor.mec.gov.br/fichaTecnicaAula.html?aula=12381>
- Cooper, S. (2010). *The design of Alice*. EUA: ACM Transactions on Computing Education. 10(4), Artigo 15 (16 páginas).
- Correia, I. (2012). Scratch(ando) de braço dado com a Matemática: imaginar, programar, partilhar. *Cadernos de Educação de Infância*, 19-22.
- Council, N. R. (2012). *Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century*. Washington, DC: The National Academies Press.
- Coutinho, C. P., & Chaves, J. H. (2001). *Investigação em Tecnologia Educativa na Universidade do Minho: uma abordagem temática e metodológica às dissertações de mestrado concluídas nos cursos de mestrado em educação*. In A. Estrela & J. Ferreira (org.), *Tecnologias em Educação: Estudos e Investigações: X Colóquio AFIRSE/AIPELF*, pp. 289-302.
- Dagdilelis, V., Evangelidis, G., Satratzemi, M., Efpoulos, V., & Zagouras, C. (2003). *DELYS: A novel microworld-based educational software for teaching computer science subjects*. Grécia: Computers & Education, 40(4), 307-325.
- Depradine, C. A. (2011). *Using gaming to improve advanced programming skills*. Barbados: Caribbean Teaching Scholar, 93–113.

- Dumas, J. S., & Redish, J. C. (1999). *A Practical Guide to Usability Testing. Exeter*. Intellect Books.
- Educação, D. G. (s.d.). *EduScratch / DGE / ERTE / CCTIC / ESE/IPS*. Obtido em 21 de 11 de 2012, de <http://eduscratch.dgdc.min-edu.pt/>
- Federoff, M. A. (2002). *Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games*. (B. Indiana University, Ed.) Obtido em Janeiro de 2013, de <http://citeserx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.8294>
- Fox, D., & Verhosek, R. (2002). *Micro Java TM Game Development*. Addison-Wesley.
- França, E., Felix, Z., Souza, M., Carneiro, T., Sousa, P. R., & Filho, C. A. (2010). *Utilização de Objetos de Aprendizagem em Sistemas Tutores Inteligentes para o ensino da Programação*. Brasil: VII SEGeT – Simpósio de Excelência em Gestão e Tecnologia.
- Gomes, A., Henriques, J., & Mendes, A. J. (2008). *Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores*. In Educação, Formação & Tecnologias; vol.1(1), pp. 93-103.
- Gomes, A., Mendes, A., & Marcelino, M. (2004). *Avaliação e evolução de um ambiente de suporte à aprendizagem da programação*. Instituto Superior de Engenharia de Coimbra: VII Congresso Iberoamericano de Informática Educativa.
- Gomes, G., Martinho, J., Bernardo, M., Matos, F., & Abrantes, P. (2012). *Dificuldades de aprendizagem de programação no ensino profissional – perspectiva dos alunos*. Lisboa: Actas do II Congresso Internacional TIC e Educação, pp. 438 – 448.
- Gomes, T. S. (2009). *O Potencial Educativo dos Massively Multiplayer Online Games: um estudo exploratório sobre os jogos Ikariam, OGame, Gladius e Metin2 e seus jogadores*. Braga: Instituto de Educação e Psicologia - Universidade do Minho.
- Gurgel, I., Almeida, E., Arcoverde, R., Sultanum, N., & Tedesco, P. (2006). *A Importância de Avaliar a Usabilidade dos Jogos: A Experiência do Virtual Team*. Obtido em 15 de Fevereiro de 2013, de SBGames 2006: <http://www.cin.ufpe.br/~sbgames/proceedings/aprovados/23657.pdf>
- Kahn, K. (2001). *Generalizing by removing detail: How any program can be created by working with examples*. EUA: Morgan Kaufmann.
- Kelleher, C., & Pausch, R. (2005). *Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers*. EUA: ACM Computing Surveys, 37(2), 83-137.
- Kieras, D. (2006). *User Interface Design for Games [online]*. Obtido de University of Michigan: www.eecs.umich.edu/~soar/Classes/494/talks/User-interfaces.pdf

- Kolling, M. (2010). *The Greenfoot programming environment*. EUA: ACM Transactions on Computing Education, 10(4), Artigo 14 (21 páginas).
- Laitinen, S. (23 de Junho de 2005). *Better Games Through Usability Evaluation and Testing*. Obtido em Fevereiro de 2013, de Gamesutra: http://www.gamasutra.com/view/feature/2333/better_games_through_usability_.php
- Lencastre, J. A. (2009). *Educação On-line: Um estudo sobre o blended learning na formação pós-graduada a partir da*. Braga: Universidade do Minho. Instituto de Educação e Psicologia. (Texto Policopiado).
- Lencastre, J. A. (2012). *Metodologia para o desenvolvimento de ambientes virtuais de aprendizagem: development research*. In Angélica Monteiro, J. António Moreira & Ana Cristina Almeida (ed.), *Educação Online: Pedagogia e aprendizagem em plataformas digitais*, (pp. 45 – 54). Santo Tirso: DeFacto Editores. ISBN: 978-989-8557-02-5.
- Lencastre, J. A., & Chaves, J. H. (2007). Avaliação Heurística de um Sítio Web Educativo: o Caso do Protótipo "Atelier da Imagem". In P. Dias, C. Freitas, B. Silva, & A. & Osório, *Actas da V Conferência Internacional de Tecnologias de Informação e Comunicação na Educação - Challenges 2007*. Braga: Universidade do Minho.
- Leutenegger, S., & Edgington, J. (2007). *A games first approach to teaching introductory programming*. EUA: 38th SIGCSE technical symposium on Computer science education.
- Malone, T. W. (1982). Heuristics for designing enjoyable user interfaces: Lessons from computer games. *CHI'82: Proceedings of the 1982 conference on Human factors in computing systems* (pp. 63-68). New York, USA: ACM.
- Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by Choice: Urban Youth Learning Programming with Scratch*. EUA: 39th SIGCSE technical symposium on Computer science education Páginas 367-371.
- Menezes, C. (2003). *Informática Educativa II – Linguagens para a Representação do Conhecimento*. Vitória: UFES.
- Murray, J. H. (1999). *Scribd*. Obtido em 7 de Agosto de 2013, de <http://pt.scribd.com/doc/36062359/Hamlet-en-La-Holocubierta>
- Nielsen, J. (1993). *Usability Engineering*. San Francisco: Morgan Kaufmann.
- Nielsen, J., & Molich, R. (Abril de 1990). Heuristic Evaluation Of User Interfaces. *CHI 90 Proceedings*, pp. 249-256.
- Nokia. (2006). *Mobile Game Playability Heuristics*. Obtido em Janeiro de 2013, de Nokia Forum: http://www.forum.nokia.com/info/sw.nokia.com/id/5ed5c7a3-73f3-48ab-8e1e-631286fd26bf/Mobile_Game_Playability_Heuristics_v1_0_en.pdf.html

