**DOCUMENTAÇÃO – SPACE INVADERS**

Nome: Gabriel Martins Medeiros Fialho

Disciplina: Programação e Desenvolvimento de Software I

Professor: Pedro O. S. Vaz de Melo

**1 – O JOGO**

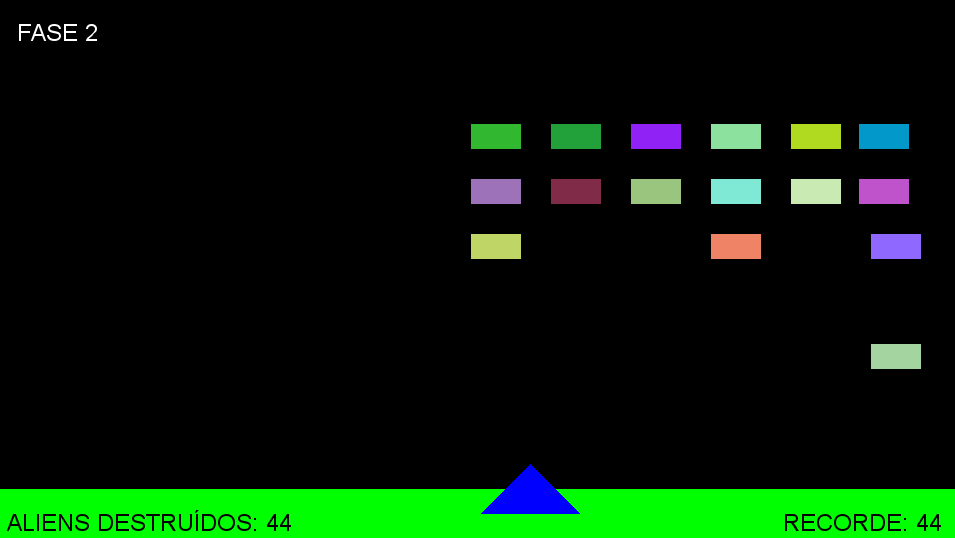
Space Invaders é um clássico jogo de arcade, um dos primeiros jogos de tiro com gráfico bidimensional. O principal objetivo é destruir naves alienígenas que estão atacando o planeta Terra. O Trabalho Prático sugerido propôs o desenvolvimento de um jogo semelhante, com o mesmo objetivo principal. No jogo, o usuário tem a oportunidade de controlar a nave de batalha que pode salvar o planeta Terra da invasão alienígena, podendo controlá-la com o teclado:

A – Enquanto pressionado, move a nave para a esquerda.

D – Enquanto pressionado, move a nave para a direita.

Espaço – Atira um projétil, que se move verticalmente, e sem angulação.

Se o projétil acertar alguma nave alienígena, a mesma é eliminada do jogo, o contador de pontuação aumenta em 1, e um novo tiro poderá ser disparado, caso contrário, um novo tiro somente poderá ser disparado quando o projétil terminar de percorrer todo seu trajeto e sair do campo de visão do jogador. O jogo acaba após cinco ondas de alienígenas serem destruídas, ou seja, após 170 alienígenas serem atingidos pelos projéteis, e imprime, no centro da tela, uma mensagem de sucesso. Caso uma nave toque o solo ou se aproxime muito do canhão, o jogo exibe uma tela de “game over” com informações que mostram se o jogador bateu o recorde e a pontuação atingida na sessão atual, além de conceder a opção de o usuário pressionar uma tecla para jogar novamente desde a primeira fase.

****

**2 – O CÓDIGO**

Linhas 1 a 8:

Inclusões das bibliotecas necessárias para o funcionamento do jogo, incluindo a biblioteca “unistd.h” para o uso da função “sleep”.

Linhas 10 a 46:

Declarações de variáveis globais, sendo que algumas podem ter o valor alterado de acordo com a fase.

Linhas 48 a 64:

Struct da nave, com sua posição x, cor, velocidade e direção do movimento. Struct do alien, com sua posição x e y, velocidade horizontal e vertical, cor e valor de existência. Struct do tiro, com sua posição inicial x e y, velocidade vertical, cor e valor de existência.

