UFMG

UNIVERSIDADE FEDERAL
DE MINAS GERAIS

DOCUMENTAÇÃO - TRABALHO PRÁTICO

DRIVING DRUNK IN THE SPACE

ALUNO: FELIPPE VELOSO MARINHO
DISCIPLINA: PROGRAMAÇÃO DE SOFTWARES
PROFESSOR: PEDRO OLMO STANCIOLI VAZ DE MELO

Belo Horizonte – MG

2022

SUMÁRIO

1 - Introdução

Este documento tem como objetivo servir como documentação para o trabalho prático final da disciplina de PDS I, onde deveríamos desenvolver um jogo eletrônico de gráfico semelhante às versões clássicas (antigas) da franquia R-type. RType é um videogame de fliperama de rolagem horizontal desenvolvido e lançado pela Irem em 1987.. Basicamente, o jogador deve se movimentar entre as fases desviando dos inimigos e cumprindo pequenos objetivos para seguir em frente tais quais: apertar botões, matar inimigos e etc. O jogo deveria ser desenvolvido em C utilizando a biblioteca Allegro e os conhecimentos adquiridos ao longo do curso.

2 - História do Jogo

Neste jogo você é um viajante espacial que está no ano de 2622 e após a derrota de seu time de futebol espacial do coração, acabou se embriagando e pegando a estrada… Mal sabia você que estava na contramão!!!

O Boss irá aparecer após um minuto. *“Aquilo é realmente um polvo mesmo ou você bebeu um pouco demais??? Será que os blocos são ônibus??? Os meteoros são carros?? Os jetskis são motos?? Os banhistas são pedestres???... Eéé enfim, mistérios… “*

**Título do Jogo*

3 - Instruções

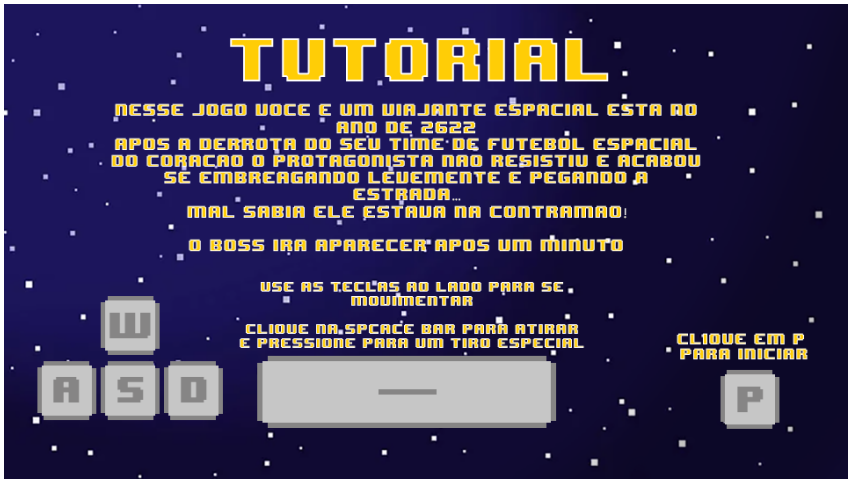
3.1 - Descrição

Em Driving Drunk in the Space seu objetivo é sobreviver e derrotar o chefe. Logo após a iniciação do jogo, abrirá uma tela de menu, onde serão apresentadas as opções de jogar e sair. Ao apertar “Sair” o jogo finaliza, ao apertar “Jogar”, o jogo abre uma tela de tutorial onde haverá as informações básicas para o jogo. Ao executar o jogador possui somente uma vida, isto é, “encostou, morreu”,. Todos os meteoros podem ser destruídos com o tiro básico que destrói um inimigo mas se destrói no processo, e o tiro avançado que ao segurar a barra de espaço por um segundo, é liberado um poderoso tiro de raio maior que só se destrói ao passar dos limites da tela. A cada meteoro destruído, o tamanho dele será adicionado a sua pontuação, ou seja, meteoros maiores, maiores pontuações. A pontuação é exibida na tela assim como o tempo que ao decorrer 1 minuto é liberado o chefe, aparecendo também seus pontos de vida. O chefe possui 15 pontos de vida, onde o tiro básico tira 1 pontos e o tiro avançado 4. Mesmo com o chefe em tela, os inimigos continuam aparecendo.