- Nova, E. S. (2011). *Projeto Educativo*. Leça da Palmeira.
- Overmars, M. (2004). *Game Design in Education*. Holanda: Institute of Information and Computing Sciences, Utrecht University.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerfull Ideas*. EUA: Basic Books.
- Paraskeva, F., Mysirlaki, S., & Papagianni, A. (2010). *Multiplayer online games as educational tools: Facing new chalenges in learning*. Computers & Education, 54, 498-505.
- Pecchinenda, G. (2003). *O Império do Instante: os novos meios e o tempo de experiência*. Belo Horizonte - Brasil: III Colóquio Brasil-Itália de Ciências da Comunicação.
- Piaget, J. (1979). *O nascimento da inteligência na criança*. Rio de janeiro: Zahar.
- Prensky, M. (2001). *Digital Game - Based Learning*. McGraw-Hill.
- Resnick, M., Maloney, J., Hernández, Rusk, Eastmond, Brennan, et al. (2009). *Scratch: Programming for Everyone*. EUA: ACM communications.
- Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing*. Indiana: Wiley Publishing, Inc.
- Schneiderman, B. (1997). *Direct Manipulation*. In *Designing the User Interface: Strategies for Effective Human-Computer (pp. 182-233)*.
- Silva, R. E., & Martins, S. W. (2007). *Ensino da Ciência da Computação através do desenvolvimento de Jogos*. Mar del Plata: VII Congresso de Informática Educativa.
- Slator, B., Hill, C., & Del Val, D. (2004). *Teaching computer science with virtual worlds*. IEEE Transactions on Education, 47(2), 269-275.
- Turkle, S. (1997). *A vida no ecrã – a identidade na era da Internet*. Lisboa: Relógio de Água.
- van Someren, M., Barnard, Y., & Sandberg, J. (1994). *The Think Aloud Method: A Practical Guide to Modeling Cognitive Processes*. London: Academic Press.
- Vos, N., Van der Meijden, H., & Denessen, E. (2011). *Effects of constructing versus playing an educational game on student motivation and deep learning strategy use*. Computers & Education, 56, 127–137.
- Vygotsky, L. (1978). *Mind in society: The Development of Higher Psychological Processes*. Cambridge: MA: Harvard University Press.
- Wiendenbeck, S., Labelle, D., & Kain, V. (April 2004). *Factors affecting course outcomes in introductory programming*. Carlow Ireland: 16th Workshop of the Psychology of Programming Interest Group.

6 ANEXOS

Anexo 1 – Planos de aula elaborados

Plano de aula 1

	Plano de Aula N°XXX Unidade 1 : Introdução à Programação	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013 Aplicações Informáticas B
---	---	--

Sumário: Apresentação da planificação do segundo período. Introdução ao Scratch.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 03 de janeiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 	Introdução ao Scratch	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula 	1'		
		<ul style="list-style-type: none"> Apresentação das noções básicas do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software) 	1'		
<ul style="list-style-type: none"> Utilizar e manipular as instruções do Scratch 	Uso de algumas Estruturas de controle	<ul style="list-style-type: none"> Proposta aos alunos de desenvolvimento de um jogo PAC-MAN (interacção dos alunos com o software) 	3'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> Aplicar os conhecimentos adquiridos no desenvolvimento de um jogo 			40'	<ul style="list-style-type: none"> Utilização do software de programação "Scratch" 	Observação direta
			45'	<ul style="list-style-type: none"> Apresentação eletrónica (PPT) Quadro e canetas 	Avaliação formativa (envio do jogo resolvido para a plataforma moodle)

Plano de aula 2

 <p style="font-size: 8px;">E.S. DA BOA NOVA LEÇA DA PALMEIRA</p>	<p>Plano de Aula N^oXXX Unidade 1 : Introdução à Programação</p>	<p>Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013</p> <p style="text-align: center;">Aplicações Informáticas B</p>
--	---	--

Sumário: Manipulação de Listas no Scratch. Realização de uma ficha prática.

Turma: 12^o B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 10 de janeiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
Percepcionar os conhecimentos prévios dos alunos		<ul style="list-style-type: none"> - Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio) 	2' 3'		Observação direta
<ul style="list-style-type: none"> • Utilizar e manipular as instruções relativas às listas no Scratch 	Listas no Scratch	<ul style="list-style-type: none"> - Apresentação das noções básicas sobre a manipulação de Listas do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software) 	40'	<ul style="list-style-type: none"> - Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> • Dominar a utilização de listas no Scratch 	Estruturas de controle	<ul style="list-style-type: none"> - Realização de uma ficha de trabalho para consolidação de conhecimentos adquiridos na aula 	40'	<ul style="list-style-type: none"> - Apresentação eletrónica (PPT) - Quadro e canetas 	Avaliação formativa (envio dos exercícios resolvido para a plataforma moodle)
<ul style="list-style-type: none"> • Aplicar os conhecimentos adquiridos na realização da ficha prática 		<ul style="list-style-type: none"> - Síntese da aula e apresentação dos objetivos para a aula seguinte. 	5'		

Plano de aula 3

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula NºXXX Unidade 1 : Introdução à Programação		

Sumário: Manipulação de Listas no Scratch. Realização de uma ficha prática.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 17 de janeiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
Percepcionar os conhecimentos prévios dos alunos • Utilizar e manipular as instruções relativas às listas no Scratch • Dominar a utilização de listas no Scratch • Aplicar os conhecimentos adquiridos na realização da ficha prática	Listas no Scratch	- Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio)	2' 3'	- Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch"	Observação direta
		- Apresentação das noções básicas sobre a manipulação de Listas do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software)	40'		Observação direta
	Estruturas de controle	- Realização de uma ficha de trabalho para consolidação de conhecimentos adquiridos na aula	40'	- Apresentação eletrónica (PPT) - Quadro e canetas	Avaliação formativa (envio dos exercícios resolvido para a plataforma moodle)
		- Síntese da aula e apresentação dos objetivos para a aula seguinte.	5'		

Plano de aula 4 (supervisionada)



