Alunos: Paulo Rodrigues, nº 83929; Rodrigo Linhas, nº 83933; Tiago Tomé, nº83862
Grupo: P2G2
Programação orientada a objetos.

Projeto de POO

Burger-Man

Universidade do Algarve – Faculdade de ciências e tecnologia
2º Ano de Engenharia informática
2024/25

Índice

O jogo:

O nosso jogo consta numa copia do pac-man, jogo mundialmente conhecido.

No nosso jogo, o personagem controlável pelo utilizador é um hambúrguer, os inimigos são baldes do lixo.

Como já foi referido, o utilizador pode controlar o personagem principal, com o teclado (w,a,s,d) que pode apanhar pelo mapa os pontos (as bolinhas típicas de pontos do pac-man) e power-ups, sendo estes power-ups um *sprite* estático que podem ser:

- tomates que dão invencibilidade durante 8 segundos;
- cebolas que eliminaria um inimigo de forma instantânea;
- queijo que tem uma probabilidade de te deixar mais rápido ou diminuir a velocidade.

Já os inimigos também podem apanhar um power-up, no caso um pickle (já que a maioria do pessoal não gosta muito de pickles nos hambúrgueres) que vai abrandar o jogador em 80% a velocidade atual do jogador. Caso o jogador apanhe um pickle não acontece nada com o jogador, apenas elimina o pickle do mapa de forma aos inimigos não o poderem apanhar.

O mapa tem um labirinto parecido no pac-man original, a única alteração é o corredor que sai do ecrã que dá de um lado para o outro. As paredes que fazem o labirinto são de cor castanha e o fundo do jogo é preto para dar maior contraste de todos os elementos presentes.

O jogo contém também vidas e pontuação. Como é lógico as vidas descem sempre que algum balde de lixo conseguir apanhar o hambúrguer e a pontuação aumenta á medida que se apanha as bolinhas de pontos já explicadas anteriormente.

Contem apenas 1 nível para facilitar o processo de criação.

Padrões de projeto:

O nosso jogo utilizou os seguintes padrões de projeto para garantir uma arquitetura flexível, organizada e de fácil manutenção:

1. Padrão Observer

Este será o nosso padrão de desenho também. A dinâmica será muito simples, o jogador emitirá notificações sempre que colidir com uma parede/inimigo e quando apanhar pontos/power-ups.
Estas notificações serão obtidas pelos observadores (***behaviours***) das respetivas classes. Como por exemplo:

- O jogador apanhou um ponto, será emitido ao sistema de pontuação essa notificação e será incrementado 10 na pontuação total.
- Outro exemplo seria a colisão com um pickle, onde existe uma decisão: foi emitido uma notificação em quem colidiu foi o jogador? Se sim não acontece nada, se não só pode ter sido o inimigo e consequente a velocidade do jogador abranda.

São exemplos básicos, mas dá para ter uma noção de como o nosso jogo irá se comportar.

2. Padrão Strategy

O padrão Strategy permite comportamentos dinâmicos para os inimigos e o jogador, usado principalmente no *IBehauvior*.
Exemplo: (EnemyBehaviour) Inimigos podem alternar entre patrulha e perseguição sem alterar a sua classe.

3. Padrão Factory Method

O padrão Factory Method simplifica a criação de objetos e facilita a manutenção do código.
Exemplo: (Transform)*backgroundTransform()* cria uma transformação básica para o objeto *background* que tem o intuito de ser usado para representar o mapa.

4. Padrão State

O padrão State gerência estados do jogador ao consumir os power-ups, temos por exemplo a invencibilidade, isolando a lógica de cada estado, evitando condições complexas no código do jogador.
Exemplo: (PlayerBehaviour) quando o jogador colide com o tomate ativa InvincibleState por 200 frames.

5. *Padrão Composite*

O padrão Composite trata grupos de objetos, temos por exemplo a lista de inimigos, coleção de pontos, como entidades únicas simplificando operações como atualização e renderização de múltiplos objetos.
Exemplo: a classe *GameEngine* gerência uma lista de *GameObject* (*loadedObjects*), permitindo várias operações (ex: *destroyAll()*).

Controle do utilizador:

O jogador controla o hambúrguer num jogo top down view, com as teclas do teclado que faz o jogador mover-se pelo mapa do jogo de forma a interagir com o ambiente do mesmo. Os pontos servem para ganhar o nível e os power-ups enquanto tenta evitar os inimigos, sendo estes os baldes de lixo.

Básicos da jogabilidade:

Durante o jogo aparecem os inimigos, os baldes de lixo, no meio do mapa. O objetivo do jogo é apanhar todos os pontos do nível enquanto se desvia de inimigos e apanha power-ups que dão vantagens no jogo.

Sons e efeitos:

O jogo terá uma música ambiente, esta para quando o jogador morre, i.e quando é apanhado por um balde de lixo. Vai existir efeitos quando se apanhar power-ups e quando houver a colisão de morte do jogador.

Mecânicas do jogo:

À medida que o jogo avança os baldes de lixo vão sendo libertados para irem atrás do hambúrguer fazendo com que seja mais difícil para o jogador apanhar os pontos. Também haverá power-ups que os baldes de lixo podem apanhar de forma a dar desvantagem ao jogador. Também vai existir power-ups para o jogador fazendo que o jogador ganhe vantagens sobre os inimigos.

Interface de utilizador:

Os pontos vão aparecer no ecrã e vão aumentar sempre que o jogador apanhe as bolinhas de pontos.

As vidas vão aparecer no ecrã e vão diminuir sempre que um inimigo conseguir atacar o jogador.

No início do jogo **não** vai aparecer um ecrã com o nome do jogo, nem vai existir nenhum menu inicial, o jogo apenas vai começar diretamente no nível e caso o player perca todas as suas vidas é apresentada uma mensagem de “Game Over” e o player não consegue mais fazer nada.