

Documentação - Trabalho Prático
"Space Invaders"
Universidade Federal de Minas
Gerais
Programação e Desenvolvimento de
Software 1–PDS
06/2025
Arthur Maciel de Silva e Sena

1- Introdução-

No semestre de 2025/1, o trabalho prático da disciplina Programação e Desenvolvimento de Software 1 foi desenvolver um jogo na linguagem C (linguagem aprendida durante o semestre letivo), para testar os conceitos aprendidos ao longo da disciplina. Para a parte gráfica do trabalho, foi utilizada a biblioteca Allegro (www.allegro.cc), apresentada aos alunos durante a disciplina.

A proposta do trabalho é que o jogo seja parecido com o famoso jogo criado em 1978, chamado *Space Invaders*, um jogo de ação e aventura no qual o jogador controla uma nave e atira com ela, lutando contra aliens que estão invadindo o planeta Terra. Os aliens se movem em grupo, e o jogo só termina quando todos os aliens são eliminados ou quando eles alcançam a base do jogo.

Como rodar-

O trabalho foi desenvolvido no sistema operacional Ubuntu, que é baseado no Linux. Logo, talvez a solução não possa ser implementada em outros sistemas operacionais, como o Windows e o Mac.

Instale o Allegro

Para instalar o Allegro5 no Linux, use os seguintes comandos:

1. `sudo apt-get install build-essential subversion cmake xorg-dev libgl1-mesa-dev libglu-dev`
2. `sudo apt-get install libpng-dev libz-dev libcurl4-gnutls-dev libfreetype6-dev libjpeg-dev libvorbis-dev libopenal-dev libphysfs-dev libgtk2.0-dev libasound-dev libflac-dev libdumb1-dev`
3. `sudo apt-get install exuberant-ctags dvi2ps dvipdfmx latex2html pandoc`
4. `sudo add-apt-repository ppa:allegro/5.2`
5. `sudo apt-get install liballegro*5.2 liballegro*5-dev`

Instale o projeto do jogo

Caso você não tenha o arquivo zipado, ele pode ser encontrado no seguinte link:

<https://github.com/ArthurMacielSs/Space-Invaders>

Utilize o comando para compilar o programa

Dentro da pasta do jogo, em um terminal, digite o seguinte comando:

```
gcc constants.c allegroFunctions.c creatingInvadersElements.c updatingInvadersElements.c  
allocationFunctions.c fileFunctions.c invaders.c -o invaders $(pkg-config allegro_dialog-5  
allegro_primitives-5 allegro_audio-5 allegro_acodec-5 allegro_ttf-5 allegro_font-5  
allegro_image-5 allegro-5 allegro_font-5 --libs --cflags) && ./invaders
```

A nave atira usando o espaço, move usando a e q, ou as setas.

Estrutura do código

Visando uma melhor organização e legibilidade do código, ele foi modularizado em alguns arquivos. A seguir, apresenta-se uma explicação básica com o nome de cada arquivo e o que ele implementa.

Invaders.c

O código principal do programa é responsável por inicializar todos os objetos utilizados pela biblioteca Allegro, bem como as variáveis principais do jogo. Ele também executa a lógica central da aplicação.

Esse código abre o arquivo que será acessado durante o programa, aloca dinamicamente a memória necessária para a matriz de aliens (cujo tamanho varia de acordo com a fase), inicializa a variável que controla o número da fase, exibe a tela inicial do jogo, e define a variável auxiliar que indica se o jogo está em andamento ou não. Além disso, imprime na tela informações como o recorde, a fase atual e a pontuação.

A fila de eventos é então iniciada, e o jogo se desenrola com base nas entradas do teclado. A sequência de eventos inclui:

- Inicialização e desenho de todos os elementos da tela: aliens, nave, cenário e tiros.
- Na função responsável por desenhar os aliens, o programa verifica quais ainda estão vivos, desativa os que foram eliminados e, caso todos tenham sido destruídos e não se esteja na última fase, retorna um valor que faz o jogo carregar a próxima fase.
- Caso o jogador vença ou perca, o jogo exibe a tela final correspondente.

O mesmo código também verifica colisões entre os aliens e a nave ou a grama. Se alguma dessas colisões ocorrer, o jogo é encerrado. Além disso, o programa gerencia a movimentação da nave e implementa a lógica dos disparos.

allegroFunctions.c

Este arquivo é responsável por inicializar a biblioteca Allegro e exibir as telas de início e fim do jogo.

