

ESCOLA SUPERIOR DE TECNOLOGIA - IPS
ANO LETIVO 2015 / 2016

ENGENHARIA DE INFORMÁTICA

PROGRAMAÇÃO AVANÇADA. PROF^a PATRICIA MACEDO

MILLESTONE 3

MIGUEL FURTADO 120221006
TIAGO MONTEIRO 140221001

Índice

Padrão Iterator.....	2
Padrão Memento.....	2
Padrão Strategy.....	2
Padrão Simple Factory.....	2
Padrão Singleton.....	2
Padrão Observer.....	3
Padrão Adapter.....	3
Padrão Abstract Factory.....	3
Padrão MVC.....	3

Padrão Iterator

O Padrão Iterator foi usado para percorrer as linhas, nomeadamente na classe linha jogo em que o padrão é usado para a verificação das peças que existem à direita e à esquerda, para desenhar as linhas e também para fazer uma deep copy das linhas.

Padrão Memento

O Padrão Memento foi usado para o método undo e para tal criámos a class Memento que vai guardar a informação do tabuleiro, a informação das peças e as jogadas. A classe JogoCareTaker onde há um atributo stack que guarda os Mementos para que quando seja feito undo possamos ter a informação de como era o jogo antes da última jogada. Na classe jogoConsola criámos o atributo careTaker do tipo JogoCareTaker que foi inicializado no construtor e posteriormente no método jogar grava os diferentes estados posteriormente a cada jogada e caso seja chamado o método undo faz um restore do último estado do jogo gravado.

Padrão Strategy

O Padrão Strategy foi usado para definir o tipo de pontuação que iria ser usado nos diferentes jogos, normal ou rápido. Para tal criámos uma classe EstrategiaTipoPontuação que é uma interface e tem o método pontuação do tipo int e que depois vai ser implementada nas classes PontuaçãoBase e PontuaçãoCorrida. Na classe PontuaçãoCorrida a pontuação é inicializada a 1000 e a cada 10 segundos que passa o utilizador perde 1 ponto e por cada peça destruída ganha 10 pontos. Na classe PontuaçãoBase o utilizador apenas ganha pontos por cada peça destruída que vai equivaler a 20 pontos por cada uma que seja destruída.

Este padrão depois é inicializado sempre que é inicializado o jogo consoante o utilizador tenha escolhido se quer o modo corrida ou base.

Padrão Simple Factory

O Padrão simple factory foi usado para a criação de peças. Dado que podem existir vários temas para o jogo, temos que definir várias peças para os diferentes temas, tendo isso em conta criámos a classe PecaFactory que é uma interface e que nesta fase estende para as classes FactoryHalloween, FactoryPokemon, FactorySmileys, FactorySouthParkn e FactoryWar.

Padrão Singleton

O padrão singleton foi usado para a criação de rankings. Como tal foi criado a classe Rankings que só tem uma única instância e nesta classe foi criado um ponto de acesso global a essa instância. Assim garantimos que apenas um objeto exista, independentemente do número de requisições que receber para criá-lo. Com este padrão implementado pode ser usado na

criação do jogo para criar os diferentes rankings que existirão no jogo com base no tipo de jogo e no tipo de pontuação.

Padrão Observer

O padrão observer foi usado para o desenvolvimento do logger. Para criamos algumas classes para as diferentes coisas que havia necessidade de serem observadas. O início do jogo que vai ser guardado e registado no logger, as jogadas que vão ser observadas e guardadas no logger, a ação de fazer undo também vai ser guardada no logger sempre que o observer registar uma ação e por fim o fim do jogo também irá ser guardado no logger.

Padrão Adapter

O padrão adapter foi usado no desenvolvimento da classe jogoRanking, usámos este padrão por já termos algo desenvolvido previamente então agora apenas seria necessário usar alguns métodos, neste caso usámos alguns métodos da classe TadRanking.

Padrão Abstract Factory

O padrão Abstract Factory foi usado para a implementação do tema em que a partir deste se cria os diferentes temas e se devolve um peça factory que vai ser usado na classe gamesSceneControler para facilitar a criação das peças a partir de um tema. Para tal foram criadas as classes AbstractFactory que apenas tem o método getTema e a class TemaFactory que implementa a classe AbstractFactory e nesta são criados os temas.

Padrão MVC

O padrão MVC que significa Model-View-Controller, foi usado para dividir todo o projeto. No model temos toda a parte lógica que foi implementada na 1ª, 2ª e 3ª fase. Na view vai ter tudo o que está associado à parte gráfica que foi implementado na 3ª fase. O controler é a junção onde podemos ver a parte gráfica a trabalhar em conjunto com a lógica.