ESCOLA SUPERIOR DE GESTÃO E TECNOLOGIA

Licenciatura de Informática

RELATÓRIO DE TRABALHO

Jogo 2D – *Zombie Run*

Cadeira:

Aplicações Multimédia

Docente:

Cláudio Barradas

Grupo:

David Santos, 150173003

João Lopes, 160173021

Vasile Timotin, 160153015

SANTARÉM, 2019

# Introdução

Este relatório tem como objetivo descrever os vários aspetos do desenvolvimento de um jogo para a *web*. Este trabalho destina-se a implementar conteúdos lecionados em sala de aula, na forma de um jogo 2d. Os conteúdos baseiam-se em saber manusear JavaScript (DOM), HTML clássico e CSS base.

O jogo terá como tema o jogador desempenhar um papel de herói, tendo um mapa/labirinto à sua disposição, com a característica de possuir obstáculos tais como: inimigos (bonecos com movimentos aleatórios, com o objetivo de apanhar o herói), paredes de vidro, portas com chaves específicas, entre outros.

A principal tarefa do jogador é passar todos os níveis sem morrer para os inimigos.

# Principais Metas

* Criação de um ou mais mapas de jogo;
* Movimentação correta do herói, tendo em conta paredes e obstáculos do mapa;
* Movimentação aleatória de inimigos;
* Fácil interação com as características do jogo (apanhar chaves, abrir portas, avançar de nível, etc.).

# Características

* As *quests* ou pontos de interrogação, mostram ao jogador as chaves necessárias para abrir os 2 tipos de portas;
* A chave de osso só abre portas de pedra.
* A chave de ouro só abre portas de madeira;
* As escadas para baixo servem para o jogador se transportar para outras escadas para iguais;
* As escadas para cima passam de nível;
* Só se passa de nível se todas as portas estiverem abertas;
* A principal tecla para as ações é a ‘Barra de Espaço’: Partir vidro, apanhar *quests*, abrir portas, transportar entre escadas e passar de nível;
* O botão “Silence” desliga a música ambiente;
* Irá aparecendo informações ao longo do jogo, no formato de *alerts* ou no painel de informações que mostram ao jogador o que fazer, ou quantas chaves possui.

# Estratégias Usadas

Optámos por utilizar como base do mapa, obstáculos e personagens um só *Array*. Tanto o herói como os inimigos, possuem um *Array* 3x3, que avalia a posição dos mesmos, e das outras 8 posições à sua volta. Estes só passam para uma das posições à sua volta se assim conseguirem (se for uma parede não podem avançar, entre outras restrições).

Para a construção de paredes usamos um método (constructMaps) que identifica os pontos ou seguimento de pontos no mapa, como qual o mapa em questão, e implementa o caractere desejado (chão, parede, chave, etc.). Este mapa é modificado consoante a passagem de nível. Esta é condicionada por um método (nextLevel), que progride o valor da variável *mapNumber*:

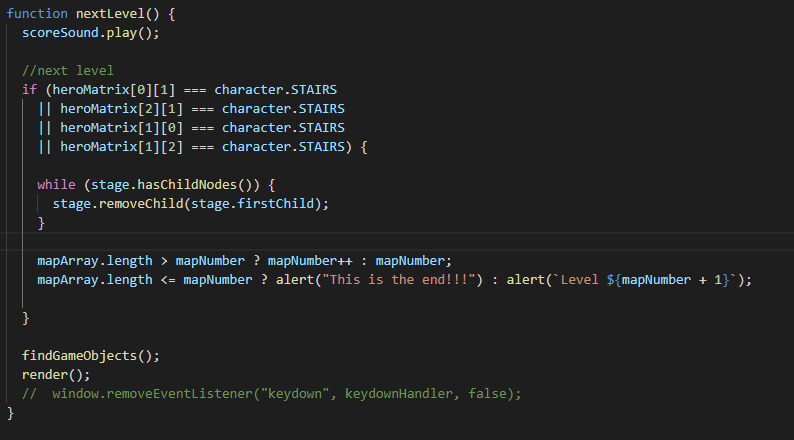


Figura - Passagem de Nível

Existe um método (render) que inicia e atualiza dos vários blocos no mapa, assim como as várias animações que alguns objetos têm.

