

Relatório

Projeto Final de Laboratório Multimédia 3

Sunshine's Quest

Gabriela Ribeiro, Joana Matos, Luís Ferreira, Pedro Sousa, Rodrigo Coelho
102391, 103579, 103572, 104232, 12525

{Gabriela.ribeiro@ua.pt , jlmatos@ua.pt , laferreira@ua.pt , pedrohpsousa@ua.pt ,
rodrigo.coelho@ua.pt }

Abstract

Este relatório pretende descrever os processos e componentes de código, conceptuais e artísticas utilizados na realização do projeto final da cadeira de Laboratório Multimédia 3, servindo como um apoio à compreensão das decisões tomadas e do código desenvolvido ao longo do projeto.

Keywords: Plataformas, JavaScript, Jogo, Heroína, Erosão Costeira, Ílhavo, Colisões, Elementos, Arte

1. Introdução

O projeto final da cadeira de Laboratório Multimédia 3 consiste em desenvolver um jogo com base em linguagens de programação como HTML, CSS e JavaScript. Para atribuir uma temática aos jogos a desenvolver foi feita uma parceria com a câmara de Ílhavo, que devido à sua localização maioritariamente costeira lançou o desafio de sensibilização para o futuro da mesma, sendo este: a erosão costeira, a proteção do litoral, sensibilização, consciencialização, sustentabilidade e ação social.

Perante a problemática apresentada o nosso grupo decidiu trabalhar numa fantasia que explora as consequências dessa erosão, surgindo assim o nosso projeto final: *Sunshine's Quest*.

Objetivos gerais:

- Narrativa dinâmica em diferentes ambientes;
- Sensibilização da subida dos mares / erosão costeira;
- Consciencialização da causa da subida dos mares (aquecimento global e degelo dos glaciares);
- A premissa principal será a proteção da câmara de Ílhavo;
- Desenvolver um jogo de plataforma;

- Pôr em prática conteúdo lecionado nas aulas, bem como aquisição de novos conhecimentos e métodos de trabalho e aprendizagem;

2. Planeamento e Pré-produção

2.1. Brainstorming

Inicialmente surgiram ideias de desenvolver um jogo de luta, por exemplo *Smash*© (Fig.4). De seguida foi sugerido algo semelhante ao jogo *Journey*© (Fig.5), algo mais direcionado a puzzles e a completar missões. Por fim, agradou-nos a ideia de criar um vilão coletivo, as empresas poluentes, que levarão à criação de demónios, tendo como objetivo, uma heroína da câmara de Ílhavo derrotar estes demónios.



Fig.4



Fig.5

2.2. Público-alvo

Este jogo destina-se a jovens maiores de 14 anos uma vez que apresenta um elevado grau de complexidade a nível de mecânicas, compreensão dos objetivos e, também graficamente.

3. Produção

3.1. Conceito/Narrativa

Sunshine's Quest é um jogo institucional que explora uma fantasia das consequências do aquecimento global e degelo dos glaciares no futuro. O jogador vai ter de deslocar a heroína, Sunshine, por plataformas, e derrotando os demónios que provocam o degelo. Também existem elementos positivos que o utilizador pode optar por apanhar (como por exemplo vida).

Inicialmente o jogador tem 2 vidas, mas pode alcançar mais 2 corações de vida apanhado comida, sendo que após as perder, o jogo acaba. Se o jogador conseguir escapar ileso aos inimigos e obstáculos (buracos), este chega ao demónio final e terá que o derrotar para finalizar o jogo. De seguida o jogador poderá visualizar o tempo e pontos que alcançou.

Ao perder o utilizador vai poder recomençar até conseguir passar o jogo.

Com este jogo, pretendemos sensibilizar o público para o futuro da subida dos mares, mas ao mesmo tempo tornar o tema jogável explorando uma fantasia que consensualize.

3.2. Estado da Arte

Nas nossas pesquisas dividimos o estado da arte em duas vertentes: Funcionalidade e Estética.