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

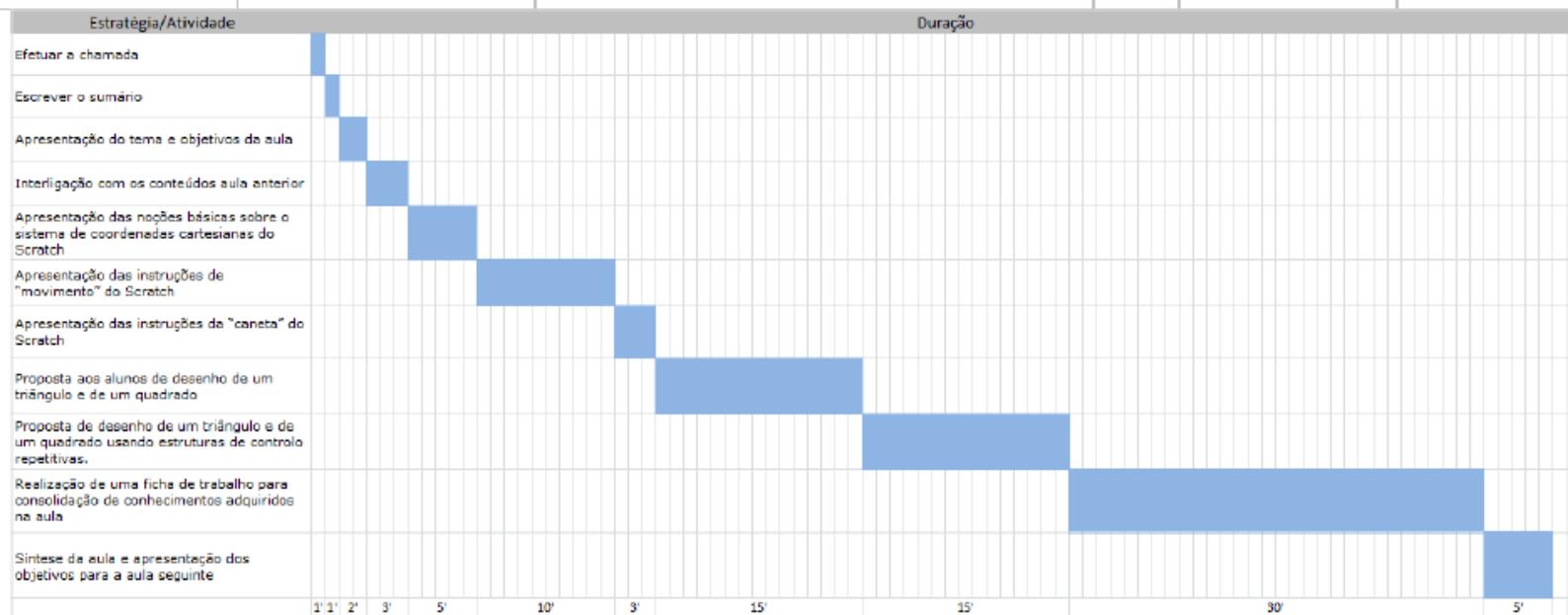
Sumário: Instruções de "movimento" e "caneta" no Scratch. Desenvolvimento de um algoritmo em Scratch para desenho de polígonos.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 24 de janeiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 	Sistema de coordenadas cartesianas	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) 	1' 1' 2' 3'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> Utilizar e manipular as instruções relativas ao "Movimento" e "Caneta" no Scratch 	<ul style="list-style-type: none"> Movimento de sprites no ecrã Desenho de figuras geométricas através do uso de movimento, caneta 	<ul style="list-style-type: none"> Apresentação das noções básicas sobre o sistema de coordenadas cartesianas do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software) Apresentação das instruções de "movimento" do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software) Apresentação das instruções da "caneta" do Scratch, recorrendo ao software (exposição de conteúdos) Proposta aos alunos de desenho de um triângulo e de um quadrado (interacção dos alunos com o software) 	5' 10' 3' 15'	<ul style="list-style-type: none"> Apresentação eletrónica (PPT) Quadro e canetas 	

<ul style="list-style-type: none"> • Dominar a utilização de estruturas de controlo no Scratch 	<ul style="list-style-type: none"> • Uso de algumas Estruturas de controle 	<ul style="list-style-type: none"> - Proposta de desenho de um triângulo e de um quadrado usando estruturas de controlo repetitivas. (interacção dos alunos com o software) 	15'		Observação direta
<ul style="list-style-type: none"> • Aplicar os conhecimentos adquiridos no desenvolvimento de um algoritmo de desenho de polígonos 		<ul style="list-style-type: none"> - Realização de uma ficha de trabalho para consolidação de conhecimentos adquiridos na aula 	30'		Avaliação formativa (envio do algoritmo resolvido para a plataforma moodle)
		<ul style="list-style-type: none"> - Síntese da aula e apresentação dos objetivos para a aula seguinte. 	5'		



Plano de aula 5



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Sumário: Manipulação de lista no Scratch. Realização de uma ficha prática.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 31 de janeiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
Percepcionar os conhecimentos prévios dos alunos		<ul style="list-style-type: none"> - Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio) 	2' 3'		Observação direta
<ul style="list-style-type: none"> • Utilizar e manipular as instruções relativas às listas no Scratch 	Listas no Scratch	<ul style="list-style-type: none"> - Apresentação das noções básicas sobre a manipulação de Listas do Scratch, recorrendo ao software (exposição de conteúdos/interacção dos alunos com o software) 	40'	<ul style="list-style-type: none"> - Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> • Dominar a utilização de listas no Scratch 	Estruturas de controle	<ul style="list-style-type: none"> - Realização de uma ficha de trabalho para consolidação de conhecimentos adquiridos na aula 	40'	<ul style="list-style-type: none"> - Apresentação eletrónica (PPT) - Quadro e canetas 	Avaliação formativa (envio dos exercícios resolvido para a plataforma moodle)
<ul style="list-style-type: none"> • Aplicar os conhecimentos adquiridos na realização da ficha prática 		<ul style="list-style-type: none"> - Síntese da aula e apresentação dos objetivos para a aula seguinte. 	5'		

Plano de aula 6 (supervisionada)

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula N°XXX Unidade 1 : Introdução à Programação		

Sumário: Algoritmos de ordenação – Ordenação por seleção. Implementação do algoritmo de ordenação por seleção no Scratch

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 07 de Fevereiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> • Percepcionar os conhecimentos prévios dos alunos 		<ul style="list-style-type: none"> - Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior, análise de alguns programas implementados pelos alunos relativos à resolução do exercício prático (organizador prévio) 	1'		
<ul style="list-style-type: none"> • Esclarecer para que são usados os algoritmos de ordenação 	Utilidade dos Algoritmos de ordenação	<ul style="list-style-type: none"> - Interrogação aos alunos sobre a necessidade de ordenar dados, dar como exemplo as páginas amarelas e a dificuldade de encontrar informação no caso dos dados não estarem ordenados de forma alfabética (exposição de conteúdos) 	10'	<ul style="list-style-type: none"> - Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" - Apresentação eletrónica (PPT) 	Observação direta
<ul style="list-style-type: none"> • Definir o que é ordenar 	Definição de ordenar	<ul style="list-style-type: none"> - Apresentar definição do que é ordenar (exposição de conteúdos) 	5'	<ul style="list-style-type: none"> - Quadro e canetas 	

