

# **Documentação - Trabalho Prático**

## **“Space Invaders”**

**Universidade Federal de Minas Gerais**  
**Programação e Desenvolvimento de Softwares I**  
**01/2025**

**Bernardo Marques Guimarães**

### **1 - Introdução**

Esta documentação tem como objetivo explicar o funcionamento e a implementação do jogo desenvolvido através da biblioteca Allegro e baseado nos conhecimentos adquiridos no decorrer do semestre na matéria de PDS-1.

### **2 - Modo de jogar**

O objetivo do jogo consiste na eliminação de todas as naves alienígenas através dos tiros disparados pela nave controlada pelo jogador, os tiros saem um por vez a medida que ultrapassam o limite superior da tela ou atingem algum inimigo.

A forma de controlar o jogo é:

“**A**” : Movimenta a nave para a esquerda;

“**D**” : Movimenta a nave para a direita;

“**ESPAÇO**” : Efetua os disparos da nave;

“**R**” : Reinicia o jogo;

A pontuação do jogo é determinada por um cálculo baseado na distância entre a nave e o alien eliminado no momento em que o tiro é disparado.

### **3 - Implementação**

#### **3.1 - Estrutura de dados**

→ A struct aliens representa o bloco de aliens, tendo em seu arquivo “.h” funções responsáveis pela inicialização do bloco, desenho do bloco de aliens, movimentação do bloco e verificação se o alien está vivo.

→ A struct nave representa a nave que é controlada pelo jogador, sendo acompanhada por funções que a iniciam, a desenharam e garantem sua movimentação.

→ A struct *tiros* representa os disparos feitos pela nave controlada pelo jogador, possuindo funções que o inicia, desenha, cria e atualiza sua posição.

### 3.2 - Funções e procedimentos

O código foi dividido em diversos arquivos “.h” e “.c” a fim de garantir maior legibilidade ao código principal e melhor organização do programa, sendo que dentre essas divisões estão “*utilidades.h*” e “*utilidades.c*”, “*alien2.h*” e “*alien2.c*”, “*nave2.h*” e “*nave2.c*”, “*tiros.h*” e “*tiros.c*”, “*colisoes.h*” e “*colisoes.c*” e “*score.h*” e “*score.c*”.

→ Em *utilidades.h* estão definidos todos os macros utilizados em todas as funções, tendo esse arquivo sido incluído em todos os demais que necessitam desses itens, além disso possui o cabeçalho da função *void draw\_cenario* que está completa em *utilidades.c*, essa função tem como objetivo dar cor ao fundo do mapa e desenhar a grama presente na parte interior da tela.

→ Em *aliens.h* está presente a struct *Alien* e os cabeçalhos das funções relativas aos aliens, sendo elas *void initAlien*, *void draw\_alien\_com\_bloco*, *void update\_all.aliens* e *int verifica.aliens*. A função *init\_alien* faz a inicialização dos aliens, conferindo se estão ativos, dando cor, determinando velocidades nos eixos “x” e “y” e a posição dos aliens, a função *draw\_alien\_com\_bloco* é utilizada para o bloco de aliens, considerando-os ativos ou não para que o desenho seja mantido na tela, a função *update\_all.aliens* tem como objetivo garantir que ao atingir as bordas da tela o bloco de aliens fará a movimentação desejada e a função *verifica.aliens* serve para analisar se os aliens estão ativos ou não.

→ Em *nave2.h* está a struct *Nave* e o cabeçalho das funções relativa à nave controlada pelo jogador, sendo *void draw\_nave*, *void init\_nave* e *void update\_nave*. A função *init\_nave* tem como objetivo a inicialização da nave, posicionando-a na tela, dando velocidade de movimentação e cor e fornecendo a base do movimento, a função *draw\_nave* garante onde a base da nave estará posicionada e a desenha e a função *update\_nave* garante a movimentação da nave pela tela respeitando os limites.

→ Em *tiros.h* está a struct *Tiros* e os cabeçalhos das funções relativas aos disparos efetuados pela nave, sendo elas *void init\_tiros*, *int update\_tiros*, *void draw\_tiros* e *void cria\_tiro*. A função *init\_tiro* passa o parâmetros do disparo que confere se ele foi efetuado ou não, a função *draw\_tiro* é

responsável pelo desenho do tiro, já a função *update\_tiros* confere se o tiro já foi efetuado e se ele está dentro dos limites da tela ou atingiu algum alien, verificando se um novo tiro pode ser disparado, caso tenha atingido algum alien, a função calcula a quantidade de pontos ganho pela morte do alien específico e a função *cria\_tiro* passa demais parâmetros do tiro, garantindo que ele sairá do ponto central mais alto da nave, se ele está ativo e a velocidade de movimentação.

→ Em *colisoes.h* está presente funções que identificam as colisões possíveis no jogo, sendo elas *int colisao\_tiro\_alien*, *int colisao\_alien\_solo* e *int colisao\_alien\_nave*. Além disso, está presente também a função *float calcular\_distancia*, responsável por calcular a distância entre o alien e a nave no momento em que o tiro que o matou foi disparado e utilizada para calcular a pontuação ganha pelo jogador por essa eliminação.

→ Em *score.h* estão armazenadas as funções responsáveis pelo carregamento do recorde, o salvamento de um novo recorde e a impressão dos textos na tela durante o jogo, as funções são *int carregar\_record*, *salvar\_record* e *draw\_pontuacao*.

## 4 - Programa principal (main)

O programa principal foi simplificado através da modularização e da criação de arquivos como demonstrado acima, porém ainda existem algumas funcionalidades nele, como a inicialização do Allegro, o carregamento de fontes, a declaração de algumas constantes utilizadas apenas no código principal. Após essa parte as primeiras funções são chamadas e um “while” é iniciado, esse “while” serve para que o que está dentro dele rode enquanto o jogo estiver funcionando, dentro é utilizado um switch para determinação de funções para teclas, como movimento, atirar e reiniciar o jogo, logo em seguida, ainda dentro do loop, são chamadas as funções implementadas anteriormente e que dão origem ao jogo, após a chamada de todas as funções existe uma verificação de vitória, imprimindo na tela o resultado (positivo ou negativo), da pontuação final e do recorde, porém antes do recorde existe uma conferência se a pontuação daquela partida supera o recorde anterior para atualização da nova maior pontuação e a impressão da mensagem para reiniciar o jogo caso desejado.