Laboratório 10: Jogo Snake (Opcional)

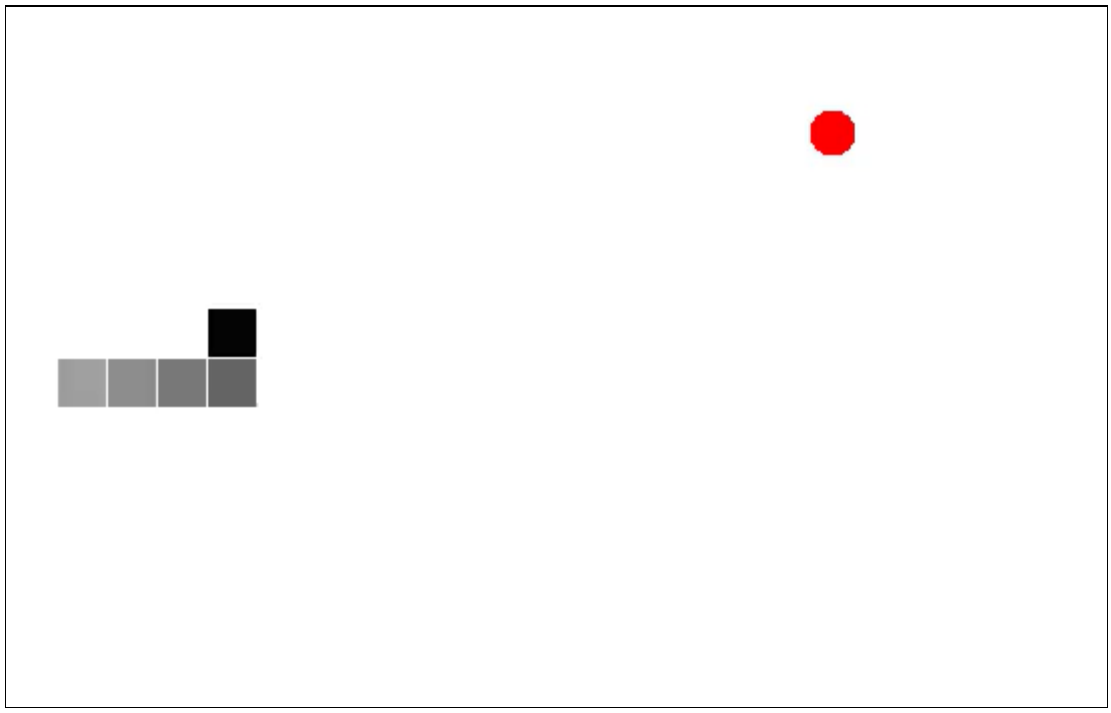

Figura 1: Jogo Snake. Clique no video para pausar.

Introdução

Snake é um jogo em que o jogador controla uma cobra em busca de comida. A cobra não pode colidir com as bordas da arena nem com ela mesma e, cada vez que ela come algo, cresce de tamanho (ver Figura 1).

Objetivo

- Neste trabalho você irá implementar um jogador automático para o Jogo Snake calculando, a cada passo, o próximo passo da cobra.
- Restrições:
  - Seu algoritmo não poderá ser completamente aleatório (como o exemplo implementado no código);
  - Seu algoritmo não poderá ser um caminho pré-definido (e.g., zig-zag).
  - Sua solução deverá levar em conta as informações disponíveis e, a cada passo, tomar uma decisão relativamente "inteligente" de qual caminho seguir.

Passos Iniciais

Para entender o que fazer, você precisará ler o JavaDoc do código-fonte, os comentários presentes no código-fonte e, em alguns casos, o próprio código-fonte, disponível aqui. Adicione os arquivos no seu eclipse e execute para jogar o jogo usando o teclado. O código-fonte é composto pelos seguintes arquivos:

Arquivo	Descrição

`Snake.java` → *JavaDoc*	Esta é a classe principal do jogo. Você não precisará modificar este arquivo, mas irá usar seus métodos públicos para pegar as informações da cobra, da comida e da arena para decidir qual será o próximo passo da cobra.
`SnakeJogador.java` → *JavaDoc*	Esta será a classe que você irá modificar para indicar qual será o próximo passo da cobra. O método `getDirecao()` desta classe será executado a cada passo da cobra para decidir qual direção seguir.
`SnakeMain.java` → *JavaDoc*	Contém o método `main` que instancia e inicia o jogo. Você precisará editar os comentários do método `main` para selecionar uma das três implementações possíveis deste método que permitem: 1. jogar o jogo na janela usando teclado; ou 2. jogar o jogo na janela usando o jogador implementado (esta será a versão que você deverá usar); ou 3. jogar o jogo sem janela usando o jogador implementado (usado na correção do trabalho).

Questão 1 – Classe `SnakeJogador`

0.00 / 10.00

Considere que as classes `Snake` e `SnakeMain` já foram submetidas. Veja o código fonte e acesse o JavaDoc das classes para entender como elas funcionam.

Nesta questão, você irá submeter a classe `SnakeJogador` contendo a implementação do método `char getDirecao()`. Este método deve retornar a direção que a cabeça da cobra deverá seguir para se mover para a próxima célula. Esta direção pode ser: **C**ima, **D**ireita, **B**aixo ou **E**squerda. No código-fonte da classe disponibilizada, existe uma implementação de exemplo para o método `char getDirecao()`. Nesta implementação, uma direção aleatória é retornada, fazendo a cobra andar cegamente pela arena.

Como implementar outras classes: como este trabalho só aceita a submissão de um único arquivo, caso você queira implementar outras classes auxiliares para a sua solução, você deverá colocá-las dentro deste mesmo arquivo. Você pode fazer isso de duas formas:

1. *Declarando uma classe dentro de outra:* neste caso, tem-se o que é conhecido como classe interna (do inglês *inner class* ou *nested class*). Classes internas podem ser `public`, `protected`, `friendly` ou `private` e seu acesso seguirá as mesmas regras do acesso a um método com estes modificadores.
2. *Declarando várias top-level classes no mesmo arquivo:* neste caso, após o término de uma classe, basta começar a declarar a próxima. Java permite, apesar de não ser recomendado, que você coloque várias classes em um único arquivo, desde que dentre

todas as classes apenas uma seja `public` e esta classe `public` deverá ter o mesmo nome do arquivo. Entretanto, o modificador de acesso das outras *top-level classes* deverá ser obrigatoriamente *friendly* (sem modificador nenhum).

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "`SnakeJogador.java`"