<ul style="list-style-type: none"> • Ordenar, utilizando o algoritmo de selecção 	Algoritmo de ordenação por selecção	<ul style="list-style-type: none"> - Apresentar aos alunos o método de ordenação através do algoritmo de selecção. Mostrar como funciona o algoritmo através de um exemplo de ordenação com uma pequena lista de valores, solicitar a ajuda dos alunos na traçagem do algoritmo (exposição de conteúdos, interação com os alunos) - Apresentar o pseudocódigo para posterior implementação do algoritmo. (exposição de conteúdos) 	15'		Observação direta	
	Estruturas de controle do scratch, listas e variáveis	<ul style="list-style-type: none"> - Implementar o algoritmo de selecção no scratch para consolidação de conhecimentos adquiridos na aula. Se os alunos terminarem antes do tempo previsto, colocar o desafio para adaptar o algoritmo de forma a ser apresentada a pontuação dos jogadores no final do jogo do projecto individual - Síntese da aula e apresentação dos objetivos para a aula seguinte. 	5'		30'	Avaliação formativa (envio do algoritmo resolvido para a plataforma moodle)
			5'			

Plano de aula 7



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Aplicações Informáticas B

Sumário: Introdução aos RPG. Realização de uma ficha prática.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 14 de Fevereiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 	Jogos RPG	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) 	1' 1' 2' 3'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor 	Observação direta
<ul style="list-style-type: none"> Utilizar e manipular as instruções do Scratch 	. Uso de instruções do Scratch	<ul style="list-style-type: none"> Introdução aos jogos RPG "Role Playing games" 	40'	<ul style="list-style-type: none"> Moodle Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> Aplicar os conhecimentos adquiridos no desenvolvimento de um jogo RPG 	.	<ul style="list-style-type: none"> Realização de uma ficha de trabalho para realização de um jogo RPG 	40' 5'	<ul style="list-style-type: none"> Apresentação eletrónica (PPT) Quadro e canetas 	Avaliação formativa (envio do RPG resolvido para a plataforma moodle)

Plano de aula 8



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Plano de Aula N°XXX Unidade 1 : Introdução à Programação

Sumário: Introdução aos Jogos Plataforma. Realização de uma ficha prática.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 21 de Fevereiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 	Jogos Plataforma	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Introdução aos jogos de plataformas 	1' 1' 2' 3' 40'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" 	Observação direta
<ul style="list-style-type: none"> Utilizar e manipular as instruções do Scratch 	<ul style="list-style-type: none"> Uso de instruções do Scratch 	<ul style="list-style-type: none"> Realização de uma ficha de trabalho para realização de um jogo Ping Pong 	40'	<ul style="list-style-type: none"> Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta
<ul style="list-style-type: none"> Aplicar os conhecimentos adquiridos no desenvolvimento de um jogo Ping Pong 	.		5'		Avaliação formativa (envio do jogo resolvido para a plataforma moodle)

Plano de aula 9

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula N°XXX Unidade 1 : Introdução à Programação		

Sumário: Desenvolvimento dos projectos individuais. Esclarecimento de dúvidas.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 28 de Fevereiro de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> • Percepcionar os conhecimentos prévios dos alunos • Aplicar os conhecimentos adquiridos no desenvolvimento de um projecto individual em Scratch 	Instruções do Scratch e algoritmia	- Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio)	2'	- Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" - Apresentação eletrónica (PPT) - Quadro e canetas	Observação direta
		- Realização dos projectos individuais em sala de aula. Esclarecimento de dúvidas	80'		Observação direta
		- Síntese da aula e apresentação dos objetivos para a aula seguinte.	5'		Avaliação formativa (envio do projecto para a plataforma moodle)

Plano de aula 10

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula N°XXX Unidade 1 : Introdução à Programação		

Sumário: Desenvolvimento dos projectos individuais. Esclarecimento de dúvidas.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 05 de Março de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 		<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) 	2' 3'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação 	Observação direta
<ul style="list-style-type: none"> Aplicar os conhecimentos adquiridos no desenvolvimento de um projecto individual em Scratch 	Instruções do Scratch e algoritmia	<ul style="list-style-type: none"> Realização dos projectos individuais em sala de aula. Esclarecimento de dúvidas 	80'	<ul style="list-style-type: none"> Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta
		<ul style="list-style-type: none"> Síntese da aula e apresentação dos objetivos para a aula seguinte. 	5'		Avaliação formativa (envio do projecto para a plataforma moodle)

Plano de aula 11

 <p>Escola Secundária da Boa Nova LEÇA DA PALMEIRA</p>	<p>Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013</p> <p style="text-align: right;">Aplicações Informáticas B</p>
<p>Plano de Aula N°XXX Unidade 1 : Introdução à Programação</p>	

Sumário: Desenvolvimento dos projectos individuais. Esclarecimento de dúvidas.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 07 de Março de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 		<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) 	<p>2'</p> <p>3'</p>	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação 	<p>Observação direta</p>
<ul style="list-style-type: none"> Aplicar os conhecimentos adquiridos no desenvolvimento de um projecto individual em Scratch 	Instruções do Scratch e algoritmia	<ul style="list-style-type: none"> Realização dos projectos individuais em sala de aula. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	<p>80'</p> <p>5'</p>	<ul style="list-style-type: none"> "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	<p>Observação direta</p> <p>Avaliação formativa (envio do projecto para a plataforma moodle)</p>

Plano de aula 13

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula N°XXX Unidade 1 : Introdução à Programação		

Sumário: Apresentação dos projectos individuais. Autoavaliação

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 14 de Março de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> • Percepcionar os conhecimentos prévios dos alunos • Mostrar os conhecimentos adquiridos no desenvolvimento de um projecto individual em Scratch 	Instruções do Scratch e algoritmia	- Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio)	2' 3'	- Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" - Apresentação eletrónica (PPT) - Quadro e canetas	Observação direta Observação direta
		- Apresentação dos projectos individuais na sala de aula.	40'		Avaliação das apresentações
		- Autoavaliação.	45'		

Plano de aula 14



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Sumário: Apresentação do projecto de trabalho de grupo Zschool. Escolha dos grupos de trabalho e divisão de tarefas.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 16 de Abril de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos Apresentar trabalho de grupo Zschool Escolha dos grupos de trabalho e atribuição das tarefas 	Trabalho em equipa	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Escolha dos grupos de trabalho Atribuição de tarefas aos grupos Síntese da aula e apresentação dos objetivos para a aula seguinte. 	<p>2'</p> <p>3'</p> <p>50'</p> <p>30'</p> <p>5'</p>	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta

Plano de aula 15



Escola Secundária da Boa Nova
LEÇA DA PALMEIRA

Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Sumário: Apresentação das ferramentas de trabalho cooperativo. Realização dos trabalhos de grupo.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 18 de Abril de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	<p>Trabalho em equipa</p> <p>Aplicação de conhecimentos de programação em Scratch e Algoritmia</p>	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Apresentação das ferramentas de trabalho cooperativo Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	<p>2'</p> <p>3'</p> <p>50'</p> <p>30'</p> <p>5'</p>	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	<p>Observação direta</p>

