

UNIVERSIDADE FEDERAL DE MINAS GERAIS

VINICIUS TRINDADE DIAS ABEL

TRABALHO PRÁTICO - PDS I

Documentação - Space Invaders

Belo Horizonte - MG
2020

Sumário

INTRODUÇÃO	3
OBJETIVO DO JOGO	3
COMO JOGAR	3
FUNCIONAMENTO DO CÓDIGO	3
STRUCT FUNÇÕES E PROCEDIMENTOS	4

INTRODUÇÃO

Este documento tem por finalidade explicar o jogo Space Invaders, desenvolvido como trabalho prático para a disciplina de Programação e Desenvolvimento de Software I. Será apresentado o objetivo do jogo, como jogá-lo e o funcionamento do código.

OBJETIVO DO JOGO

Space Invaders retrata um cenário onde aliens tentam invadir a Terra, e o objetivo do jogador é impedir que esta invasão seja bem-sucedida. Para isto, ele controlará uma nave, que poderá realizar um disparo por vez, quando o tiro da nave atinge um alien, ou vai para o espaço (sai da tela), o jogador pode realizar um novo disparo. O jogador precisa acertar cada alien apenas uma vez para eliminá-lo, e ao acertar todos os invasores, ele vence o jogo. Entretanto, se algum alien conseguir atingir a Terra ou a nave, o grupo alienígena vence, uma vez que concluiu seu objetivo de invadir a Terra.

COMO JOGAR

Para controlar a nave, o jogador pressiona a tecla 'A' para movimentá-la para esquerda e a tecla 'D', para direita. O disparo do tiro da nave é realizado ao pressionar a tecla 'ESPAÇO', sendo que só poderá ser feito um disparo por vez, como dito anteriormente.

Caso o jogador feche o jogo antes de eliminar todos os aliens, será considerado desistência, e, conseqüentemente, uma vitória dos aliens.

FUNCIONAMENTO DO CÓDIGO

Além da biblioteca Allegro, o código do jogo conta com um '#define' e diversos 'const' que facilitam o seu desenvolvimento e permite que sejam feitas alterações, como a velocidade ou tamanho de algum elemento, sem muitas complicações.

Ele foi estruturado com alguns 'struct', varias funções e procedimentos que deixam a leitura e entendimento, não só da função 'main', mas de todo o código, mais fácil. Estas funções serão melhor detalhadas a seguir.

STRUCT FUNÇÕES E PROCEDIMENTOS

`typedef struct Nave` - Este struct da nave define um tipo de variável Nave com velocidade, posição e cor.

`typedef struct Alien` - Este struct dos aliens define um tipo de variável Alien com velocidade, posição, cor e um int vivo para dizer se o alien foi atingido ou não.

`typedef struct Tiro` - Este struct do tiro define um tipo de variável Tiro com velocidade, posição, cor e um int up para dizer se o tiro foi disparado.

`void initNave(Nave *nave)` - Este procedimento inicializa a nave, preenchendo-a com os dados iniciais do jogo.

`void initAliens(Alien aliens[n][m])` - Este procedimento inicializa todos os aliens (são uma matriz), preenchendo-os com os dados iniciais do jogo.

`void initTiro(Tiro *tiro, Nave *nave)` - Este procedimento inicializa o tiro, preenchendo-o com os dados iniciais do jogo. Ele é inicializado fora da tela, porque ainda não foi acionado e não quero que seja inicializado na mesma posição de algum alien, e por ainda não ter sido acionado pelo jogador, não tem problema estar fora da tela.

`void draw_scenario()` - Este procedimento desenha o cenário, que é a grama/Terra.

`void draw_nave(Nave nave)` - Este procedimento desenha a nave que o jogador vai controlar.

`void draw_alien(Alien aliens[n][m], int i, int j)` - Este procedimento desenha a matriz de aliens que tentarão invadir a Terra.

`void draw_tiro(Tiro *tiro)` - Este procedimento desenha o tiro da nave.

`int colisao_alien_solo (Alien aliens[n][m])` - Esta função verifica se houve uma colisão entre algum alien e o solo.

`int colisao_alien_nave (Alien aliens[n][m], Nave nave)` - Esta função verifica se houve uma colisão entre algum alien e a nave.

`int colisao_tiro_alien (Tiro *tiro, Alien aliens[n][m])` - Esta função verifica se houve uma colisão entre o tiro e algum alien, caso a colisão ocorra, este alien é eliminado e mandado ao espaço (fora da tela), para uma distância segura em que não poderá voltar a atacar.

`void update_nave (Nave *nave)` - Este procedimento atualiza a nave, é ele que permite que a nave se mova.

`void update_alien(Alien aliens[n][m])` - Este procedimento atualiza os aliens, é ele que implementa o movimento dos aliens.

`void update_tiro(Tiro *tiro)` - Este procedimento atualiza o tiro, é ele que chama o procedimento para desenhá-lo e possibilita o movimento do tiro.

`int modificaRecorde(char nome_arquivo[], int cont)` - Esta função é responsável por atualizar o recorde quando o jogador consegue uma pontuação mais alta do que alguém já havia conseguido.

`int main()` - Esta é a função principal, é ela quem faz o jogo funcionar, criando uma tela e um temporizador, inicializando módulos allegro, instalando mouse e teclado, e também, é responsável pelas filas de eventos. Além destas responsabilidades, é ela quem cria os aliens, a nave e o tiro, e chama as outras funções na hora certa, dando vida ao jogo. Ela possui um 'while (playing)' que manda desenhar e atualizar todos os elementos do jogo até os aliens ou o jogador vencer, além de verificar os comandos do jogador, para movimentar a nave ou atirar. Quando o alguém vence (aliens ou jogador), a função main manda atualizar o recorde e depois atualiza a tela mostrando o resultado e o recorde atual. Por fim, após mostrar o resultado, a função main fecha o jogo.