

## Projeto: Game Engine (Motor de Jogo)

Projeto para trabalho em grupo

O enunciado do projeto estará disponível em vídeo após a sua apresentação nas aulas práticas

### Entregável 1/3:

- Cada grupo escreve um *pequeno* relatório incluindo:
  - i) a identificação do projeto, do entregável, do número do grupo e da identificação dos elementos que participaram na elaboração do trabalho *e só esses*;
  - ii) a descrição do jogo escolhido para concretização do motor de jogo;
  - iii) o diagrama de classes UML inicial (ignorando a interface gráfica com o utilizador), incluindo uma breve descrição do diagrama e da responsabilidade de cada classe;
  - iv) todas as opções de projeto tomadas, designadamente padrões de projeto;
  - v) testes unitários JUnit.

comprima num arquivo zip e submeta como *TP1* a <http://deei-mooshak.ualg.pt/~jvo/POO/Entregas> (uma submissão por grupo; depois de submetido não será possível alterar)

Até 27 de abril de 2025 – será aplicada uma penalização por cada dia de atraso.

### Entregável 2/3:

- Ignorando a interface gráfica com o utilizador, submeta o seu código e relatório atualizado (dentro de um ficheiro zip) como *TP2* para <http://deei-mooshak.ualg.pt/~jvo/POO/Entregas> incluindo:
  - i) entregável 1 (eventualmente melhorado)
  - ii) Javadoc
  - iii) o diagrama de classes UML de implementação
  - iv) qualquer conclusão ou observação relevante
  - v) referências bibliográficas, páginas web consultadas, incluindo ferramentas IA e respetivas prompts, quando aplicável

(uma submissão por grupo; depois de submetido não será possível alterar)

Até 11 de maio de 2025 – será aplicada uma penalização por cada dia de atraso.

**Entregável 3/3:**

- Tendo em consideração a interface gráfica com o utilizador, submeta o seu código e relatório (dentro de um ficheiro zip) como TP3 para <http://deei-mooshak.ualg.pt/~jvo/POO/Entregas> incluindo:
  - i) Todos os elementos dos entregáveis 1 e 2, eventualmente melhorados e incluindo agora o desenvolvimento e utilização da interface gráfica.

(uma submissão por grupo; depois de submetido não será possível alterar)

Até 25 de maio de 2025 – será aplicada uma penalização por cada dia de atraso.

**Requisitos adicionais para todos os entregáveis, quando aplicável**

- 1) Reuse as classes desenvolvidas anteriormente pelos elementos do grupo.  
Quando uma classe tiver sido desenvolvida ao longo do semestre (e.g., classe Ponto) o grupo terá que escolher uma das classes (Ponto) desenvolvida pelos seus membros. Não serão consideradas para avaliação classes diferentes das existentes no grupo. Havendo no grupo classes com responsabilidades semelhantes, o grupo seleciona aquelas que considerar mais apropriadas e ignora as outras;
- 2) Utilize desenvolvimento guiado por testes;
- 3) Empregue os princípios e técnicas de programação orientada a objetos;
- 4) Empregue, pelo menos, dois padrões de projeto e justifique o seu emprego no relatório;
- 5) Utilize classes genéricas, quando necessário;
- 6) Utilize os mecanismos de gestão de exceções;
- 7) O cliente inicializa apenas a classe principal do jogo, a do *user interface* (UI), relaciona-as e chama um método para iniciar o jogo;
- 8) A UI tem de estar separada do modelo do jogo, obrigatoriamente. A comunicação entre o jogo e a UI é efetuada através de uma *Application Programming Interface* (API) definida por uma ou mais Java interfaces;
- 9) A UI só é gráfica no entregável 3.
- 10) Comente apropriadamente o código, i.e.,

**Classes:**

- i) Descrição da responsabilidade da classe
- ii) @author
- iii) @version (incluir uma data)
- iv) @inv (descrição da invariante usada)

Métodos:

- i) Descrição do que o método faz
- ii) @param (descrição do parâmetro incluindo restrições de valores se necessário, isto é, pré-condições)
- iii) @post condições (se aplicável)
- iv) @return (se aplicável)
- v) @see (qualquer referência bibliográfica, sítio web consultado, ou ferramenta de IA e respectivas prompts, usadas no desenvolvimento do código respetivo)