Ao derrotar o chefe é acrescentado 5000 pontos no score e exibido sua pontuação

final por 3 segundos, igualmente se o jogador for morto. Se for record, é exibido um texto com “Novo Recorde!”.

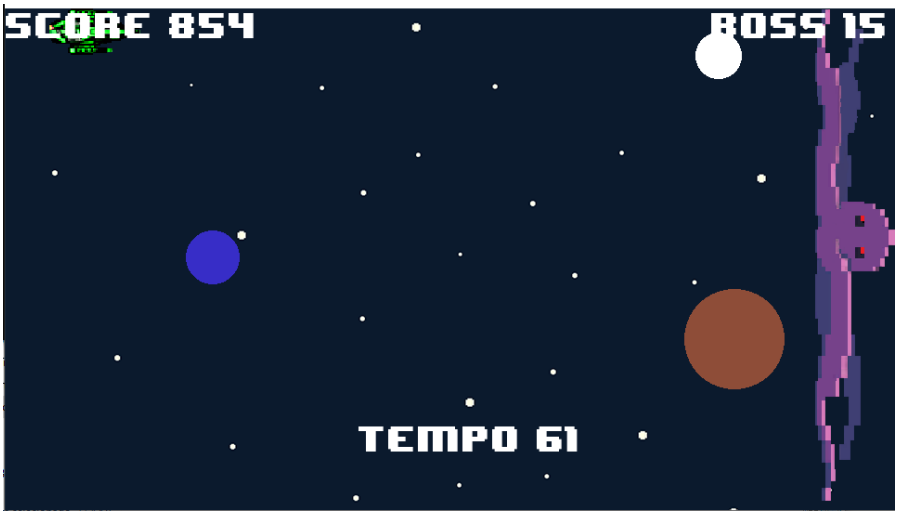
*Tela de Menu



*Tela de Tutorial

3.2 - Telas do Jogo

*Tela do Jogo em ação



*Tela do Jogo - Batalha com o Chefe

*Tela do Jogo - Exibição da Pontuação

3.3 - Controles

No menu e tutorial:

W, A, S, D ou SETA CIMA, SETA ESQUERDA, SETA DIREITA, SETA BAIXO - Seleciona as caixas de seleção do menu.
P - Inicia o jogo.

Em jogo:

W, A, S, D - Movimenta a Nave.
Clicando:
Espaço - Atira - Tiro básico.

Pressionando:
Espaço (1s) - Atira - Tiro especial.

4 - Descrição do Código

Linha 01 à 11: Includes das bibliotecas utilizadas no jogo.

Linha 13 a 32: Declaração das constantes e variáveis globais usadas no jogo.

Linha 36 a 49: Struct da Nave do jogador com campos de posição, velocidade, direção, vida, pontuação, cor e o bitmap para skin.

Linha 51 a 57: Struct do Bloco com campos de posição, largura, altura, e cor.

Linha 59 a 66: Struct do Meteoro com campos de posição, vida, velocidade, tamanho, e cor.

Linha 68 a 76: Struct do Tiro com campos de posição, vida, velocidade e tamanho.

Linha 78 a 88: Struct do Chefao com campos de posição, velocidade, tamanho, largura, altura, vida, cor e o bitmap para skin.

Linha 90 a 97: Struct do TiroC do chefe com campos de posição, velocidade, tamanho, vida e cor.

Linha 98 a 104: Struct das Estrelas do jogador com campos de posição e velocidade.

Linha 105 a 113: Função usada para números aleatórios.

Funções para Iniciar elementos do jogo

Linha 117 a 121: Função que inicia as variáveis globais.

Linha 123 a 132: Função que inicia as variáveis do Bloco.

Linha 133 a 142: Função que inicia as variáveis do Chefe.

Linha 143 a 155: Função que inicia as variáveis da Nave.