Plano de aula 16

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula NºXXX Unidade 1 : Introdução à Programação		

Sumário: Realização dos trabalhos de grupo.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 23 de Abril de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	Trabalho em equipa Aplicação de conhecimentos de programação em Scratch e Algoritmia	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	2' 3' 80' 5'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"

Plano de aula 18

	Plano de Aula N°XXX Unidade 1 : Introdução à Programação	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013 Aplicações Informáticas B
---	---	--

Sumário: Realização dos trabalhos de grupo.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 02 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> • Realização do trabalho de grupo Zschool 	Trabalho em equipa Aplicação de conhecimentos de programação em Scratch e Algoritmia	<ul style="list-style-type: none"> - Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula - Interligação com os conteúdos aula anterior (organizador prévio) - Realização do trabalho de grupo. Esclarecimento de dúvidas - Síntese da aula e apresentação dos objetivos para a aula seguinte. 	2' 3' 80' 5'	<ul style="list-style-type: none"> - Livro de ponto - Computadores - Vídeo Projetor - Moodle - Utilização do software de programação "Scratch" - Apresentação eletrónica (PPT) - Quadro e canetas 	Observação direta Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"

Plano de aula 19 (supervisionada)



Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Aplicações Informáticas B

Sumário: Apresentação de um protótipo de jogo de forma a ultrapassar dificuldades identificadas

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 07 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Percepcionar os conhecimentos prévios dos alunos 		<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior(organizador prévio) 	<p>1'</p> <p>1'</p> <p>2'</p> <p>15'</p>		
<ul style="list-style-type: none"> Apresentar protótipo, identificar situações onde os grupos estão com dificuldade 	programação no desenvolvimento de jogos	<ul style="list-style-type: none"> Apresentar o protótipo e jogar mostrando as semelhanças em termos de desenvolvimento com o projecto final de grupo (exposição de conteúdos). Identificar partes do código e mostrar aos diferentes grupo possíveis soluções para os problemas com que se depararam até ao momento. 	25'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) 	Observação direta
<ul style="list-style-type: none"> Continuar com a realização dos trabalhos de grupo com auxílio do protótipo de forma a ultrapassar as dificuldades identificadas 	Programação em scratch	<ul style="list-style-type: none"> Continuar com os trabalhos de grupo recorrendo ao protótipo. Ao longo desta fase da aula vou deslocar-me por entre os grupos e inquirir os alunos em que partes o protótipo poderá ser útil ou não. 	40'	<ul style="list-style-type: none"> Quadro e canetas 	Observação direta

		- Síntese da aula e apresentação dos objetivos para a aula seguinte.	5'		Avaliação formativa (envio dos trabalhos desenvolvidos para a dropbox)
--	--	--	----	--	--

Plano de aula 20



Plano de Aula N^oXXX Unidade 1 : Introdução à Programação

Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Sumário: Realização dos trabalhos de grupo.

Turma: 12^o B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 14 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	<p>Trabalho em equipa</p> <p>Aplicação de conhecimentos de programação em Scratch e Algoritmia</p>	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	<p>2'</p> <p>3'</p> <p>80'</p> <p>5'</p>	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	<p>Observação direta</p> <p>Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"</p>

Plano de aula 21

	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013	Aplicações Informáticas B
Plano de Aula NºXXX Unidade 1 : Introdução à Programação		

Sumário: Realização dos trabalhos de grupo.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 21 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	Trabalho em equipa Aplicação de conhecimentos de programação em Scratch e Algoritmia	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	2' 3' 80' 5'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"

Plano de aula 22

	Plano de Aula N°XXX Unidade 1 : Introdução à Programação	Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques Escola Secundária da Boa Nova Ano Letivo 2012/2013 Aplicações Informáticas B
---	---	--

Sumário: Realização dos trabalhos de grupo.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 28 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	Trabalho em equipa Aplicação de conhecimentos de programação em Scratch e Algoritmia	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	2' 3' 80' 5'	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	Observação direta Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"

Plano de aula 23



Plano de Aula N^oXXX Unidade 1 : Introdução à Programação

Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Sumário: Realização dos trabalhos de grupo.

Turma: 12^o B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 30 de Maio de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Realização do trabalho de grupo Zschool 	<p>Trabalho em equipa</p> <p>Aplicação de conhecimentos de programação em Scratch e Algoritmia</p>	<ul style="list-style-type: none"> Efetuar a chamada Apresentar o sumário Apresentação do tema e objetivos da aula Interligação com os conteúdos aula anterior (organizador prévio) Realização do trabalho de grupo. Esclarecimento de dúvidas Síntese da aula e apresentação dos objetivos para a aula seguinte. 	<p>2'</p> <p>3'</p> <p>80'</p> <p>5'</p>	<ul style="list-style-type: none"> Livro de ponto Computadores Vídeo Projetor Moodle Utilização do software de programação "Scratch" Apresentação eletrónica (PPT) Quadro e canetas 	<p>Observação direta</p> <p>Avaliação formativa. Envio dos ficheiros em Scratch desenvolvidos para a "dropbox"</p>

Plano de aula 24



Escola Secundária da Boa Nova
LEÇA DA PALMEIRA

Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Aplicações Informáticas B

Sumário: Avaliação final do projecto. Realização de Focus Group com parte dos alunos da turma.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 04 de Junho de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
<ul style="list-style-type: none"> Avaliar o projecto Zschool 		<ul style="list-style-type: none"> - Efetuar a chamada - Apresentar o sumário - Apresentação do tema e objetivos da aula 	<p>1'</p> <p>1'</p> <p>2'</p>	<ul style="list-style-type: none"> - Livro de ponto - Computador portátil 	<p>Observação directa e gravação áudio do Focus Group</p>
		<ul style="list-style-type: none"> - Realização de Focus Group com os mesmo alunos que realizaram o Focus group inicial. 	<p>86'</p>		

Plano de aula 25



Plano de Aula NºXXX Unidade 1 : Introdução à Programação

Prof. Orientador: F. Filipe / Prof. Estagiário: Miguel Marques
Escola Secundária da Boa Nova
Ano Letivo 2012/2013

Aplicações Informáticas B

Sumário: Auto-avaliação.