Segundo a funcionalidade, o jogo desenvolvido assemelha-se a um jogo de plataforma, por exemplo, *Super Mario*® (Fig.1). O jogador vai poder movimentar a sua personagem conforme.

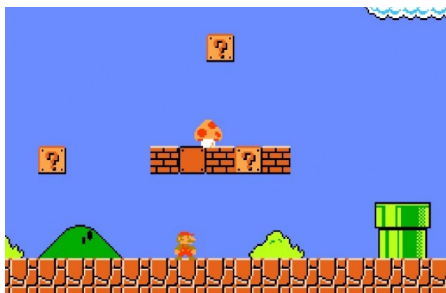


Fig.1

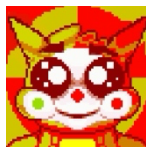


Fig. 2



Fig. 3

Relativamente à estética do jogo, recorremos à *PixelArt* e ao efeito de *parallax* para aumentar o realismo do cenário. Queríamos formas simples, porém estéticas e uma paleta cromática que consistisse em cores quentes (Fig.2 e Fig.3)

3.2.1. Música

Para manter consistência entre as vertentes visuais e auditivas do jogo, optámos também por seguir esta abordagem no que toca à sonoridade do jogo, tanto na parte das músicas utilizadas bem como os efeitos sonoros, tentando reproduzir um efeito "retro" semelhante e característico dos jogos antigos em que nos inspirámos para produzir o nosso próprio jogo. Por forma a alcançarmos este objetivo e termos um resultado

satisfatório bem como uma fácil manipulação do áudio no código, o grupo acabou por produzir toda a "soundtrack" e efeitos sonoros utilizados no projeto.

3.3. Elementos

No ecrã estarão visíveis determinados elementos interativos de forma a permitir um bom funcionamento e boa compreensão da dinâmica do jogo, a significação de cada um dos elementos encontra-se disponível nas instruções do jogo.

3.3.1. Itens Colecionáveis

Ao longo das plataformas o jogador será exposto um item de *powerup* que permite ganhar vidas extra para usar ao longo do jogo

3.3.2. Sistema de Vidas

No início de jogo, o jogador começa com duas vidas, mas poderá aumentá-las infinitamente. Estas estão localizadas no canto superior esquerdo.

3.3.3. Personagem

A nossa heroína é uma habitante de Ílhavo que decide pôr um fim aos demónios, ao partir numa aventura por diferentes cenários até chegar à localização do demónio final.

A personagem é o elemento diretamente controlável pelo jogador, esta é controlada através das teclas a, d e w que permitem o movimento (para a esquerda e para a direita) e salto, respetivamente.

O movimento da personagem foi conseguido através de *sprite* (Fig.6) constituído por diversas imagens em *loop*.



Fig. 6

3.3.4 Inimigos

Ao longo do jogo existem inimigos (Fig.7) (pequenos monstrinhos que se deslocam) quando estes entram em contacto com o player tiram-lhe uma vida, se esta for menor que zero, o jogador perde, o jogo acaba e reinicia. Esta perda de vidas pode ser contornada pelos *powerups* que adicionam uma vida, estes inimigos poderão ser mortos pelas pedras projetadas pelo player.



Fig.7

3.6.1 Pedras

Este mecanismo consiste em itens, pedras, (Fig.12) que são arrecadadas ao longo do jogo e podem ser usadas ao longo do mesmo para matar os inimigos e/ou o boss final. Primeiramente posicionamos as pedras ao longo do percurso, guardamos esse valor e mostramos a quantidade de pedras disponíveis para atirar no canto superior esquerdo. Para a animação deste mecanismo foi necessário mudar os *sprites* da heroína quando apanhada a primeira pedra e quando atira a mesma, (Fig.11) bem como mudar os *sprites* dos inimigos e do boss quando atingidos. Além disso, atribuímos um valor a uma tecla (“espaço”) para atirar a pedra à qual atribuímos gravidade e velocidade à mesma. Após várias tentativas na implementação deste mecanismo, finalmente conseguimos com sucesso a implementação do mesmo.

Caso este mecanismo não resultasse teríamos um plano B, porém não tão dinâmico e eficiente.

