



Licenciatura em Engenharia Informática

Multimédia e Computação Gráfica 2023/2024

Trabalho Prático nº 3

PyGame

Realizado em Dezembro de 2023

Afonso Fonseca nº 2020117512
André Sequeira nº 2022123691

Índice

Conteúdo

| | |
|---------------------------------------|----|
| 1. Introdução | 4 |
| 2. Métodos | 5 |
| 3. Resultados..... | 6 |
| 3.1 Menu..... | 6 |
| 3.2 Single Player..... | 7 |
| 3.3 Multiplayer | 10 |
| 3.4 Highscores | 12 |
| 4. Discussão | 13 |
| 5. Tempo Gasto com a Disciplina | 13 |
| 6. Conclusão | 14 |
| 7. Referencias | 15 |

Índice De Figuras

| | |
|--------------------------------|----|
| Figura 1 - Whack A Mole..... | 4 |
| Figura 2 - Menu | 6 |
| Figura 3 - Tutorial 1 | 7 |
| Figura 4 - Single Player | 8 |
| Figura 5 - Background..... | 8 |
| Figura 6 - Inimigo..... | 8 |
| Figura 7 - Máscara..... | 8 |
| Figura 8 - Sprites | 9 |
| Figura 9 - Gameover | 9 |
| Figura 10 - Tutorial 2 | 10 |
| Figura 11 - Tutorial 3 | 10 |
| Figura 12 - Multiplayer | 10 |
| Figura 13 - Gameover..... | 11 |
| Figura 14 - Highscores | 12 |

1. Introdução

Serve o presente documento para fazer a síntese do terceiro trabalho prático na unidade curricular de Multimédia e Computação Gráfica cujo objetivo se prende à realização de um jogo 2D ou uma animação em 3D.

Para o nosso projeto optamos por desenvolver um jogo 2D com recurso à biblioteca PyGame.

O nosso jogo teve como inspiração os jogos de arcade “*Whac-a-mole*” (fig. 1) em que o jogador tem um martelo e tem de bater na cabeça de toupeiras que vão subindo dos vários buracos da máquina ao longo do tempo, ganhando pontos por cada uma em que acerta. A nossa interpretação do jogo baseia-se não em toupeiras, mas sim num personagem da franquia *Pokemon*, Diglett. O nosso jogo permite acesso a um menu que permite aceder a um modo de 1 jogador, 2 jogadores e a uma tabela que mostre as melhores pontuações, todas estas funcionalidades irão ser melhor explicadas mais à frente.



Figura 1 - Whack A Mole

2. Métodos

Este projeto foi realizado na linguagem de programação Python com recurso à biblioteca PyGame. Esta biblioteca permite-nos criar e renderizar objetos num espaço 2D, fazer também a deteção de colisões entre objetos, deteção de teclas pressionadas, importe de áudio, permite ainda importar imagens e atribuir estas a retângulos. Estes retângulos que podem ser posteriormente modificados para cumprir os objetivos pretendidos através de translações e rotações. Podem ainda ser realizadas várias operações através da deteção de colisões entre objetos ou ainda por exemplo da deteção do rato quando é pressionado quando em contacto com um objeto.

Utilizamos também a biblioteca *pandas*, que foi trabalhada em aula, na implementação do sistema de pontuações, no qual os dados são exportados para um ficheiro csv. Isto, pois, o jogo tem de manter registo dos oito melhores scores a qualquer momento para poderem ser acedidos através da opção “Highscores” no menu.

Os diversos aspetos do jogo foram testados ao longo do seu desenvolvimento, mesmo assim ainda existem algumas falhas, que com mais algum trabalho poderiam eventualmente ser resolvidas. As falhas que chegaram ao trabalho foram as seguintes:

- Alguns problemas de colisões;
- Raramente o jogo não reproduz alguns sons;
- É possível ver às vezes sprites a aparecerem e a desaparecerem logo;
- Sprites trocados.