Turma: 12º B
Horário: 10:15 às 11:45 – 90 Minutos
Data: 06 de Junho de 2013



Objectivos	Conteúdos	Estratégias/Atividades	Tmp	Recursos	Avaliação
• Auto-avaliação		- Efetuar a chamada	1'		
		- Apresentar o sumário	1'		
		- Apresentação do tema e objetivos da aula	2'		
		- Auto-avaliação por parte dos alunos.	86'	- Livro de ponto - Computador portátil	Observação directa

Anexo 2 - Grelha de registo de avaliação heurística da usabilidade

Grelha de registo de avaliação heurística com especialistas: Avaliação da usabilidade

Nome do Avaliador:

Nome do Protótipo:

Data da avaliação: ____ / ____ / ____

Tempo de início: ____ horas ____ min

Tempo de término: ____ horas ____ min

Grau de severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface
1	Cosmético	Não há necessidade imediata de solução
2	Simple	Problema de baixa prioridade (pode ser reparado)
3	Grave	Problema de alta prioridade (deve ser reparado)
4	Catastrófico	Muito grave, deve ser reparado de qualquer forma

Descrição do problema	Proposta de solução	Severidade	Heurísticas de Nielsen
			1- Visibilidade do estado do sistema O sistema deve manter o utilizador informado sobre o que está a acontecer, através de feedback adequado; a resposta do sistema deve ser imediata (para tempos superiores a 10 seg o sistema deve possuir indicadores de progresso) O jogador é sempre capaz de identificar a sua pontuação/status no jogo? São utilizados sons para fornecer feedback significativo? A cada comando do jogador existe uma resposta do sistema?
			2- Semelhança entre o sistema e o mundo real A linguagem colocada no interface deve conter termos e conceitos familiares ao utilizador; devem-se seguir as orientações do mundo real, em que a informação apareça numa ordem natural e lógica O jogo está de acordo com os padrões dos jogos digitais de forma a minimizar a curva de aprendizagem? O jogador é envolvido no jogo de forma fácil? O jogo utiliza linguagem familiar ao utilizador (palavras, frases, conceitos familiares)?

Avaliação e Concepção de Materiais Escolares de Informática

Grelha de registo de avaliação heurística com especialistas: Avaliação da usabilidade

Descrição do problema	Proposta de solução	Severidade	Heurísticas de Nielsen
			<p>3- Controlo e liberdade O utilizador deve poder desfazer e refazer ações, retornar ao estado anterior ou seguir adiante o utilizador; deve poder interromper processos e sair sempre que quiser</p> <p>Os movimentos do personagem são facilmente controláveis? É permitido sair do jogo em qualquer altura? É permitido pausar o jogo em qualquer altura? É fácil encontrar as opções disponíveis?</p>
			<p>4- Consistência e Padrões Devem-se seguir as convenções em termos de linguagem e ter o mesmo critério ao longo de todo o sistema; o utilizador não deve ter que pensar se palavras diferentes significam a mesma coisa</p> <p>A interface do jogo é consistente (controlo, cor, tipografia, menu e design de diálogo)? O método de movimentação do cursor e das teclas ao longo do jogo é consistente?</p>
			<p>5- Prevenção de erros Melhor que boas mensagens de erro é ter um projeto que previne a ocorrência de problemas; avisar quando processos de risco estiverem a ser realizados; todos os processos devem ser fáceis de serem concluídos</p> <p>O jogo previne erros antes de eles acontecerem, através de mensagens de aviso como "Tem certeza de que quer sair?"</p>
			<p>6- Reconhece ao invés de lembrar Devem-se tornar os objetos, ações e opções visíveis pois o utilizador não deve ter que (re)lembrar informação; instruções para o uso do sistema devem estar visíveis ou facilmente recuperáveis sempre que necessário</p> <p>As sugestões (mensagens) estão corretamente posicionadas?</p>

Grelha de registo de avaliação heurística com especialistas: Avaliação da usabilidade

Descrição do problema	Proposta de solução	Severidade	Heurísticas de Nielsen
			<p>7- Flexibilidade e eficiência de utilização O sistema deve oferecer mais do que uma maneira do utilizador cumprir a tarefa; o sistema deve poder atender tanto a utilizadores novatos como experientes</p> <p>O interface é simples o suficiente para o jogador iniciado? O jogador avançado pode usar atalhos para melhorar o seu desempenho?</p>
			<p>8- Estética e desenho minimalista Os diálogos não devem conter informação irrelevante; textos concisos, demasiado texto e imagens distraem o utilizador layout limpo e simples</p> <p>O interface é visualmente agradável? O menu encaixa no estilo gráfico? O menu não interfere com a área onde ocorre o jogo?</p>
			<p>10- Ajuda e documentação O sistema deve poder ser usado sem documentação de ajuda; as informações necessárias para completar a tarefa devem estar disponíveis a cada momento; a ajuda deve ser fácil de aceder e focada na tarefa do utilizador, relacionar passos concretos a serem dados, não muito longa</p> <p>O jogo é apresentado por meio de um introdução interessante e esclarecedora? A função "ajuda" é visível? O interface de ajuda é consistente com os interfaces do protótipo que o suporta? A informação de ajuda é objetiva (o que eu posso fazer no jogo)? São fornecidos pequenos itens de ajuda no decorrer do jogo, como por exemplo, personagens que aparecem à medida que o jogo evolui, com dicas e informações dos próximos passos a seguir? Os utilizadores podem facilmente comutar entre a ajuda e o jogo e recomeçar o jogo onde tinham parado?</p>

Anexo 3 - Grelha de registo de avaliação heurística do divertimento

Grelha de registo de avaliação heurística com especialistas: Avaliação do divertimento

Nome do Protótipo: _____		Data da avaliação: ____/____/____			
		Tempo de início: ____horas ____min			
		Tempo de término: ____horas ____min			
Escala de Severidade (Thumbs-Up Scale)					
1 Muito pobre 2 Pobre 3 OK 4 Bom 5 Muito bom					
Heurísticas de Malone		Heurística	Descrição do problema	Proposta de solução	Severidade
<p style="text-align: center;">Desafio</p> 1 São claros os objetivos a serem atingidos em todas as etapas do jogo? 2 Existe informação, sempre que necessário, do quão distante estás de cumprir os objetivo? 3 O nível de dificuldade é desafiador ao longo do jogo?					
<p style="text-align: center;">Fantasia</p> 4 O jogo apela às minhas emoções? 5 Todos os elementos do jogo são fáceis de reconhecer?					
<p style="text-align: center;">Curiosidade</p> 6 Os elementos do jogo são intuitivos na sua utilização? 7 Sempre que recomences podes saltar a introdução para não estares à espera? 8 O jogo é divertido?					

Anexo 5 – Guião do *Focus Group* inicial

Objetivo – avaliar a motivação dos alunos pelos conteúdos da disciplina de Aplicações Informáticas (12.º ano) e pelas metodologias de ensino-aprendizagem

- 1- Antes de começarem o ano letivo, o que esperavam aprender na disciplina de Aplicações Informáticas e de que forma (através de aulas teóricas, exercícios práticos individuais, trabalhos de grupo)?
- 2- A disciplina de Aplicações Informáticas está a corresponder às vossas expectativas em termos de conteúdos?
- 3- Estavam “a contar” ter programação?
- 4- O que acham da programação? (Desafiante? Frustrante? Difícil? Horrível? Maravilhosa?)
- 5- Expectativas para o módulo de multimédia?