Linha 157 a 188: Função que inicia as variáveis das Estrelas, dentro de um laço que limita a quantidade de meteoros por vez na tela e atribui valores randômicos para posição, três tipos de iniciação com velocidades diferentes .

Linha 190 a 199: Função que inicia as variáveis do Meteoro, dentro de um laço que limita a quantidade de meteoros por vez na tela.

Linha 201 a 212: Função que inicia as variáveis do Tiro, dentro de um laço que limita a quantidade de tiros por vez na tela.

Linha 214 a 226: Função que inicia as variáveis do Tiro do Chefe, dentro de um laço que limita a quantidade de tiros por vez na tela.

Funções para Desenhar os Elementos do jogo

Linha 229 a 233: Função usada para desenhar o cenário.

Linha 235 a 241: Função usada para desenhar a Nave, passando o bitmap, e as posições da nave.

Linha 249 a 258: Função usada para desenhar o meteoro, dentro do laço para randomizar o tamanho de cada meteoro em tela, enquanto ele não for destruído.

Linha 260 a 268: Função usada para desenhar o tiro, dentro do laço para randomizar o tamanho de cada tiro em tela, enquanto ele não for destruído.

Linha 270 a 274: Função usada para desenhar o Chefe, passando o bitmap, e as posições do chefe.

Linha 276 a 298: Função usada para desenhar o plano de fundo, dentro do laço para randomizar o tamanho de cada estrela em tela, enquanto ela não for destruída, passando 3 as variedades de estrela.

Linha 300 a 308: Função que .

Funções que atualizam as variáveis declaradas nos structs.

Linha 312 a 317: Função que atualiza as coordenadas da nave, adicionando um valor multiplicado pela velocidade.

Linha 319 a 350: Função que atualiza as coordenadas das estrelas, subtraindo um valor multiplicado pela velocidade para simular um efeito de paralaxe e novamente dentro de um loop para randomizar os tamanhos das estrelas e definir a quantidade máxima em tela. A função possui 3 “if”, no quais servem para definir 3 tipos de velocidades diferentes. Aqui também é limitado a área das estrelas na tela.

Linha 352 a 363: Função que atualiza as coordenadas do tiro, adicionando um valor multiplicado pela velocidade definida nas funções de inicialização. Novamente possui um laço onde é definido a quantidade de tiros em tela por vez e enquanto o tiro não for destruído ele é atualizado. Aqui também é limitado a área dos tiros a tela.

Linha 364 a 376: Função que atualiza as coordenadas dos meteoros, subtraindo um valor multiplicado pela velocidade para simular um efeito de movimento e novamente dentro de um loop para randomizar os tamanhos e definir a quantidade máxima em tela. Enquanto o tiro não for destruído e estiver em tela ele é atualizado. Aqui também é limitado a área das estrelas na tela.

Linha 378 a 387: Função que atualiza as coordenadas do bloco, adicionando um valor multiplicado pela velocidade. Aqui também é limitado a área dos meteoros na tela.

Linha 389 a 400: Função que atualiza as coordenadas do tiro do chefe, adicionando um valor multiplicado pela velocidade definida nas funções de inicialização. Novamente possui um laço onde é definido a quantidade de tiros em tela por vez e enquanto o tiro do chefe não for destruído ele é atualizado. Aqui também é limitado a área dos tiros a tela.

Linha 403 a 419: Função que define o movimento dos meteoros e randomiza o “nascimento” de cada um na tela enquanto não for destruído.

Linha 421 a 454: Função define o movimento do tiro. Dentro do laço “for”, para limitar a quantidade de tiros em tela e se o tiro não for destruído, se a duração do tiro, definida por uma variável mais à frente do código, for maior ou igual a 1 segundo o tiro assume um tamanho de 20 de raio, senão assume 4. Ambos também são definidos como “vivos” na tela e são disparados a partir do ponto x,y da nave. No final também é aberto o arquivo de som e colocado para tocar o efeito de Laser Shot para cada tiro.

Linha 456 a 471: Função que define o movimento dos tiros do chefe. Dentro do laço “for”, para limitar a quantidade de tiros em tela e se o tiro não for destruído, ele é definido como “vivo” e tem seu nascimento definido randomicamente pelas suas coordenadas.

