 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Época de Recurso	Ano letivo 2023/2024	Data
	Curso Licenciatura em Engenharia Informática Licenciatura em Segurança Informática em Redes de Computadores	Data entrega	
	Unidade Curricular Estruturas de dados		

Observações

- Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de **Estruturas de Dados (ED)** e irá servir para avaliar a respetiva componente prática.
- Os estudantes deverão juntar-se em grupos de **2 elementos** de modo a dividir, da melhor forma, as tarefas definidas neste trabalho. Excecionalmente, e quando se justifique, poderão ser considerados grupos com outro número de elementos.

Objetivos

- Utilizar os conhecimentos sobre estruturas de dados para escolher as estruturas de dados que melhor se aplicam à resolução do problema proposto;
- Conceber e implementar, eficaz e eficientemente, o algoritmo de resolução do problema proposto.

Implementação

- Deverá ser usada a linguagem Java;
- O código deverá estar comentado através do JavaDoc;
- Não pode ser usada nenhuma coleção da plataforma de coleções do Java. Sempre que for necessário terá de selecionar a estrutura de dados com o comportamento desejado desenvolvida durante as aulas. Cada grupo deverá usar as **suas** versões e que devem **obrigatoriamente** utilizar as *Abstract Data Type* (ADT) disponibilizadas nas aulas.

Nota: As principais alterações em relação ao enunciado de época normal estão sinalizadas a sublinhado.


Resumo

O objetivo deste trabalho é desenvolver um jogo denominado de *Capture the Flag* ¹. Nesse contexto, dois jogadores enfrentam-se na missão de capturar a bandeira do adversário. Cada jogador conta com um conjunto de *bots* cujo comportamento de movimentação é definido pelos jogadores. Isso significa que cada *bot* se movimentará consoante o algoritmo definido para este pelo jogador, como, por exemplo, um algoritmo que direciona o *bot* a seguir os pontos que compõem o caminho mais curto até o campo inimigo em termos de distância. Essa abordagem proporciona aos jogadores um controle estratégico sobre as táticas de movimentação de seus *bots*, intensificando a competitividade e a complexidade do jogo.

Cada partida deste jogo deve seguir o seguinte fluxo:

1. Deve ser possível aos jogadores gerar um novo mapa inserindo as características do mapa de jogo ou possibilitar a importação de um mapa previamente definido e exportado (o formato de exportação é da responsabilidade de cada grupo de trabalho). Na geração de mapas, os jogadores acordam e configuram as características do mapa de jogo introduzindo: 1) a quantidade de localizações existentes no mapa, 2) o tipo de caminhos

¹ Capturar a bandeira

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Época de Recurso	Ano letivo 2023/2024	Data
	Curso Licenciatura em Engenharia Informática Licenciatura em Segurança Informática em Redes de Computadores	Data entrega	
	Unidade Curricular Estruturas de dados		

que serão gerados, i.e., bidirecionais ou não e 3) a densidade das arestas, p.e., 50% num grafo direcionado indica que $(N * (N - 1)) * 0.5$ arestas devem estar presentes². Após escolhidas as definições, os mapas devem ser gerados aleatoriamente considerando as características inseridas. As distâncias entre os pontos também devem ser geradas aleatoriamente num intervalo de 1 a 15 quilómetros. Deve ser possível exportar o mapa para que possa ser utilizado noutra partida;