Objetivo – avaliar as perspetivas dos alunos para o futuro

- 6- Quem quer seguir a área da informática? E área da multimédia?
- 7- No ensino superior, como pensam que será a aprendizagem? Através de fichas de trabalho individual? Em grupo? Mais autonomamente?
- 8- E no mundo do trabalho? Imaginam-se isolados num gabinete onde podem desenvolver o vosso trabalho individualmente ou integrados numa equipa multidisciplinar?
- 9- Para o desenvolvimento de um produto qualquer na área da informática (site, base de dados, videoclip, software educativo, um jogo,...) que elementos (recursos humanos) é que acham que são necessários?

Objetivo – avaliar a percepção dos alunos de Artes sobre os de Ciências e Tecnologias e vice-versa

- 10- Há uns mais importantes do que outros? (consoante as repostas podemos provocá-los, de forma a revelarem o que pensam dos alunos das áreas das ciências e tecnologias vs Artes com observações do tipo “O produto pode estar muito bonito, mas sem a programação não vai funcionar...” ou “O produto pode estar a funcionar na perfeição, mas o utilizador pode não se sentir satisfeito ao utilizá-lo por questões ligadas, por exemplo, ao design do interface, cores, ícones, etc...”) No final desta questão a “moral” terá de ser sempre a de que todos são imprescindíveis.

11- O que acham de fazer um projeto em que tenham de trabalhar em equipa, mas na área que mais gostam (programação, design)?

Objetivo – avaliar o conhecimento e interesse dos alunos por determinadas ferramentas

12- Conhecem o *Scratch*? Sabem para que serve?

13- Estariam interessados em trabalhar com esta ferramenta para criar, por exemplo, um jogo?

14- E que ferramentas utilizariam se tivessem de realizar um trabalho em equipa à distância?

15- O que acham da inserção das redes sociais (como o *Facebook*) para desenvolver trabalhos em grupo?

Objetivo – avaliar a predisposição dos alunos para um trabalho colaborativo/cooperativo

16- Estariam dispostos a trabalhar em parceria com a outra turma? Sim, não? Porquê?

Anexo 6 – Guião do *Focus Group* final

Objetivo – Avaliar a evolução das expectativas dos alunos, no decorrer da disciplina, em relação aos conteúdos de Aplicações Informáticas (12 º ano) e às metodologias de ensino-aprendizagem

- 1- Qual é o balanço geral que fazem da disciplina ao longo do ano?
- 2- Que parte dos conteúdos e trabalhos desenvolvidos ao longo do ano gostaram mais? Da programação Pascal ou do Projeto “Zschool”?
- 3- Consideraram mais motivantes as metodologias baseadas em fichas de trabalho ou no desenvolvimento de projetos individuais/grupo?

Objetivo – Avaliar a perceção dos alunos relativamente à utilidade do projeto colaborativo/cooperativo para o contexto do ensino superior ou de trabalho

- 4- Que tipos de competências sociais e cognitivas acham que desenvolveram com este projeto? (autonomia, espírito crítico, capacidade de comunicação/argumentação, programação, tratamento de som, imagem, etc.)
- 5- Consideram que a metodologia de trabalho utilizada neste projeto será útil para o vosso futuro académico e profissional?

Objetivo – Avaliar a perceção dos alunos de Artes sobre os de Ciências e Tecnologias e vice-versa

- 6- Consideram que, neste projeto, houve alguma das duas turmas que teve um papel mais importante do que a outra?
- 7- Como avaliam a comunicação com a outra turma? (sentiram que os colegas correspondiam aos vossos pedidos, que respeitaram as vossas opiniões...)
- 8- Acham que era importante trabalharem no mesmo espaço físico?
- 9- Se tivessem de trabalhar com alguém da outra turma, que tipo de perfil acham que deveria ter esse colega? Como acham que poderiam trabalhar? Presencialmente, através da Internet...?

Objetivo – Avaliar a perceção da utilidade e interesse dos alunos por determinadas ferramentas

- 10- Durante o desenvolvimento deste projeto, utilizaram o *Facebook*, a *Dropbox* e a plataforma Moodle. Como avaliam a importância dessas ferramentas para o desenvolvimento do trabalho? Eliminariam alguma? De qual gostaram mais? Qual a mais útil?

11- (12ºB) Como avaliam a importância da utilização do Scratch para a criação do jogo?

Objetivo – avaliar a predisposição dos alunos para o trabalho colaborativo

12- Se voltassem a fazer um projeto com esta metodologia, o que mudariam?

Anexo 4 – Guião do jogo *Zschool*

Cena 0: Introdução

Plano de Fundo: Logotipo do jogo

Descrição: Introdução do jogo

Áudio: música de fundo

Imagens: Logotipo, botão de texto

Animação: Animação do logotipo do jogo e botão “saltar introdução”

Navegação: botão “saltar introdução” avança para a cena 2.

Cena 1: Cenário inicial

Plano de Fundo: Portão da escola.

Descrição:

Um aluno chega à escola secundária da Boa Nova e apercebe-se que aconteceu algo de estranho. Entretanto um ser rastejante aproxima-se e explica-lhe que a escola foi inexplicavelmente invadida por zombies. Essa mesma figura revela que ele próprio tentou salvar a escola mas não conseguiu deixando nas mãos do aluno a tarefa de encontrar o antídoto que curará as pessoas infetadas. Entrega-lhe uma régua que terá de usar para se defender dos zombies e diz-lhe que deve procurar a receita do antídoto na biblioteca, buscar os ingredientes na cantina e, finalmente, prepará-lo no laboratório de química.

Diálogo

NPC (fala 1) - Socorro, ajuda-me! A escola foi invadida por zombies!

Aluno (fala2)- Que se passou?

NPC (fala3)- Quando cheguei à escola tudo estava normal até que começaram a aparecer zombies um de cada vez e a infetar as pessoas. Tentei resolver a situação, mas fracassei.

Aluno (fala4)– E o que é que fizeste?

NPC (fala5)- Na biblioteca perdi um livro com uma poção para curar zombies. Usei também esta régua (dá-lhe) para me defender e esta página do livro que consegui arrancar ao fugir. A escola depende de ti!

Áudio: Gritos de pessoas, e barulhos de zombies, portões a ranger, diálogo

Imagens: Pessoa a rastejar até ao personagem, régua, página do livro, personagem principal, ícones (escudo, vidas e energia), cenário de fundo (portão da escola).

Animação: Personagem a caminhar em direção ao nível inicial.

Transição: Zoom para pátio da escola

Cena 2: Passagem para a biblioteca

Plano de Fundo: Pátio da escola (Dentro da escola em frente ao polivalente)

Descrição: Dadas as instruções do NPC (ser rastejante), o aluno terá de ultrapassar vários zombies para conseguir chegar à biblioteca. Com a régua terá de destruir os zombies.