Linha 474 a 498: Função que prende a nave na tela. São utilizados 4 “if” que fazem que quando a nave chega na borda da tela suas coordenadas sejam atualizadas pra dentro da tela. Assim é definida a área do jogo.

Funções de colisão:

Linha 502 a 511: Colisão do tiro com o bloco. Dentro de uma laço “for”, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações do bloco com as coordenadas do desenho do tiro.
Se o tiro estiver nas áreas delimitadas pelos condicionais do bloco, o tiro é destruído.

Linha 512 a 523: Colisão do meteoro com o bloco. Dentro de uma laço “for”, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações do bloco com as coordenadas do desenho do meteoro.
Se o meteoro estiver nas áreas delimitadas pelos condicionais do bloco, o meteoro é destruído.

Linha 525 a 557: Colisão do tiro com o meteoro. Dentro dos laços “for”, com as limitações de quantidade de elementos por tela dos tiros e meteoros, enquanto os tiros e meteoros estiverem ativos, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas do desenho do meteoro com as coordenadas do desenho do tiro.
Se o tiro bater no meteoro, ambos são destruídos, é carregado e tocado o som de explosão e é adicionado ao score o tamanho do meteoro destruído.

Linha 559 a 586: Colisão de meteoro com meteoro. Dentro dos laços “for”, com as limitações de quantidade de elementos por tela dos meteoros, enquanto os meteoros estiverem ativos, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas dos desenhos dos meteoros.

Linha 588 a 631: Colisão do meteoro com o bloco e com a nave. Dentro do laço “for”, com as limitações de quantidade de elementos por tela dos meteoros, enquanto os meteoros estiverem ativos, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas dos desenhos da nave com os do meteoros. Se a nave invadir os limites das coordenadas de algum meteoro, a nave morre e é carregada e tocada a sample de som de morte. Para a morte da nave é retornado 1 da função para que o jogo possa ser finalizado na main. Na colisão com o bloco, se a nave invadir os limites das coordenadas de algum bloco, o som de morte é tocado, igualmente no caso anterior e também é retornado 1.

Linha 633 a 670: Colisão do chefe com os tiros e com a nave. Dentro do laço “for”, com as limitações de quantidade de elementos por tela dos meteoros, enquanto os meteoros estiverem ativos, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas dos desenhos da nave com os do chefe. Se a nave invadir os limites das coordenadas do chefe, a nave morre e é carregada e tocada a sample de som de morte. Para a morte da nave é retornado 1 da função para que o jogo possa ser finalizado na main. Na colisão com os tiros, se um tiro invadir os limites das coordenadas do chefe, se for tiro simples, a vida do chefe reduz em um ponto, se for tiro avançado a vida do chefe reduz em 4 pontos.

Linha 672 a 713: Colisão do tiro do chefe com o bloco e com a nave. Dentro do laço “for”, com as limitações de quantidade de elementos por tela dos tiros do chefe, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas dos desenhos da nave com os do meteoros. Se o tiro do chefe tocar na nave, a nave morre e é carregada e tocada a sample de som de morte. Para a morte da nave é retornado 1 da função para que o jogo possa ser finalizado na main. Na colisão com o bloco, se o tiro invadir os limites das coordenadas de algum bloco é retornado 1.

Linha 715 a 736: Colisão do meteoro com os tiros do chefe. Dentro do laço “for”, com as limitações de quantidade de elementos por tela dos tiros e meteoros, enquanto os meteoros e tiros estiverem ativos, é colocada a limitação dentro de condições do “if”, através de um cálculo das limitações das coordenadas dos desenhos dos tiros do chefe com os do meteoros. Se um tiro atingir um meteoro, o meteoro é desativado.

Funções de Score e Recorde

Linha 740 a 753: A função score basicamente escreve o SCORE na tela, com o uso da função da biblioteca allegro, "al draw_text".

Linha 755 a 765: Função para encriptar os dados de uma string.

Linha 767 a 779: Função que abre o arquivo em modo de leitura e verifica se há ou não um arquivo recorde, havendo, a função chama a função de encriptar e retorna o recorde salvo no arquivo.

