

FACAMP – Linguagens de Programação (COM710)
Para demonstração/entrega no dia 27/11/2019
Avaliação individual: trabalhos copiados terão nota zero
Peso na média: 30% da média de trabalhos
Space Invaders 2019

Objetivo

O objetivo do trabalho é desenvolver uma implementação baseada no jogo Space Invaders em Java, utilizando Java Swing/AWT com Canvas.

Requisitos e Regras

O jogo desenvolvido deve receber input do teclado (tratar o evento do acionamento de teclas), atualizar a interface gráfica a cada frame e conter os seguintes elementos:

- **Nave**, que se localiza na parte inferior da tela e se move para a esquerda e para a direita com as teclas AD e ← → .
 - Movimentação: A nave não pode sair da tela, e nem se mover verticalmente.
- **Tiro** da nave, que inicia na posição da nave e viaja em velocidade constante para cima. Não é possível atirar uma segunda vez quando o míssil ainda está na tela e não colidiu com nenhum monstro. Disparo através da tecla Espaço.
 - Movimentação: O míssil viaja apenas verticalmente para cima. Uma vez que ele sai da tela ou atinge um monstro, ele desaparece e a nave pode atirar um novo míssil. **Caso a barra de espaço se mantenha pressionada, um novo míssil deve ser disparado toda vez que disponível segundo as regras acima.**
- **Grid** de monstros que se movimentam de forma descontínua (em saltos) no canto superior da tela. Eles devem desaparecer uma vez que o tiro os atinge.
 - Movimentação: Os monstros se movem todos numa mesma direção horizontal até estarem próximos do final da tela. Então, se movem uma vez para baixo e novamente na horizontal, na direção oposta à anterior (vide jogo original. Ex.: <https://www.minijogos.com.br/jogo/space-invaders/>).
- **Label** com score, que revela quantos monstros foram mortos.

Além disso, o jogo (tela principal) deve manter uma coleção/matriz com os objetos monstros, que será acessada constantemente para atualização de seus status (ativo, morto, ...). Quem faz a modificação de posição da matriz e atualiza a tela deve ser uma thread auxiliar.

Deve ser implementada uma terceira linha de execução (thread) que, de tempos em tempos, sorteia um dos monstros ainda ativos, e aplica nele um comportamento especial:

- Ele deve ser retirado da coleção e assumir um comportamento diferente (executado em uma nova thread também), movendo-se diagonalmente e para baixo, tendo como alvo final a nave;
- A concorrência de acesso pelas diferentes threads à coleção/matriz de monstros deve ser devidamente tratada.

Tela exemplo

A seguinte imagem serve como exemplo e não precisa ser seguida fielmente.
Podem ser incluídos desenhos (gif, p.e) no lugar dos objetos geométricos.