Diálogo

NPC - Para chegares à biblioteca vais ter de passar por estes zombies que te vão tentar atacar. Usa a régua para os matares. Cada vez que um zombie te tocar vais perdendo energia e quando ficares sem energia perdes uma vida e voltas ao início do minijogo.

Áudio: Gritos de pessoas, e barulhos de zombies, portões a ranger, barulho da régua a matar os zombies, zombies a morrer.

Imagens: zombies, personagem, régua.

Animação: Zombies a caminhar na direção da personagem. Personagem a destruí-los com a régua, personagem a mover-se.

Navegação: Setas (cima, baixo, direita, esquerda) para movimentação do personagem principal, espaço para atacar.

Transição: Escurecer

Cena 3: Biblioteca

Plano de Fundo: Biblioteca

Descrição: Na biblioteca, o aluno terá de encontrar as folhas do livro onde consta a receita do antídoto. Entretanto, precisa de avançar os obstáculos e destruir os zombies que vão surgindo.

Diálogo

NPC - Apanha todas as páginas do livro de química para descobrires o antídoto. Salta por cima dos obstáculos para te desviares, destrói os zombies com a régua, usando a tecla space. Usa as setas para te moveres e, para saltar mais alto, carrega duas vezes seguidas na seta 'cima'.

Áudio: Voz que dá as indicações, barulho dos zombies a serem esmagados e música fundo.

Imagens: Imagem de fundo-biblioteca, obstáculos e elementos do cenário (estantes, monitores, livros, etc).

Animação: Zombies a caminhar na direção da personagem, Personagem a destruí-los com a régua, personagem a mover-se.

Navegação: Setas (cima, baixo, direita, esquerda) para movimentação do personagem principal, espaço para atacar, duas vezes na seta cima para saltar.

Transição: Escurecer

Cena 4: Cantina

Plano de Fundo: Cantina

Descrição: Na cantina, o aluno é confrontado com uma funcionária zombie que lhe atira vários objetos. O aluno terá de apanhar os alimentos "bons", fugir dos objetos maus e apanhar os talheres: as bombas e os pratos tiram energia; os alimentos dão energia e os talheres servirão de armas para destruir a funcionária zombie.

Depois disso, o aluno deverá destruir a funcionária. Para isso, atira-lhe os talheres que reuniu para destruir a funcionária zombie enquanto esta se vai desviando. Quando esta morre, faz drop do item da cura para a barra do interface dos elementos recolhidos.

Diálogo

(Início) NPC - Desvia-te das bombas, apanha a comida boa para energia e os talheres para te defenderes.

(Depois do 1º desafio) NPC - Agora que reuniste os itens necessários, destrói a funcionária para conseguires o elemento para a cura, atirando-lhe talheres. Usa as setas para te moveres e o space para atirar os talheres.

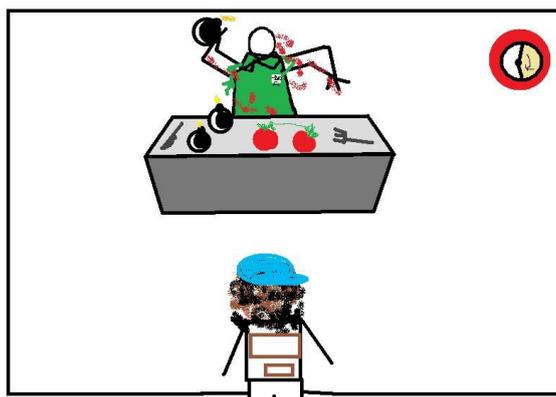
Áudio: Barulho de zombies; Som de atirar coisas; Voz off do NPC; som_menosvida; som_maisvida; som_armas (talheres);

Imagens: Bomba, Pratos, Garfo, Faca, Maçã, Banana, Brócolos, Funcionária, Relógio, Item da cura (planta)

Animação: funcionária zombie a atirar os objetos, aluno a deslocar-se para a direita e esquerda, objetos a voar pelo ar, funcionária a deslocar-se para a esquerda e direita.

Navegação: Setas do teclado (esquerda, direita e baixar);

Transição: Escurecer



Cena 5: Laboratório de química

Plano de Fundo: laboratório de química

Descrição: Conseguido o elemento da cura, o aluno chega ao laboratório de química e prepara o antídoto. Mistura ingredientes disponíveis na bancada com o item da cura, recorrendo ao livro com a receita. Dá-se uma explosão e todos os zombies da escola se transformam novamente em humanos. A escola está salva!

Áudio: Barulho de zombies (vindo do exterior); som de produtos químicos a borbulhar, música de fundo, som explosão

Imagens: Bomba, Pratos, Garfo, Faca, Maçã, Banana, Brócolos, Funcionária, Relógio, Item da cura (planta)

Animação: aluno a misturar os ingredientes, explosão

Transição: Explosão

Cena 6: Cenário de sucesso

Plano de Fundo: Pátio da escola (cenário da cena 2)

Descrição: Os zombies transformam-se todos em humanos.

Áudio: Barulho de zombies; música de fundo

Imagens: cenário de fundo, Zombies

Animação: zombies a transformar-se em humanos.

Transição: escurecer

Cena 7: Cenário de insucesso

Plano de Fundo: Grande plano da personagem principal

Descrição: O aluno transforma-se em zombie.

Áudio: musica de derrota, barulho de zombie

Imagens: personagem principal humano e zombie

Animação:. Transformação do aluno em zombie

Transição: Escurecer e volta ao início do jogo

Créditos

Plano de Fundo: Texto com os autores e respectivas funções

Áudio: música defundo

Animação:. Texto em scroll automático

Transição: Escurecer e volta ao início do jogo

Jogabilidade - cenas 2, 3, 4

Cada mini jogo inicia-se com:

- 5 quadrados de energia
- 3 vidas
- 1 escudo

-Destrói os zombies a cada 6 reguadas que acerta no zombie mas cada vez que é tocado, perde 1 reguada (ex. deu 4 reguadas sem ser tocado pelo zombie – faltam 2- mas depois é tocado passam a faltar 3 reguadas para o destruir)

- Perde 1 quadrado de energia cada 2 vezes que é tocado por um zombie

- Quando termina a energia perde uma vida + perde capacidade de dar reguadas (tem de ir fugindo dos zombies durante o tempo sem energia e, se necessitar, usar o escudo) – Ao fim de 5 segundos ganha 1 quadrado de energia com o qual terá de continuar o jogo. Se o perder, volta a a perder uma vida.

- O escudo só pode ser utilizado 1 vez

- Para passar de nível tem de destruir 4 zombies

- Perde quando acabam as vidas

- Na cantina, a funcionária zombie é destruída ao fim de ser atingida por 4 talheres.

Ajuda

Todas as cenas têm uma aba na barra inferior que permite abrir e fechar com um pequeno texto de ajuda. Para além disso, contém a opção para ouvir a ajuda do NPC.