A movimentação aleatória dos inimigos é feita com diferentes métodos. Existe uma variável, cujo valor é dado aleatoriamente. Depois do valor obtido, esta variável servirá como condição num *switch*, que executa métodos diferentes com movimentações diferentes:

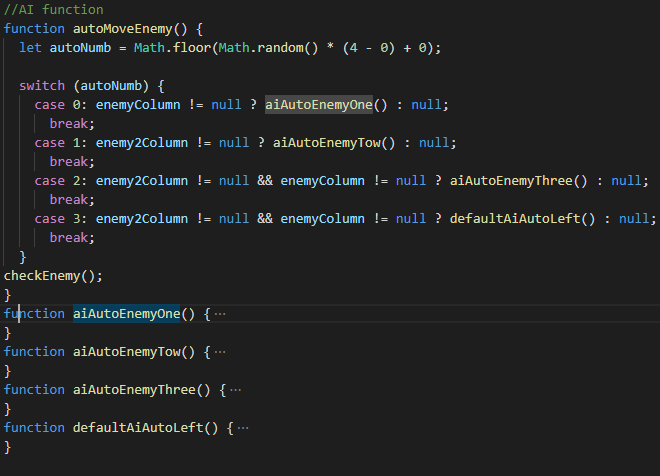
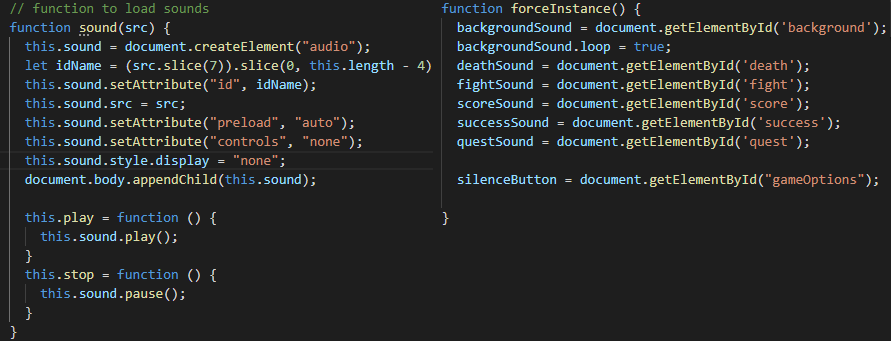


Figura - Movimentação Aleatória

A atribuição de sons é feita à volta de variáveis globais que mais tarde perante o progresso do jogo são reproduzidas:



# Notas

O tópico “Principais Metas” reflete o que o grupo se focou mais em concretizar no geral e não o que o Docente propôs na Descrição do Trabalho, embora estes dois se cruzem inevitavelmente.

O tópico “Estratégias usadas” apenas descreve alguns dos principais métodos desenvolvidos em *JavaScript*.

O jogo encontra-se com o principal idioma em Inglês, no âmbito de uma maior extensibilidade de publico alvo.

# Considerações Finais

Neste trabalho o grupo conseguiu desenvolver as suas competências nas linguagens *JavaScript*, *HTML* e *CSS*, notando um progresso no entendimento destes ao longo do projeto. Este progresso sentiu-se tanto em termos de proposição de diferentes maneiras de desenvolver algoritmos, como na correção e determinação de possíveis erros de compilação ou de má programação. Ainda assim sentimos que temos muito a aprender.

Temos consciência que poderíamos ter optado por usar estratégias de desenvolvimento diferentes ou mais eficazes, mas como aplicação de conhecimentos e investigação, optamos por prosseguir ideias que, por sua vez, podem não ter sido as mais corretas.

# Anexos

1. Ficheiro JS;
2. Ficheiro HTML;
3. Ficheiro CSS;
4. Pasta com imagens do jogo;
5. Pasta com *icons* do painel de ajudas;
6. Pasta com sons do jogo.

Figura 5 – Painel de Ajudas