3. Resultados

3.1 Menu

Ao iniciar o jogo encontramos-nos com um menu que nos dá 3 opções:

- Single Player: Para apenas uma pessoa jogar
- Multiplayer: Permite 2 pessoas jogarem em simultâneo competindo por um score melhor.
- Highscore: Irá permitir visualizar os 8 melhores scores.

Enquanto no menu o jogo irá tocar música e o rato irá ser apresentado como um martelo.



Figura 2 - Menu

3.2 Single Player

Ao pressionar a opção “Single Player” a tela irá apresentar um pequeno tutorial no qual é explicado o que faz cada inimigo:

- Diglett: Inimigo mais comum, quando pressionado aumenta o score por +1.
- Shiny Diglett: Inimigo bastante raro (1% de chance de aparecer), ao pressionar +100 de score.
- Diglett Bomba: Este inimigo quando pressionado irá fazer um som de uma explosão e retirar 1 coração ao jogador
- Diglett Vida: Este só aparece depois do jogador perder pelo menos 1 vida e depois de ser pressionado dá um coração e faz um som apropriado.

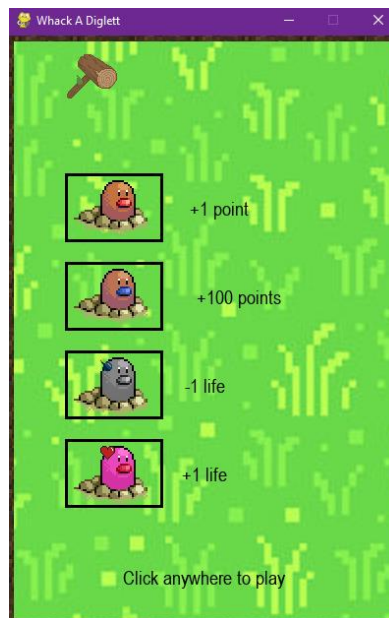


Figura 3 - Tutorial 1

Além de darem 1 ponto, os inimigos normais quando pressionados repetidamente sem que o utilizador pressione ou numa bomba ou no background, ativam um sistema de “combo” em que por cada inimigo em que o utilizador acerta recebe pontos extra, sendo que em quantos mais acertar sem falhar nenhum, mais pontos ganha. O jogo irá ficar neste menu até que a tela seja pressionada em qualquer área como indicado. A música do menu irá também continuar a tocar até que saia do tutorial.

Após sair do tutorial o jogo começa. Sendo o objetivo conseguir o máximo de pontos possíveis.

Cada um dos inimigos, quando atingido muda de imagem para representar que já foi pressionado ao mesmo tempo que reproduz um ruído próprio. O jogador perde vidas ao atingir o fundo do ecrã ou ao pressionar nas bombas, sendo que o jogo termina ao acabarem as vidas.



Figura 4 - Single Player

Os inimigos aparecem entre o nosso background e uma máscara de modo que apenas sejam visíveis após a imagem subir o suficiente para ultrapassar a máscara. Voltando novamente para baixo até estarem totalmente tapados pela máscara.



Figura 5 - Background



Figura 6 - Inimigo



Figura 7 - Máscara

A cada instante que um inimigo aparece a sua posição é gerada através da função `random`, que irá gerar um número aleatório entre 0 e 8 que irá corresponder a um conjunto de coordenadas iniciais que estão guardadas num array. Será também de forma aleatória escolhido o tipo de inimigo que irá aparecer.

A partir do momento em que é detetada uma colisão entre a imagem associada ao rato e um inimigo o jogo irá trocar o sprite do inimigo que levou hit por uma segunda versão do sprite ao mesmo tempo que faz um barulho apropriado e atribui a pontuação correta.



Figura 8 - Sprites

Quando o jogador acaba por chegar a 0 vidas, quer seja por carregar nas versões dos inimigos com bombas ou por simplesmente falhar o click no inimigo e carregar na tela. O utilizador é enviado para um ecrã de *Gameover*. Este ecrã irá mostrar a pontuação com que o utilizador acabou assim como um campo onde o utilizador poderá inserir o seu nome.