Linhas 65 a 190:

Procedimentos de inicialização da nave, dos aliens, do tiro; Procedimentos de desenho do cenário, da nave, dos aliens e do tiro; Procedimentos de atualização da posição e do valor de existência da nave (não há valor de existência), dos aliens e do tiro; Procedimento de colisão do tiro com o alien; Funções de colisão do alien com o solo e com a nave; Função de contagem do número de aliens existentes.

Linhas 191 a 584:

Função Main

**A FUNÇÃO MAIN:**

**A função main é chamada quando o jogo é iniciado, ou seja, na fase 1, ou quando o jogador avança para a próxima fase, de forma que novos valores sejam atribuídos a algumas variáveis a fim de dificultar o extermínio dos alienígenas de acordo com a fase atual.**

Linhas 192 e 193:

Declaração da variável “playing” com valor 1 e atribuição do valor 0 à variável “vitória”, para que, sempre que a função main seja chamada (quando é necessário reiniciar o jogo), a verificação de que o usuário está jogando e ainda não venceu a fase atual ocorra tranquilamente.

Linhas 195 a 275:

Estrutura de condições que alteram o valor de algumas variáveis, que dificultam gradativamente o jogo de acordo com a fase em que o jogador está.

Linhas 276 a 366:

Procedimentos de inicialização do allegro, da leitura de ponteiros de fonte e declaração de um ponteiro tipo FILE para abertura do arquivo “recorde.txt”.

Linhas 368 a 374:

Tiro, nave e vetor de aliens declarados de acordo com seus respectivos tipos de dados compostos; Inicialização da nave e dos aliens (inicialização do tiro não é feita nesse momento, mas quando o usuário pressiona espaço).

Linhas 376 a 380:

Faz a leitura do recorde armazenado no arquivo “recorde.txt” somente se a fase for a primeira, pois, caso contrário, o recorde mostrado após a primeira fase pode não ser referente ao recorde anterior ao jogo ser aberto.

Linha 383:

Inicializa o temporizador do allegro, que define a frequência do disparo de eventos do tipo timer.

Linha 384: Loop principal do jogo, com a condição de a variável “playing” ter valor 1.

**INÍCIO DO LOOP**

Linhas 386 e 387:

Inicialização da fila de eventos a serem processados.

Linha 389:

Início da estrutura de condições que têm como parâmetros os eventos allegro da fila de eventos criada anteriormente:

SE O EVENTO ALLEGRO FOR UM TIMER:

Linha 392: Chamada do procedimento que desenha o cenário.

Linha 394: Chamada do procedimento que atualiza a posição da nave.

Linha 396: Chamada do procedimento que atualiza a posição dos aliens.

Linha 398: Chamada do procedimento que atualiza a posição do tiro.

Linha 400: Chamada do procedimento que desenha a nave na posição atual.

Linha 402: Chamada do procedimento que desenha os aliens em suas respectivas posições atuais.

Linha 404: Chamada do procedimento que desenha o tiro na posição atual.

Linha 406: Chamada do procedimento que verifica a colisão do tiro com o alien e atualiza seus valores de existência se necessário.

Linhas 410 e 412: Imprimem na tela a quantidade de aliens destruídos e a fase atual.

Linha 414: Atualiza a variável “playing” para que, se a função de colisão dos aliens com o solo ou com a nave retorne 0, “playing” receba o valor 0 e o jogo acabe.

Linhas 416 a 424: Leitura do arquivo “recorde.txt”. Caso a pontuação atual seja maior que o recorde, reescreve o arquivo com a pontuação atual.

Linha 426: Escreve no canto inferior direito da tela o recorde atual.

Linha 428: Função “al\_flip\_display()” necessária para que a tela atualize com os procedimentos feitos.

Linhas 430 a 433: Condição que conta e imprime os segundos passados no prompt de comando.

Linhas 434 a 440: Condição que altera o valor das variáveis “vitoria” para 1 e “playing” para 0 caso todos os aliens da fase atual sejam destruídos. Caso a fase atual seja a última, também altera o valor da variável “vitoriafinal” para 1, que serve como parâmetro para exibir a mensagem de que todos os alienígenas foram exterminados e fecha o jogo após alguns segundos.

Linhas 443 a 446:

SE O EVENTO ALLEGRO FOR UM CLIQUE NO BOTÃO DE FECHAR

Altera o valor das variáveis “fechar” para 1 e “playing” para 0, para que a janela feche instantaneamente.

