 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo
	Segundo momento de avaliação - AC	2017/2018
	Curso	LEI – Licenciatura em Engenharia Informática LSIRC - Licenciatura em Segurança Informática em Redes de Computadores
	Unidade Curricular	Paradigmas da Programação

## Destinatários

Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de Paradigmas de Programação que pretendam obter aprovação à Unidade Curricular.

## Objetivos

Com a realização do trabalho prático, pretende-se que os alunos ponham em prática todos os conhecimentos adquiridos na utilização do paradigma de programação orientado a objetos e a sua implementação na linguagem de programação Java, demonstrando as suas apetências em:

- Conhecer e compreender os conceitos fundamentais associados ao paradigma da programação orientada a objetos;
- Conceber e implementar, para problemas concretos, soluções que tenham por base o paradigma da programação orientada a objetos.
- Reconhecer e compreender a semântica e a sintaxe da linguagem Java.
- Reutilizar, alterar e desenvolver código recorrendo à linguagem Java tendo em vista um determinado problema com regras semânticas específicas.

Considere ainda que:


- Não é permitida a utilização de API's/conceitos Java que não tenham sido alvo de lecionação - ano letivo 2017-2018 - da unidade curricular Paradigmas de Programação. Os alunos que pretendam utilizar API's adicionais devem atempadamente pedir autorização a um dos docentes da unidade curricular.
- Não é permitida a utilização de coleções Java predefinidas (*Java Collections Framework*).
- Os recursos de suporte ao trabalho referenciados no enunciado, são de utilização **obrigatória**.

## Enunciado

Uma empresa de produção de videojogos da região do vale do souse, a *ARGA* (Advanced Research Game Agency), pretende construir um videojogo para cimentar a sua posição no mercado dos videojogos. Assim decidiu lançar um desafio aos jovens programadores da região para o desenvolvimento de um pequeno protótipo de forma a averiguar a aceitação na região do vale do souse e em geral no mercado global dos videojogos. O desafio consiste no desenvolvimento de um videojogo de corridas (corridas de carros, de motos, etc.) em duas dimensões (2D), que tire partido das potencialidades da biblioteca de desenvolvimento, recentemente desenvolvida pela empresa.

Os requisitos estabelecidos pela *ARGA* são os seguintes:

- O jogo deverá ser desenvolvido em linguagem Java e utilizando os recursos disponibilizados (disponíveis na plataforma moodle);
- O jogo deverá ser constituído por várias pistas de corridas (vulgarmente chamadas de circuito de corrida ou autódromos);
- Cada pista é constituída por um número de voltas, um veículo, e vários *checkpoints* associados;
- Cada veículo possui especificações próprias de velocidade, travagem, direção e é conduzido por um piloto;
- Deverá ser possível controlar o veículo na pista em que se encontrar inserido;
- Deverá ser possível a visualização dos tempos obtidos no final da corrida para uma determinada pista;


 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo
	Segundo momento de avaliação - AC	2017/2018
	Curso	LEI – Licenciatura em Engenharia Informática LSIRC - Licenciatura em Segurança Informática em Redes de Computadores
	Unidade Curricular	Paradigmas da Programação

- Deverá ser possível o armazenamento e carregamento dos dados relativos aos tempos obtidos para uma determinada pista;
- Deverá ainda ser possível detetar colisões entre o veículo e os limites da pista (**Nota:** esta funcionalidade não necessita de implementação, já que a biblioteca fornecida deteta colisões, dado os limites definidos do veículo e da pista);
- Para além das APIs de suporte, a *ARGA* disponibiliza uma biblioteca com os contratos base (2018.MA02.AC.Resources) que devem (no mínimo) ser implementados para a realização deste trabalho.