O nome de utilizador e o score associado serão posteriormente apresentados na opção “Highscores”, caso este esteja nos 8 melhores. O nome é obtido através do ficheiro de pontuações. Ao ler esse ficheiro é possível ver o ultimo jogador com a mais recente data de modificação. Ao carregar em qualquer zona da janela este ecrã irá desaparecer e o utilizador volta ao menu inicial.



Figura 9 – Gameover

3.3 Multiplayer

Ao pressionar a opção “Multiplayer” a tela irá apresentar um pequeno tutorial no qual são explicados os controlos do primeiro jogador. Ao pressionar em qualquer parte do ecrã com o rato novamente serão apresentados os controlos para o segundo.

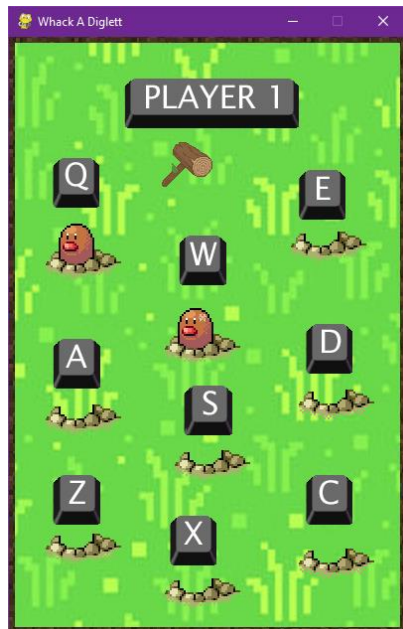


Figura 10 - Tutorial 2



Figura 11 - Tutorial 3

Ao pressionar uma terceira vez o jogo começa. Como neste modo apenas é utilizado o teclado a imagem do martelo que tem acompanhado o rato até agora vai desaparecer de modo a ocultar o rato por completo deste modo a deixar a janela de jogo mais limpa.

Neste modo a pontuação sofre uma pequena alteração, perdendo a mecânica de “combo” sendo que agora cada inimigo básico só dá apenas 1 ponto.

Como existem agora dois jogadores a jogar em simultâneo o primeiro jogador a pressionar num inimigo irá receber a pontuação e mesmo que o segundo jogador pressione o mesmo inimigo este não irá dar pontos, mas também não irá tirar um coração.

No multiplayer os jogadores só iriam poder perder corações caso, pressionem num inimigo bomba ou caso pressionem a tecla correspondente a um buraco que está vazio.

Este modo de jogo só acaba quando ambos os jogadores chegam a zero vidas, sendo que a partir do momento em que um jogador fica sem corações, este já não pode competir.



Figura 12 - Multiplayer

Assim que o último jogador vivo perde todos os corações que lhe restam, o jogo termina, sendo de seguida encaminhados para um ecrã de “Game Over” no qual são apresentados os scores de ambos os jogadores. Neste modo de jogo os scores não são guardados pois não achámos adequado ao objetivo.



Figura 13 - Gameover

3.4 Highscores

Ao pressionar a opção “Highscores” no menu, irá ser apresentado a tela de highscores que mostra os 8 jogadores com as maiores pontuações.

O jogo vai obter os resultados a partir de um ficheiro csv que é modificado cada vez que um jogador perder no modo “Single Player”. Esse ficheiro deverá conter o nome de jogador (name), a pontuação (score) e a data de modificação do score (last_modified). Para tal tarefa é usada a biblioteca pandas que consegue ler e escrever ficheiros csv e muito mais. Ao tentar escrever os highscores no ficheiro, primeiro é testado se um ficheiro com o nome “highscores.csv” existe na página de jogo e caso não exista ele vai criar um. A seguir é testado se já existe algum jogador com o nome especificado na tela de gameover. Caso não exista, irá ser adicionado uma linha com o nome, a pontuação e a data no formato unix timestamp. Caso exista irá ser verificado se o novo score é maior que o anterior e se tal acontecer é substituído o novo score. No final o campo de data de modificação irá ser sempre alterado independentemente se o jogador tem ou não uma maior pontuação.



