

Programação Orientada por Objetos

Ficha 1

1. No Greenfoot abra o cenário **crabgame-v2** (disponível no Moodle). Teste o cenário e analise o código de forma a entender as diferentes funcionalidades que definem o comportamento dos atores. Crie um método chamado **randomTurn** que faz que objetos da classe Crab girem aleatoriamente para a esquerda entre 0 e 30° em 20% dos casos. Faça a chamada deste método em **act()**. Consulte **Greenfoot Class Documentation** no menu **Help** para obter informação sobre métodos sempre que necessário.
2. Retire do método **act()** da classe Crab a chamada aos métodos **turnAtEdge()** e **randomTurn()**. Crie um método chamado **checkKeyPress** que permite controlar o movimento do objeto da classe Crab com as teclas A e L do teclado (girar 5° para a esquerda e direita, respetivamente). Faça a chamada deste método em **act()**.
3. Crie uma nova classe no cenário.
 - a. A nova classe deverá ser uma subclasse de **Actor** e deverá ter o nome **Lobster**. Use para esta classe a imagem *lobster.png*, que já faz parte da galeria do Greenfoot. Teste.
 - b. Neste momento, um objeto da classe **Lobster** não faz nada quando colocado no mundo. Faça que o objeto se mova 5 células em cada iteração, usando para esse fim um método herdado da classe **Actor**. Além disso, crie um método para que objetos desta classe girem 20° ao tocar as fronteiras do mundo.
 - c. Crie um método que faz que objetos desta classe façam um giro 10% das vezes, que pode ser para a esquerda ou para a direita, de um valor que varia aleatoriamente entre 0 e 40°.
4. Modifique o cenário de forma que objetos da classe Lobster detetem e *comam* objetos da classe Crab. Deverá ouvir-se o som *au* (ficheiro *au.wav*) cada vez que o objeto da classe Crab for apanhado. Teste.
5. Inclua no código um método chamado **populateWorld()** (retire o método atual **prepare()**) que inicializa no **CrabWorld**:
 - a. 6 (no total) objetos da classe Worm em posições aleatórias.
 - b. 4 objetos da classe Lobster junto aos cantos do mundo.
 - c. 1 objeto da classe Crab no centro do mundo.

Pista: a classe **CrabWorld** tem 2 métodos herdados de **World** que devolvem a largura e altura do mundo que devem ser utilizados para posicionar os objetos.
6. Modifique o código de forma a que a imagem de fundo do mundo seja a do ficheiro **bluerock.jpg**, disponível na pasta de imagens do projeto.

7. Modifique o cenário de forma que o jogo acabe quando o objeto da classe Crab for apanhado. Procure na documentação da classe **Greenfoot** um método que permita fazer isto.
8. Crie uma variável do tipo **inteiro** que conte o número de objetos da classe Worm que foram comidos pelo objeto da classe Crab. Quando todos os objetos da classe Worm forem comidos (são 6 no total) deverá ouvir-se o som *fanfare* (ficheiro *fanfare.wav*) e o jogo deverá acabar.
9. Coloque 2 obstáculos no mundo, nas coordenadas (200,200) e (400,400), que não podem ser transpostos pelos objetos da classe Lobster. Isto é, objetos desta classe deverão girar se encontram o obstáculo.
10. Modifique o código de forma que os obstáculos sejam 2 vezes maiores.