

Universidade Federal do Paraná  
Avaliação #04 - Programação 02  
BCC1 (Prof. Vinícius) e BCC2 (Prof. David)

## ***Space Invaders***

### Implementação

*Space Invaders* é um *videogame arcade* do tipo atire neles (*shoot 'em up*) desenvolvido por Tomohiro Nishikado em 1978 [1]. *Space Invaders* foi o primeiro jogo de tiro fixo e estabeleceu o modelo para o gênero. O objetivo deste jogo é derrotar ondas descendentes de *aliens* (enemies) com um laser que se move horizontalmente para ganhar o máximo de pontos possíveis. Uma versão livre desse jogo pode ser encontrada na internet em [2] e acessada por <https://freeinvaders.org/>.

Como vimos na A3, é possível implementar uma versão simplificada do jogo *space invaders* através de um mapa de grade utilizando uma matriz. Para a A4, o objetivo é terminar de implementar a lógica do jogo *space invaders* e, em seguida, incluir uma interface gráfica jogável sobre ela.

Esse será um desenvolvimento mais “livre” e podemos compreender o objetivo final como **DESENVOLVER UM JOGO O MAIS PARECIDO POSSÍVEL COM O SPACE INVADERS ORIGINAL** (<https://freeinvaders.org/>).

Assim, estabelece-se apenas um conjunto de requisitos mínimos para o jogo, sendo que a implementação destes equivale a 60% da nota total do trabalho e funcionalidades extras só serão avaliadas dada a implementação dos requisitos mínimos.

## **REQUISITOS RELACIONADOS ÀS FUNCIONALIDADES QUE DEVEM SER IMPLEMENTADAS**

### **(INIMIGOS)**

- Devem existir três tipos de inimigos:
  - Fracos, não atiram caso exista uma entidade à sua frente ou se já existir um projétil na coluna, causam um (1) de dano em obstáculos;
  - Intermediários, não atiram caso exista um projétil na coluna, causam dois (2) de dano em obstáculos;
  - Fortes, podem atirar mesmo se existir até um projétil na coluna, causam dois (2) de dano em obstáculos.
- A cada “rodada” apenas dois (2) inimigos podem atirar, dando preferência àqueles mais próximos à nave.
- Os inimigos têm apenas uma vida, sendo esta removida uma vez que o mesmo é acertado por um projétil da nave.

### **(NAVE)**

- A nave pode se mover à direita e à esquerda, até as bordas do mapa.
- A nave pode atirar caso não exista um projétil seu na coluna.
  - O projétil da nave causa um (1) de dano em obstáculos.
- A nave inicia com três vidas, sendo que uma vida é removida cada vez que a mesma é acertada; e uma vida adicionada a cada fim de rodada (ao máximo de cinco vidas).

### **(OBSTÁCULOS)**

- Obstáculos podem ser colocados entre a nave e os inimigos.
  - Após receber dez (10) de dano, o obstáculo é removido;
  - Você pode definir o número de obstáculos a partir do tamanho do tabuleiro (seu critério);
  - Os obstáculos não são reiniciados no início das rodadas.

### **(RODADAS)**

- Uma vez que todos os inimigos forem mortos, uma nova rodada deve ser iniciada, carregando novos inimigos.

### **(ETC)**

- Deve existir um contador de pontos entre todas as rodadas (não reiniciado).
  - Um inimigo fraco morto atribui dez (10) pontos ao jogador;
  - Um inimigo intermediário morto garante vinte (20) pontos ao jogador;
  - Um inimigo forte morto garante quarenta (40) pontos ao jogador.

## **REQUISITOS RELACIONADOS À INTERFACE QUE DEVEM SER IMPLEMENTADAS**

### **(MENU)**

- Deve existir um menu principal com a funcionalidade de iniciar o jogo.

### **(ENTIDADES E ANIMAÇÕES)**

- Você deve utilizar uma imagem específica para...
  - Nave;
  - Inimigo fraco (duas imagens alternantes);
  - Inimigo intermediário (duas imagens alternantes);
  - Inimigo forte (duas imagens alternantes).
- Cada tipo de inimigo deve ter um projétil diferente.
- Deve existir uma animação de explosão do inimigo e da nave, quando estes forem derrotados.

### **(TELA)**

- O placar e a quantidade de vidas do jogador devem estar explícitos na tela
- Todas as funcionalidades descritas devem estar acessíveis e/ou visualizáveis através da interface gráfica

## LISTA DE FUNCIONALIDADES ADICIONAIS E OPCIONAIS (VOCÊ PODE COMPLETAR 100 AQUI)

- Inimigos se movem, em bloco, para a direita e para a esquerda [15 PONTOS]
- Projéteis da nave podem se chocar com os dos inimigos e se anularem [10 PONTOS]
- Inserção de colunas vazias entre os inimigos [5 PONTOS]
- Progressão de dificuldade a cada nível (pelo menos 4 progressões) [10 PONTOS]
- Nave inimiga (vermelha) com pontuação extra atrás de todos os inimigos [10 PONTOS]
- Implementação de poderes bônus (dois tiros ou escudo) temporários para a nave [15 PONTOS]
- Implementação de novos tipos de inimigos, incluindo novo modelo de ataque e nova imagem [5 PONTOS CADA, NO MÁXIMO DOIS NOVOS]
- Implementação de um inimigo que atira projéteis em diagonal e que ricocheteiam na borda do mapa [15 PONTOS]
- Salvar e carregar um jogo salvo [10 PONTOS]

**SEJA CRIATIVO:** se você quiser propor novas funcionalidades para a lista, procure o professor! As funcionalidades legais terão um peso atribuído e serão incluídas na lista de opcionais.

## REQUISITOS TÉCNICOS E ENTREGA DO TRABALHO

Considere as seguintes informações para executar e entregar o trabalho:

- Você deve utilizar a **linguagem C** para a implementação do trabalho
  - E a **biblioteca Allegro** para a implementação da interface gráfica
- Você deve preparar um ***makefile*** (pelo menos, com operações *make* e *make clean*) para a compilação do seu programa
- Você deve **observar todos os requisitos de entrega do trabalho elencados no programa da disciplina**, não cumprir os mesmos acarretará um desconto de 5/100 na sua nota final
- Não esqueça de **tratar possíveis erros**