| HIGHSCORES | |
|------------|-----|
| Mario | 205 |
| André | 65 |
| Jose | 51 |
| Ana | 31 |
| Isa | 21 |
| Maria | 20 |
| Pedro | 5 |
| Armando | 4 |

Figura 14 - Highscores

4. Discussão

Este trabalho pratico tem como base o que foi dado nas aulas de Multimédia e Computação Gráfica. Neste trabalho não só foi utilizado o que aprendemos em relação à biblioteca PyGame mas também aplicamos alguns conceitos da terceira matéria dada (Ciência de Dados) com o objetivo de organizar os scores obtidos pelos utilizadores, para isto utilizamos a biblioteca Pandas.

Os resultados obtidos foram como esperados, contudo, houve alguns problemas e ainda existem alguns erros, raramente pode haver erros na deteção da colisão podendo o utilizador perder uma vida quando não devia, raramente os inimigos podem não fazer nenhum som quando atingidos, contudo, diríamos que o no contexto do que era pedido e do tempo que dedicamos ao projeto este atingiu as nossas expectativas, deixando-nos par à prova o que foi aprendido nas aulas e também explorar novos conceitos.

Uma última nota foi o facto de não termos conseguido criar um executável para correr o nosso jogo com recurso ao PyInstaller. O erro com que nos deparamos parece ser algum problema relacionado com o método como organizamos as imagens e achamos que com um pouco mais de tempo teríamos resolvido este problema.

Neste momento para correr o jogo basta executar o ficheiro “main.py”.

5. Tempo Gasto com a Disciplina

| Aluno | Nº Total de aulas | Nº Presenças | Nº de Horas | Nº de Horas Presentes |
|--------|-------------------|--------------|-------------|-----------------------|
| Afonso | 24 | 21 | 2-3h | 52h |
| André | 24 | 20 | 2-3h | 50h |

Afonso Fonseca:

Tempo gasto com a disciplina em aula por semana: 5h.

Tempo gasto com a disciplina extra-aula por semana: 2-3h

Tempo gasto no desenvolvimento do Trabalho Prático nº3: 32h

André Sequeira:

Tempo gasto com a disciplina em aula por semana: 5h.

Tempo gasto com a disciplina extra-aula por semana: 2-3h

Tempo gasto no desenvolvimento do Trabalho Prático nº3: 30h

6. Conclusão

Com o fim deste projeto podemos dizer que atingimos os requisitos propostos tanto no enunciado da avaliação periódica como da avaliação final. Tendo não só feito um jogo funcional com recurso ao PyGame como também ter implementado um menu, multiplayer e uma maneira de aceder aos 8 maiores scores obtidos.

Mesmo não tendo sido um trabalho fácil achamos que conseguimos por as nossas capacidades à prova e fizemos o nosso melhor, tentando implementar no trabalho outras técnicas aprendidas anteriormente como foi o caso da biblioteca Pandas que foi usada para a parte do trabalho relativa aos highscores juntamente com a biblioteca PyGame.

Por fim achamos o tema deste trabalho algo divertido e interessante no contexto da cadeira de Multimédia e Computação Gráfica pois nos deixa aplicar não só aquilo que é esperado como também os outros tópicos abordados na cadeira.

7. Referencias

PyGame (no date) *pygame* Available at: <https://www.pygame.org/> (Accessed: 10 December 2023).

Pandas (no date) *pandas*. Available at: <https://pandas.pydata.org/> (Accessed: 10 December 2023).

Pyinstaller (no date) *PyInstaller Manual - PyInstaller 6.2.0 documentation*. Available at: <https://pyinstaller.org/> (Accessed: 10 December 2023).