Instituto Superior de Engenharia de

Coimbra



Departamento de Engenharia Informática e de Sistemas

Engenharia Informática

Programação Orientada a Objetos 2019/2020

Simulador Corridas de Carros

C++

Trabalho Prático - Checkpoint 1

Diogo Marques- 21240125

Miguel Abrantes – 21240373

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

A primeira versão tinha as seguintes classes:

* + - Carro
    - Piloto

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

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

Os conceitos identificados foram:

* Corrida
* Carro
* Piloto (O conceito de polimorfismo devido à existência de vários Pilotos – Não implementado)
* Autódromo
* Pista
* Garagem
* Campeonato //Não sei se corrida e campeonato poderão ser os mesmos xD
* DVG (Classe abstrata que contém todos os tipos de objetos do simulador).

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

Autódromo: contém os objetos Pista e Garagem que são criados, assim como destruídos pela mesma.

Equipa: contém os Pilotos e Carros que estejam a agir como uma equipa, ou seja, que o piloto esteja no carro. Ambos os seus objetos são criados e eliminados na classe DVG, independentemente da classe Equipa.

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

A responsabilidade “criar equipa” está atribuída à classe DVG, porque tem a acesso a todos os pilotos e carros

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

* Classe Piloto: tem dados e responsabilidades relativos apenas aos pilotos, como saber se está dentro de um carro ou não.
* Classe Carro: tem dados e responsabilidades relativos apenas aos carros, como saber se o carro está em movimento ou não.

1. **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: Interface
* Responsabilidade da lógica da aplicação: Autodromo, Pista, Garagem, Carro, Piloto, Jogo e DVG.

1. **Identifique o primeiro objeto para além da camada de iteração 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 objeto da classe Jogo.

1. **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 Jogo representa a envolvente de toda a lógica. É a classe que vai ser utilizada por exemplo para ler os ficheiros e também delega funcionalidades noutras classes, como por exemplo para obter a listagem de todas as entidades existentes, esta vai “pedir” à classe DVG.

1. **Dê um exemplo de uma funcionalidade que varia conforme o tipo do objeto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade. (*Não é necessário responder a esta pergunta na meta 1*). //Vale a pena responder ?**
2. **Apresente as principais classes da aplicação através da seguinte informação:**

**Classe:** Piloto

**Responsabilidades:**

Guardar todos os pilotos.

Obter nome e tipo de um piloto.

Verificar se está associado a um carro.

Parar e iniciar o movimento do carro que o piloto estiver a conduzir.

**Colaborações:** Carro

**Classe:** Carro

**Responsabilidades:**

Atribuir ID automático a um carro novo.

Obter informações sobre o carro (Vel. máxima, cap. da bateria, etc.).

**Colaborações:**

**Classe:** Pista

**Responsabilidades:**

Remover Equipas da pista

**Colaborações:** Equipa

**Classe:** Garagem

**Responsabilidades:**

**Colaborações:**

**Classe:** Autodromo

**Responsabilidades:**

**Colaborações:**

**Classe:** Interface

**Responsabilidades:**

Recebe e interpreta comandos do utilizador

**Colaborações:** Jogo

**Classe:** Campeonato

**Responsabilidades:**

**Colaborações:**

**Classe:** Jogo

**Responsabilidades:**

Responsável por ler ficheiros de texto com informação a carregar

Iniciar novos campeonatos

Iniciar uma corrida

**Colaborações:** DVG, Autodromo, Campeonato

**Classe:** DVG

**Responsabilidades:**

**Colaborações:**

**Classe:** Equipa

**Responsabilidades:**

**Colaborações:**

**Funcionalidades Implementadas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente do Trabalho** | **Realizado** | **Realizado Parcialmente** | **Não Realizado** |
| **Modo 1** |  |  |  |
| Execução de todos os comandos (exceto dgv) | **x** |  |  |
| Cria-se 1 autódromo | **x** |  |  |
| Cria-se um tipo de piloto | **x** |  |  |
| **Modo 2** |  |  |  |
| Comando “campeonato” | **X** |  |  |
| Comando “passatempo” | **X** |  |  |
| Carros moverem-se | **x** |  |  |