A função **initialize_Allegro** prepara todos os recursos necessários para o funcionamento do jogo:

- Cria a janela, a fila de eventos e o temporizador.

- Inicializa os módulos de imagem, áudio, fontes e primitivas.
- Carrega a sprite dos aliens, as fontes e os sons do jogo (música e efeito de colisão).
- Registra os eventos do teclado e da tela.
- Se algo falhar, a função mostra uma mensagem de erro e encerra o programa com segurança.

As funções **show_start_screen** e **show_end_screen** exibem a tela inicial com o nome do jogo e a mensagem de "aperte qualquer tecla", e a tela final com a pontuação, o recorde e a opção de sair do jogo.

allocationFunctions.c

Este arquivo contém funções responsáveis por gerenciar a alocação dinâmica da matriz de aliens do jogo.

alloca_alien: aloca memória para uma matriz bidimensional de structs Alien, usada para representar os inimigos em cada fase. Usa o return para mostrar se a alocação funcionou..

free_alien: libera a memória alocada nos aliens.

Constants.c

Nesse arquivo estão todas as constantes usadas no jogo, por meio dele é possível alterar a velocidade, tamanho, e comprimento de qualquer elemento. Facilitando as trocas e a legibilidade do código.

creatingInvadersElements.c

Este arquivo define funções para inicializar e desenhar elementos do jogo, incluindo:

- **Nave** (initNave, draw_nave)
- **Aliens** (initAlien, initAllAliens, draw_alien, drawAllAliens)
- **Cenário** (draw_scenario)
- **Tiro** (init_shots, fire_shot, draw_shots)

Cada função configura a posição, aparência e estado dos objetos ou os desenha na tela usando a biblioteca Allegro.

fileFunctions.c

Este arquivo contém funções que leem e atualizam o recorde do jogo em um arquivo de texto chamado record.txt:

- **pega_recorde():** abre o arquivo e lê o valor do recorde.
- **adiciona_recorde():** sobrescreve o recorde se a pontuação atual for maior.

As duas funções usam FILE * e verificam se a leitura do arquivo teve algum erro.

Invaders.h

É o arquivo de cabeçalho, onde todas as funções e constantes são armazenadas, e as structs do tiro, nave, alien declaradas.

updatingInvadersElements.c

Este código implementa a lógica de movimentação e colisão no jogo, da seguinte forma:

- **update_nave():** Move a nave para esquerda ou direita conforme o usuário pressiona as teclas.
- **update_all.aliens():** Move todos os aliens e inverte sua direção caso toquem as bordas da tela, descendo uma linha.
- **colisao_alien_solo / colisao_all_alien_solo():** Verifica se algum alien chegou ao solo e finaliza o jogo se isso ocorrer.
- **colisao_alien_nave() / colisao_all_alien_nave():** Detecta colisões entre aliens e a nave do jogador.
- **update_shots():** Atualiza a posição do tiro e desativa se sair da tela.
- **shot_hit():** Detecta se o tiro atingiu algum alien. Se sim, desativa o alien e o tiro, aumenta a pontuação e toca o som de colisão.

Funções Implementadas

Obrigatórias

O jogo cumpriu com todas as recomendações de implementação que do trabalho, implementando as seguintes funcionalidades:

- Controle preciso dos movimentos da nave
- O grupo de naves deve ter ao menos 4 linhas e 5 colunas
- Deve haver ao menos 30 pixels de espaço entre as naves

- As naves devem se mover juntas, conforme o comportamento do jogo original
- O jogo deve permitir apenas um tiro por vez
- O tiro deve eliminar a nave ao colidir com ela
- Se qualquer nave tocar o solo, o jogo deve terminar
- Se qualquer nave colidir (ou estiver muito próxima) do canhão, o jogo deve terminar
- O tiro deve ser vertical, sem mudança de direção ou angulação
- O cenário deve exibir a pontuação do jogador
- Exibição e armazenamento do recorde

Opcionais

Além disso, o jogo também conta com as seguintes funcionalidades opcionais

- Tela de início.
- Tela de fim
- Fases com dificuldade aumentada
- Sprites de aliens e efeitos visuais
- Sons de colisão
- Música de fundo