



Instituto Superior de Engenharia de Coimbra
Cadeira de POO
Ano Lectivo 2020/2021

TRABALHO PRÁTICO DE POO

Meta 2
P8

Diogo Oliveira Braz Monteiro - 2017016078
Diogo Miguelo Semião - 2017019992

1. Quais foram as classes consideradas na primeira versão da aplicação que foi testada?

A primeira versão tinha as seguintes classes e principais funcionalidades:

- Território
 - Criação de todos os territórios do jogo
 - Obter a descrição textual de todos os territórios
- Jogador
 - Armazenar todas as estatísticas do utilizador que está a jogar o jogo
 - Obter a descrição textual de todas as estatísticas do jogador
- Mundo
 - Ler os comandos inseridos pelo utilizador
 - Lista toda a informação necessária ao jogo
 - Carrega comandos de ficheiros
 - Controla o fluxo do jogo
 - Insere territórios no mundo do jogo

Estas classes e funcionalidades foram testadas com uma simples função main.

2. Quais os conceitos/classe que identificou ao ler o enunciado?

Os conceitos identificados foram:

- Jogador
- Territórios
- Continente
- Ilha
- Duna
- Montanha
- Castelo
- Fortaleza
- Pescaria

- Refúgio dos Piratas
- Planície
- Mina
- Mundo

3. Relativamente a duas das principais classes da aplicação, identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objectos.

Território: Os objetos da classe território são criados e destruídos na classe Território e são armazenados no Mundo.

Jogador: Os objetos desta classe são criados e destruídos na classe Jogador e armazenados no Mundo.

4. Indique um exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento.

A responsabilidade de “conquistar territórios” está atribuída à classe Mundo porque tem a colecção de territórios.

5. De entre as classes que fez, escolha duas e justifique por que considera que são classes com objetivo focado, coeso e sem dispersão.

Classe Território: tem dados e responsabilidades relativos apenas aos territórios, como por exemplo obter resistência dos mesmos, ouro etc..

Classe Jogador: tem dados e responsabilidades relativas apenas ao jogador e daquilo que a ele está associado, como territórios onde apenas procura saber as estatísticas desses territórios.

6. Relativamente à aplicação entregue, quais as classes que considera com responsabilidades de interface com o utilizador e quais as que representam a lógica?

Responsabilidade de interface: Mundo

Responsabilidades da lógica da aplicação: Jogador, Território, Save, Tecnologias

7. Identifique o primeiro objecto para além da camada de interacção com o utilizador que recebe e coordena uma funcionalidade de natureza lógica?

As ordens vindas da camada de interação com o utilizador são recebidas e processadas por um objecto da classe Território.

8. A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes? Indique um exemplo em que esta classe delega uma funcionalidade noutra classe.

A classe Mundo representa a envolvente de toda a lógica. Para adicionar uma nova conquista é delegado à classe Jogador para adicionar ao seu vetor um novo território.

9. Dê um exemplo de uma funcionalidade que varia conforme o tipo do objecto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade.(Não é necessário responder a esta pergunta na meta 1).

O método da classe Território chamado skill, é uma funcionalidade que varia mediante a classe derivada, de território, criada. Cada território tem a sua “habilidade” que varia mediante o turno, o ano quando é conquistado, etc..

```
virtual void skill(int ano, int turno);
```

Esta função virtual está definida na classe Território e é reescrita nas suas derivadas. Exemplo:

Planície.h:

```
void skill(int ano, int turno)override;
```

Planície.cpp:

```
void Planície::skill(int ano, int turno)
{
    if (ano == 1) {
        prod = 1;
    }
    if (ano == 2) {
        prod = 2;
    }
}
```

A quando da chamada da função skill em que o objeto construído seja uma planície esta irá realizar a função reescrita nela. Este comportamento acontece para todos os territórios que tenham alguma “habilidade”.

10. Apresente as principais classes da aplicação através da seguinte informação:

Classe: Território

Responsabilidades:

- Criação de todos os territórios do jogo

- Obter a descrição textual de todos os territórios

Colaborações: Jogador, Mundo

Classe: Jogador

Responsabilidades:

Armazenar todas as estatísticas do utilizador que está a jogar o jogo

Obter a descrição textual de todas as estatísticas do jogador

Colaborações: Mundo, Território

Classe: Mundo

Responsabilidades:

Ler os comandos inseridos pelo utilizador

Lista toda a informação necessária ao jogo

Carrega comandos de ficheiros

Controla o fluxo do jogo

Insere territórios no mundo do jogo

Colaborações: Jogador, Território, Save

Funcionalidades implementadas

Componente do trabalho	Realizado	Realizado Parcialmente	Não Realizado
Comando cria	X		
Comando lista	X		
Comando carrega	X		
Comando conquista	X		
Todos os comandos pedidos no enunciado	X		