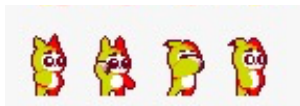
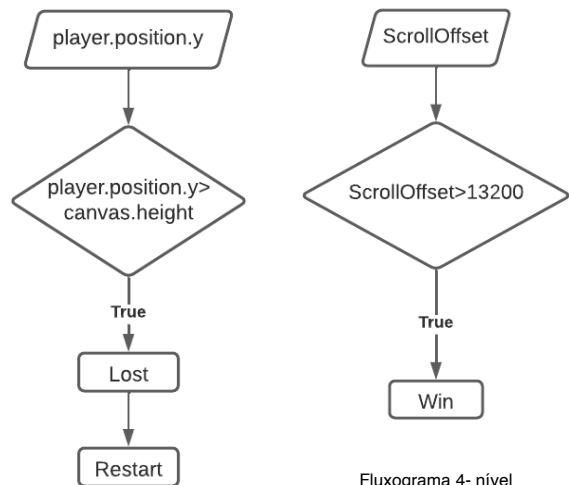
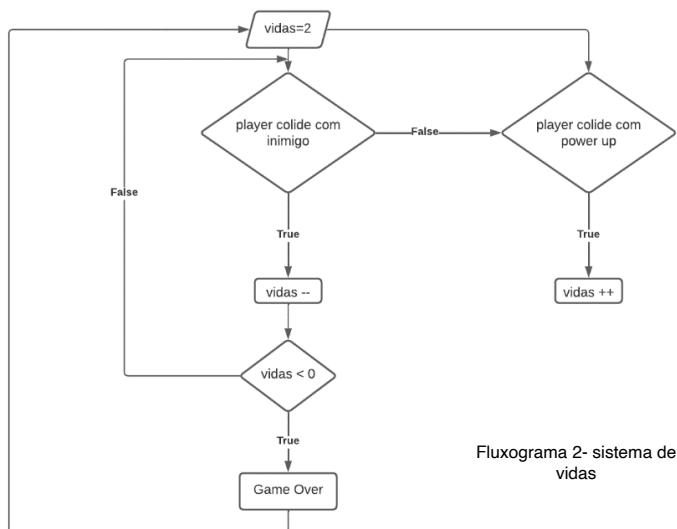
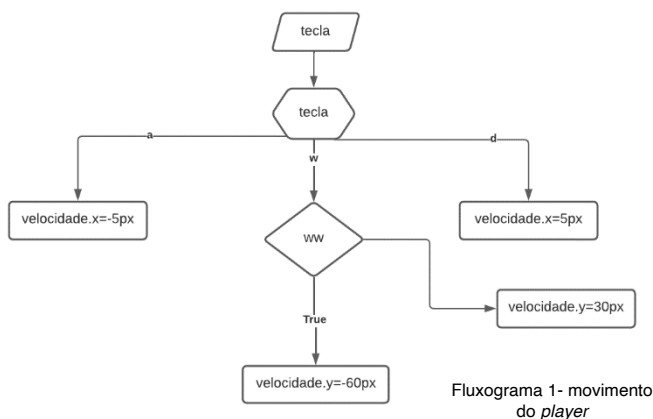


Fig.11



Fig.12

3.4. Fluxograma



Fluxograma 3- mecanismo de queda, morte e restart

Fluxograma 4- nível completo

3.5 Soluções técnicas

Iremos agora desenvolver as soluções técnicas face aos desafios criados pela implementação do nosso conceito. Devido ao conceito de plataforma, irá ser abordado, principalmente, em casos de mecânica, funcionalidades relativas ao código na linguagem de Javascript.

3.5.1 Comandos

Os controlos utilizados em Sunshine's Quest são as teclas “A”, “D”, “W” ao serem pressionadas a personagem move-se 5 px para a esquerda, 5 px para a direita e 30 px para cima, respetivamente (um duplo clique na tecla “W” faz com que a personagem salte duas vezes, 60 px para cima).