- Depois do mapa ser definido, ambos os jogadores selecionam a localização das bandeiras no mapa³ que corresponde à base do jogador.
- De seguida os jogadores acordam e estipulam o número de *bots* que cada jogador terá à sua disponibilidade. O número de *bots* para cada jogador pode ser distinto.
- Dado o número predefinido de *bots* mencionado anteriormente, os jogadores devem, de forma dinâmica, atribuir a cada um deles algoritmos proveniente das diversas opções que cada grupo de trabalho deve disponibilizar. Estes algoritmos determinarão o comportamento específico que cada *bot* adotará durante sua ronda no jogo. Os algoritmos atribuídos aos *bots* determinarão como se irão movimentar durante o jogo (capturar e devolver a bandeira), influenciando diretamente a estratégia de cada *bot* para capturar a bandeira inimiga e voltar com a bandeira para a sua base.
- Quando os jogadores iniciam a partida, deve ser decidido aleatoriamente qual o jogador que irá começar. No início da partida todos os *bots* deverão estar localizados na mesma posição que a bandeira do seu jogador.
- A cada ronda, alternando pelos jogadores, um dos *bots* deverá movimentar-se. Os *bots* devem adotar uma abordagem em que cada um, à vez, move-se de acordo com uma lógica sequencial, assegurando que cada *bot* tenha sua oportunidade de avançar, sem a repetição imediata de movimentos, proporcionando assim uma distribuição equitativa das ações durante o jogo, e.g., num jogo com 3 *bots*, joga o *bot* 1, depois o 2, seguindo do 3. Na quarta ronda, deve voltar ao *bot* 1.
- Cada localização pode acolher mais do que um bot. Se ao entrar uma localização existir um bot adversário com a bandeira da sua equipa, esta deverá voltar para a sua base. No entanto, se um bot com a bandeira entra numa localização, a bandeira permanece na posse do bot. Se coincidirem as duas bandeiras na mesma localização, ambas voltam para a base.
- O jogo termina quando a bandeira de um jogador chega à base da equipa adversária.

Deverá ser possível aos jogadores acompanhar visualmente as movimentações realizadas. Os grupos podem recorrer a bibliotecas externas para esta componente do trabalho, no entanto, a utilização da consola é perfeitamente aceitável.

Os estudantes devem disponibilizar pelo menos 3 algoritmos de movimentação para os *bots*, sendo que deverá utilizar obrigatoriamente o caminho mais curto e a árvore geradora de custo mínimo.

² Onde N representa o número de vértices no grafo

³ De modo a permitir a jogabilidade, os jogadores não deverão poder escolher a mesma localização.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Época de Recurso	Ano letivo 2023/2024	Data
	Curso Licenciatura em Engenharia Informática Licenciatura em Segurança Informática em Redes de Computadores	Data entrega	
	Unidade Curricular Estruturas de dados		

Pretende-se que os estudantes desenvolvam uma API para suportar as funcionalidades do jogo. Esta API deverá ser utilizada por um programa que permita a jogabilidade na consola, por exemplo.

Avaliação

- Apenas serão considerados para avaliação os trabalhos entregues antes da data limite definida pelos docentes da UC e disponibilizada no Moodle. A não submissão do trabalho até esta data invalida a sua avaliação;
- Todos os elementos dos grupos devem submeter os elementos de avaliação. A submissão deverá consistir num ficheiro .zip cujo nome deve seguir a norma ED_EN_8XXXXXX_8XXXXXX (8XXXXXX indica o número de estudante de um elemento do grupo);
- A defesa é obrigatória e será realizada no dia do exame da época correspondente (ver calendário de exames). Para avaliar o desempenho durante a defesa poderá ser pedido a cada estudante, a implementação de novas funcionalidades ou a modificação de funcionalidades já existentes no seu trabalho. Cada estudante poderá ainda ser questionado sobre as decisões tomadas durante a implementação do projeto. Poderão ser colocadas a cada estudante, questões orais e/ou escritas. As questões escritas poderão implicar a utilização de um computador e a implementação de alterações ao projeto submetido (neste caso, pode ser necessário uma nova submissão no final da prova). Será avaliado o desempenho do estudante e o seu conhecimento relativamente ao projeto submetido.
- Critérios de avaliação:
 - A escolha apropriada das estruturas de dados e o uso destas será o fator de avaliação preponderante em todas as funcionalidades implementadas.
 - Boas práticas:
 - Comentários e JavaDoc.
 - Uso de controlo de versões (desde o início do projeto).
 - Teste unitários.
 - Uso das convenções do Java (ex.: <https://www.geeksforgeeks.org/java-naming-conventions/>).