Linhas 447 a 463:

SE O EVENTO ALLEGRO FOR PRESSIONAR UMA TECLA

Imprime no prompt de comando o código da tecla; Inicia a movimentação da nave para a esquerda caso a tecla pressionada seja a tecla “A” ou inicia a movimentação da nave para a direita, caso a tecla pressionada seja a tecla “D” ou chama o procedimento da inicialização do tiro caso a tecla pressionada seja “ESPAÇO”.

Linhas 465 a 480:

SE O EVENTO ALLEGRO FOR SOLTAR UMA TECLA

Imprime no prompt de comando o código da tecla; Interrompe a movimentação da nave para a esquerda caso a tecla levantada seja a tecla “A” ou interrompe a movimentação da nave para a direita, caso a tecla levantada seja a tecla “D”.

Linhas 482 a 581: **FIM DO LOOP**

**Estruturas de condições para decidir o que será feito após o loop ser quebrado, ou seja, quando a variável “playing” estiver com o valor igual a 0.**

Linhas 482 a 485:

Se a variável “fechar” tiver valor igual a 1, significa que o usuário clicou no botão de fechar, então o programa desaloca os ponteiros do display e da fila de eventos.

Linhas 486 a 508:

Se a variável “vitoriafinal” tiver valor igual a 1, significa que o usuário finalizou a última fase. Os códigos das linhas 487 a 504 fazem a tela piscar de branco e azul por 2 segundos, e os códigos das linhas 505 a 507 atualizam a tela com a mensagem “TODOS OS ALIENÍGENAS FORAM DESTRUÍDOS!!!” e seguram a exibição da tela com essa mensagem por 5 segundos.

Linhas 509 a 549:

Se a variável “vitoria” tiver valor igual a 0, significa que o usuário não conseguiu completar a fase atual. Os códigos das linhas 510 a 515 piscam a tela de branco e vermelho por um segundo, e os códigos das linhas 516 a 529 escrevem no centro da tela a frase em destaque “GAME OVER” e, logo abaixo, caso o jogador tenha batido o recorde, a mensagem “Você fez <pontuação> pontos e bateu o recorde de <recorde> pontos!!!” ou, caso o jogador não tenha batido ou igualado o recorde, a mensagem “Sua pontuação <pontuação> Recorde: <recorde>”, além de, nas duas situações escrever abaixo a mensagem: “Pressione qualquer tecla para jogar novamente”. Após isso, a execução é pausada por 2 segundos usando as funções sleep e al\_rest para que o usuário não reinicie o jogo acidentalmente ao pressionar uma tecla dentro de 2 segundos. Os códigos das linhas 530 a 549 criam uma nova fila de eventos e um loop de condições que, caso o evento seja um timer, descansa a tela por um frame, caso o evento seja o aperto de uma tecla, fecha a janela, redefine a variável global “fase” para o valor zero e chama a função main, o que fará o jogo reiniciar na primeira fase e, caso o evento seja o clique no botão de fechar, fecha a janela.

Linhas 550 a 581:

Se a variável “vitoria” tiver valor igual a 0, significa que o usuário completou a fase atual. Os códigos das linhas 551 a 556 piscam a tela de branco e azul por um segundo, e os códigos das linhas 557 a 561 escrevem no centro da tela a palavra em destaque “VITÓRIA!!!!!!!!” e, logo abaixo, a mensagem “Pressione qualquer tecla para jogar a próxima fase”. Após isso, a execução é pausada por 2 segundos usando as funções sleep e al\_rest para que o usuário não avance a fase acidentalmente ao pressionar uma tecla dentro de 2 segundos. Os códigos das linhas 562 a 580 criam uma nova fila de eventos e um loop de condições que, caso o evento seja um timer, descansa a tela por um frame, caso o evento seja o aperto de uma tecla, fecha a janela, incrementa 1 à variável global “fase” e chama a função main, o que fará o jogo iniciar na próxima fase e, caso o evento seja o clique no botão de fechar, fecha a janela.

Linhas 582 a 584:

Procedimento padrão de desalocamento dos ponteiros do display e da fila de eventos para que, ao chegar o fim da execução da função main, ou seja, o fim do jogo, a janela seja fechada.