3.5.2 Plataformas

Para a construção do mapa de plataformas foi necessário criar três imagens para cada nível com diferentes dimensões de plataforma (Fig.8). Após esse processo estas foram importadas para o nosso código de JavaScript, foram criadas classes para definir posições e tamanhos das mesmas e posteriormente foram localizadas através de *arrays*.

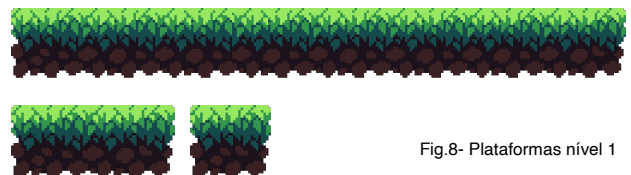


Fig.8- Plataformas nível 1

3.5.3 Parallax

De forma a criar um cenário mais realista e dinâmico, decidimos utilizar o *parallax* nos diferentes níveis (Fig.9 e Fig.10). Assim, para cada nível, os cenários necessitam de ser constituídos por *layers* que se movimentam a velocidades diferentes conforme o movimento do jogador.

Para programar assim a *parallax* foi necessário igualar cada uma destas imagens a um *x* e *y* com valor 0, fazer *position absolute* e adicionar *z-index* diferentes para assim ocuparem toda a resolução do ecrã e ficarem sobrepostas. Por fim, criámos variáveis para definir a velocidade de cada uma destas imagens e uma velocidade geral que nos permitiu realizar a operação e associarmos a cada uma destas imagens dando assim a sensação de movimento.



Fig.9- Parallax nível 1



Fig.10- Parallax nível 2

3.5.4 Itens

No que toca ao funcionamento dos itens, a comida, foi preciso criar uma variável para a imagem da mesma, criar uma *class* (tal como a das plataformas), posicionar várias vezes a mesma imagem através de *arrays*, adicionamos também gravidade e foi necessário detetar a colisão destes itens com as plataformas para estes se manterem na janela (devido à gravidade) e criámos também a colisão entre a Sunshine que implica que estes mesmos desapareçam mal a nossa personagem toque neles.

3.5.5 Colisões

Relativamente às colisões, existem três distintas; a colisão com as plataformas, que evita a queda de todos os elementos pousados nas mesmas ao longo do jogo, a colisão com os inimigos que danificam a personagem retirando-lhe uma vida e mudando os *sprites* da mesma para "*hurt*", e ainda a colisão com objetos que logo depois desaparecem, o bolo e o boss final, no bolo é acrescentada

uma vida e no boss final vai retirando energia ao mesmo. Para facilitar e simplificar o código, uma vez que estas colisões foram utilizadas várias vezes ao longo do código, foram criadas funções para as mesmas. Assim, cada função compara a largura e comprimento dos elementos em questão em parâmetros e guarda este mesmo valor. De seguida ao longo do código, são substituídos os parâmetros por variáveis que representam elementos do jogo, como o *player*, os inimigos, a comida, o boss e as plataformas. Este mesmo processo é inserido num "*If*", assim se esta condição/ comparação mencionada na função for verdadeira, procede-se ao código seguinte (por exemplo, no caso da colisão de qualquer objeto com as plataformas iguala-se a velocidade de queda -gravidade- a 0, assim estes quando colidem com a plataforma ficam pousados).

3.5.6 Relógios

Através do HTML, CSS e JS foi criado um timer que possibilita que o utilizador saiba quanto tempo demorou a ultrapassar o jogo. Foi criada uma *label* no HTML que foi estilizada através do CSS. Já no *JavaScript* foi criado um *setInterval* que permitisse que a contagem tanto dos minutos como dos segundos através de duas operações matemáticas.

3.5.6 Vidas

O nosso sistema de vidas está diretamente interligado, tanto com os vários itens de comida posicionados ao longo da Plataforma, como com a colisão do jogador com o inimigo. Para isto foi necessário criarmos colisões entre estes dois objetos, como já foi dito anteriormente, e após esta, dependendo se é com a comida ou com os inimigos, a vida da jogadora, que está explícita no canto superior esquerdo, vai aumentando ou desce, respetivamente. Para isto ser possível apenas criámos uma variável "vida" e um "*if*" para cada elemento que permitia adicionar ou subtrair uma vida dependendo do objeto em que é realizada a colisão.

