
Projeto de POO

Burger-Man

Índice

A ideia geral do jogo:	2
Padrões de projeto:.....	3
1. Padrão Observer	3
2. Padrão Strategy	3
3. Padrão Factory Method	3
4. Padrão State	3
5. Padrão Composite	4
Controle do utilizador:	4
Básicos da jogabilidade:	4
Sons e efeitos:	4
Mecânicas do jogo:	4
Interface de utilizador:	5

A ideia geral do jogo:

A nossa ideia consta numa imitação barata do pac-man, jogo mundialmente conhecido e em princípio com implementações básicas. No nosso jogo, o personagem controlável pelo utilizador é um hambúrguer, os inimigos são baldes e caixotes do lixo.

Como já referi, 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 com a mesma velocidade.

Já os inimigos também podem apanhar um power-up, no caso um pickle (já que a maioria do pessoal gosta muito de pickles nos hambúrgueres) que vai abrandar o player em 75 por cento. Caso o player apanhe um pickle não acontece nada com o player, apenas elimina o pickle do mapa de forma aos inimigos não o poderem apanhar.

O mapa deve ter o labirinto como no pac-man original, sendo que as paredes que fazem o labirinto devem utilizar um design com azulejos com cores como nas cadeias de fast food (cores vivas e fortes com padrões). Já o ambiente do jogo deve entrar nesse mesmo design com o chão com azulejos pretos e brancos.

Deve ter vidas e pontuação. Como é lógica as vidas descem sempre que algum balde ou caixote do lixo conseguir apanhar o hamburger e a pontuação aumenta á medida que se apanha as bolinhas de pontos já explicadas anteriormente.

Pensamos em fazer apenas 1 nível para facilitar o processo de criação do jogo.

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 player emitirá notificações sempre que colidir com uma parede/inimigo e quando coletar pontos/power-ups.

Estas notificações serão obtidas pelos observadores das respetivas classes.

Como por exemplo:

- O player coletou um ponto, será emitido ao sistema de pontuação essa notificação e será incrementado 10 na pontuação total.
- Outro exemplo seria a colisão com um inimigo, aonde existe uma decisão: foi emitido uma notificação em que o player contem invencibilidade e ainda esta valida? Se sim o inimigo morre, se não o player morre.

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 player, usado principalmente no *IBehavior*.

Exemplo: 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.simpleTransform(Point position)* cria uma transformação básica para objetos estáticos que pode ser usado para os pontos no mapa.

4. Padrão State

O padrão State gerencia estados do player ao consumir os powerups, temos por exemplo a invencibilidade, isolando a lógica de cada estado, evitando condições complexas no código do player.

Exemplo: quando o player colide com o tomate ativa *InvincibleState* por 8 segundos

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* gerencia uma lista de *GameObject* (*loadedObjects*), permitindo várias operações (ex: *destroyAll()*).

Controle do utilizador:

O player controla o hamburger num jogo top down view, com as teclas do teclado que faz o player 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 e caixotes do lixo.

Básicos da jogabilidade:

Durante o jogo aparecem os inimigos, os baldes e caixotes do lixo, no meio do mapa. O objeto 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 e consequentemente mudará quando o player morrer e/ou apanhar um power-ups. Vai existir efeitos quando se apanhar power-ups, quando matar inimigos e quando houver a colisão de morte do player.

Mecânicas do jogo:

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

Interface de utilizador:

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

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

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.