Assim, numa primeira fase, torna-se necessária a criação de uma API em linguagem Java que permita expor as funcionalidades inerentes à gestão dos vários elementos que constituem o jogo de corridas. Deverá incluir pelo menos as seguintes (outras poderão existir) funcionalidades:

- Deverá ser possível adicionar as diversas pistas, inserir um veículo e o respetivo piloto relativo ao videojogo criado. O relacionamento entre os diversos conceitos encontra-se descrito nos recursos disponibilizados para a implementação da API;
- Obter a informação de cada pista relativamente aos tempos obtidos. A API deverá ser responsável por apresentar ao utilizador os tempos obtidos em determinada pista por determinado veículo e piloto que o conduziu (**Fig. 1**);
- Deverá ser possível armazenar em ficheiro os tempos obtidos em determinada pista para posteriormente serem apresentados. Os ficheiros deverão ser criados por pista, ou seja, deverá existir um ficheiro com os tempos obtidos para uma determinada pista (**Fig. 2**);
- Os dados relativos às dimensões da pista, checkpoints, e disposição inicial do veículo são fornecidos num ficheiro próprio (mesmo formato que o apresentado na **Fig. 2**);
- Os dados relativos ao tipo de veículo, modelo, velocidade, etc., são fornecidos num ficheiro próprio (mesmo formato que o apresentado na **Fig. 2**);
- A API deverá ser construída para que a adição de novos elementos (por exemplo, novas pistas) não implique a alteração do código implementado para as funcionalidades especificadas nos pontos anteriores.


Deverá ainda criar um programa que permita testar todas as funcionalidades implementadas.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo
	Segundo momento de avaliação - AC	2017/2018
	Curso	
	LEI – Licenciatura em Engenharia Informática LSIRC - Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular	
	Paradigmas da Programação	

Na **Figura 1** é apresentado um pequeno exemplo do final de uma corrida e os tempos obtidos com a mesma (os recursos disponibilizados e as imagens da figura são fornecidas com o projeto). Na imagem abaixo também é possível visualizar a tabela com **todos os resultados obtidos para uma determinada pista**. A tabela de resultados apenas mostra os sete melhores resultados obtidos ordenados ascendentemente pela melhor volta numa determinada pista (neste caso é para a pista “Level01”). Se o tempo obtido em determinada corrida for inferior ao pior resultado da tabela (os seja se for melhor que o sétimo resultado) ou se a tabela ainda não está totalmente preenchida, o resultado obtido aparece a **negrito** como mostra a figura. Caso contrário, o tempo obtido na presente corrida não é apresentado na tabela de resultados.



**Figura 1.** Exemplo do final de uma corrida e os tempos obtidos com a mesma.

	Tipo de Prova	Ano letivo
	Segundo momento de avaliação - AC	2017/2018
	Curso	
	LEI – Licenciatura em Engenharia Informática LSIRC - Licenciatura em Segurança Informática em Redes de Computadores	
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Unidade Curricular	
	Paradigmas da Programação	

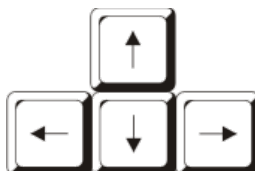
É ainda apresentado na **Figura 2**, um exemplo do ficheiro para armazenamento e carregamento dos tempos obtidos para cada pista.


```
[
{
  "Position": 1,
  "positionDetails": {
    "Vehicle": "car",
    "PilotName": "Telmo",
    "PilotId": 77,
    "BestLap": 17523,
    "TotalLaps": 2,
    "BestTime": 30450
  }
},
{
  "Position": 2,
  "positionDetails": {
    "Vehicle": "car",
    "PilotName": "Telmo",
    "PilotId": 77,
    "BestLap": 20394,
    "TotalLaps": 1,
    "BestTime": 20394
  }
}
]
```

**Figura 2.** Exemplo do ficheiro para armazenamento e carregamento dos tempos.

Tenha em consideração que:

- São disponibilizadas algumas imagens para inclusão no cenário do jogo. Pode, no entanto, adicionar novas imagens. As imagens devem encontrar-se no formato *png* e possuir fundo transparente;
- É disponibilizado juntamente com os recursos, a API JsonSimple (dentro da pasta *libraries*) para leitura e armazenamento dos vários ficheiros. Podem no entanto usar outras API's para o efeito.
- O controlo do veículo encontra-se pré-configurado para que a sua jogabilidade seja realizada através das teclas direcionais (tecla cima e baixo é para acelerar e travar, respetivamente e esquerda e direita é para viragem do veículo):



 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo
	Segundo momento de avaliação - AC	2017/2018
	Curso	LEI – Licenciatura em Engenharia Informática LSIRC - Licenciatura em Segurança Informática em Redes de Computadores
	Unidade Curricular	Paradigmas da Programação

## Elaboração do trabalho

Este trabalho é realizado em grupo que deverá ser composto no máximo por **dois** alunos da unidade curricular. A **constituição dos grupos de trabalho** deve ser feita através da página da unidade curricular de Paradigmas de Programação em <http://moodle.estg.ipp.pt>. De forma a simplificar e agilizar o processo de avaliação deste momento de avaliação, grupos de trabalho da mesma turma devem criar o grupo de trabalho na turma a que pertencem. Quando o grupo de trabalho envolve alunos de turmas diferentes, podem escolher uma das turmas 1, 2, 3 ou 4.

## Datas e considerações

O trabalho deve ser entregue até às **23:55** horas do dia **03/06/2018**, devendo a entrega ser feita através da página da unidade curricular de Paradigmas de Programação em <http://moodle.estg.ipp.pt>. Será ainda realizada a defesa do trabalho no horário de aulas respetivo para cada turma. A defesa será realizada por turnos e a hora exata para cada aluno (tendo em consideração o horário da respetiva turma) será comunicada na plataforma moodle após a entrega do trabalho. No caso de um grupo ser constituído por alunos de turmas diferentes, a marcação do horário será aleatória.

Considera-se por **defesa satisfatória**, quando o aluno **prova** que realizou o trabalho submetido e que **domina todos** os conceitos dados na disciplina de Paradigmas de Programação e aplicados na resolução do trabalho. Tentativas de **fraude**, resultarão na avaliação do trabalho como: **Fraude Académica**.

## Observações:

A entrega dos trabalhos deve obedecer aos requisitos identificados no ponto **Formato da entrega**. Os alunos que não realizem a entrega do trabalho até à data/hora definida serão sujeitos a penalização.

## Formato da entrega

Os trabalhos entregues deverão evitar (se possível) utilizar caminhos absolutos ou endereços específicos, de modo a que possam ser facilmente utilizados em qualquer máquina. Para além disso e no sentido de facilitar a receção dos vários trabalhos recebidos, estes deverão observar as seguintes regras:

- **Todos os elementos do grupo** deverão submeter o trabalho no link respetivo (**Entrega do Trabalho - Turno 1, 2, 3 ou 4**) na turma a que pertencem;
- O trabalho desenvolvido deverá ser entregue através do moodle, através da submissão de um ficheiro com o nome PP\_MA02\_<nr\_do\_aluno>\_<nr\_do\_aluno>.zip, contendo:
  - Os ficheiros criados incluindo o(s) projeto(s) do IDE Netbeans e uma pasta com a distribuição (jar) da solução proposta.
  - Recorra a comentários JavaDoc, e não só, de modo a documentar, o mais exaustivamente possível, o código desenvolvido.
  - Cada ficheiro de código entregue por cada grupo terá de possuir no início do mesmo um comentário com pelo menos a seguinte informação (com as adaptações óbvias para cada aluno/grupo):

```
/*
 * Nome: <Nome completo do aluno>
 * Número: <Número mecanográfico do aluno>
 * Turma: <Turma do aluno>
 *
 * Nome: <Nome completo do colega de grupo>
 * Número: <Número mecanográfico do colega de grupo>
 * Turma: <Turma do colega de grupo>
 */
```