3.5.7 Lançamento de pedras

O objetivo do lançamento das pedras é derrotar os inimigos e o boss final. Estas são apanhadas ao longo do caminho e no canto superior esquerdo é possível ver quantas pedras foram arrecadadas ao todo. A estes elementos foram atribuídas imagens, velocidade e gravidade para o movimento das mesmas parecer mais realista, foi também feita a colisão com os elementos a destruir para que, quando esta acontecesse, estes fossem eliminados, no caso dos inimigos, após uma colisão, no caso do boss após cinco. A colisão com as plataformas é também muito importante para que as pedras batam na plataforma e saltem, continuando assim o seu trajeto.

3.6 Ideias abandonadas

3.6.1 Diferentes níveis e mini bosses

Mal o projeto foi nos apresentado o grupo decidiu se juntar para reunirmos diferentes ideias. Inicialmente tínhamos chegado à conclusão de que queríamos executar um jogo de plataformas com diferentes níveis e estes mesmo níveis seriam intercalados por mini bosses que, só após a derrota do mesmo, seria possível passar para o próximo. Após a apresentação desta ideia aos docentes, os mesmo aconselharam a fazer apenas dois níveis. Decidimos assim seguir o conselho dos professores e realizamos um jogo que possuía esses mesmos dois níveis de plataforma onde no final do segundo existiria dois bosses “finais”

3.6.2 Kaboom

Após algumas tentativas de posicionar as plataformas e concluirmos a dificuldade que estas mesmas davam tentamos usar uma *framework* denominada por *Kaboom* pois esta oferecia-nos ferramentas que facilitava esse processo, assim como também existia variados tutoriais por onde nos podíamos guiar. Após a estruturação das plataformas, assim como, a movimentação da nossa personagem apercebemo-nos que não seria possível inserir as *parallax* já antes construídas assim como outros elementos. Por estas razões abandonamos esta ideia e optamos por realizar o projeto apenas com *JavaScript* sem recurso a uma *framework*.

4. Resultados

4.1 Discussão e análise crítica

Com a conclusão deste projeto, consideramos o sucesso da realização do mesmo e sentimo-nos bastante satisfeitos com o resultado. Enfrentamos vários obstáculos e problemas e conseguimos ultrapassá-los todos, uns com mais dificuldade do que outros, porém foram ultrapassados com sucesso. Tiramos assim um balanço bastante positivo deste projeto pois pusemos em prática matéria lecionada em aula bem como a aquisição de novos conhecimentos em *JavaScript*.

Além disso, consideramos também que a nível concetual fomos bem-sucedidos conseguindo corresponder às expectativas iniciais de sensibilização para a subida do nível médio das águas do mar devido à erosão costeira.

4.2 Trabalho futuro

De facto, a escassez de tempo foi uma limitação para o desenvolvimento deste projeto, pelo que, pusemos de parte algumas ideias inicialmente pensadas. Assim sendo, futuramente, com mais tempo, estas ideias abandonadas (abordadas anteriormente) poderiam ser mais facilmente concretizadas. Além disso, gostaríamos também de tornar

o jogo e os seus elementos responsivos para vários tipos e tamanhos de ecrã.

5. Bibliografia

freeCodeCamp.org. “Create a Platformer Game with JavaScript - Full Tutorial.” YouTube, 21 Nov. 2018, www.youtube.com/watch?v=w-OKdSHRlFA

“JavaScript Arrays.” *W3schools.com*, 2019, www.w3schools.com/js/js_arrays.asp

Make Your Own TileMap with Vanilla Javascript. medium.com/geekculture/make-your-own-tile-map-with-vanilla-javascript-a627de67b7d9

“PDF’s Das Aulas.” *idp.ua.pt* elearning.ua.pt/course/view.php?id=8227

“Simple Platform Game in JavaScript with Source Code.” www.youtube.com/watch?v=8DYbFEBnJRs.