Linha 781 a 790: Função que abre o arquivo em modo de escrever, chama a função de encriptar, grava o valor de x passado no parâmetro e fecha o arquivo.

Linha 793 a 805: Função que define se a vida do chefe chegar a zero é tocado o som de explosão, ele é desativado e é retornado 1, significando que você venceu o jogo.

Linha 806 a 809: A função score basicamente escreve a vida do chefe na tela, com o uso da função da biblioteca allegro, "al draw_text".

Abertura da main - função principal:

Linha 815: Faz com a semente da função rand varie com o tempo.

Linha 817 a 836: Declaração de variáveis da biblioteca allegro.

Linha 838 a 919: Inicialização das funções da biblioteca Allegro.

Linha 928 a 977: Inicialização das imagens usadas e conferência se todas foram carregadas.

Linha 979 a 1012: Inicialização dos parâmetros de som usados no programa, carregamento de algumas samples e registro de eventos.

Linha 1014 a 1035: Inicializações das funções usados no jogo e chamada dos structs. Definição do tamanho dos arrays de Tiros e TirosC ”tiros do chefe”;

Linha 1038: Inicia o temporizador.

Linha 1039 a 1058: Declaração de variáveis usadas enquanto há o evento de playing.

Enquanto o jogo está aberto - Menu:

Linha 1062 a 1083: Armazena a variável de evento após um tempo e se o cronômetro começar a variável menu, iniciada com o valor 1, abre a sample da música principal, desenha os menus e o botão.

Linha 1084 a 1169: Enquanto no menu, é declarada a variável de foiP ”foi play” e aberto o evento de Key_Down fazendo com que nos casos de Key_Up W, Key_Down e S a região do bitmap do botão seja alterada para outra. No caso, os bitmaps dos botões são desenhados de forma que tenha uma parte para simular o botão normal e outra para simular selecionado. Nos casos de Key_Down e S é ativada a variável continuar que define se o jogador selecionou sair ou jogar. No caso de Key_Enter é aberto uma tela com um pequeno tutorial e escrito se for apertado a tecla P, o jogo inicia.

Enquanto o jogo está aberto - Gameplay:

Linha 1173 a 1405: Com o jogo rodando, é iniciado um novo timer, usando a função **al_start_timer()** que servirá para contar o tempo do jogo. Após isso é chamado as funções de **desenhaCenario(), atualizaBloco(), desenhaBloco(), atualizaNave(), mantemNaveNaTela(), desenhaNave(), atualizaMeteoro(), desenhaMeteoro(), movimentaMeteoro(), atualizaTiro(), desenhaTiro(), score(), colisaoMeteoroBloco(),**

colisaoMeteoroMeteoro(), colisaoTiroBloco(), colisaoTiro() e passado os parâmetros requisitados por elas. É definido uma variável que através da função **al_get_timer_count()**, aumenta +1 a cada segundo, sendo assim se essa variável atingir 60 é liberado o chefe através das funções: **desenhaChefao(), printaChefeVida()**, as funções do tiro do chefe **desenhaTiroC(), atualizaTiroC(), movimentaTiroC()**. É colocada uma instância para tocar e definido que se o retorno da função **colisaoTiroC()** == 1, é tocada a sample de morte é exibida a tela final o mesmo acontece se as funções **colisaoMeteoro()** e **colisaoChefao()** == 1. A lógica da tela final basicamente se repete em todos os "if" que significam a morte da nave e através de sprintf, as função **al_draw_text()**, **getRecorde()** e **setRecorde**, se a pontuação, (definida pela variável "escore"), for maior que o recorde (lido no arquivo pela função **getRecord()**), é printado um novo recorde na tela é exibido a pontuação, caso seja menor só é exibido a pontuação por três segundo e o jogo fecha.
Semelhantemente acontece se o vilão é derrotado, porém agora também é printado "VOCÊ VENCEU" na tela.

Linha 1407 à 1410: Se o tipo do evento for o de fechamento de tela, Clicar no "X", o jogo é fechado

Linha 1412 à 1469: Se o tipo de evento for um pressionar de uma tecla, quando a tecla espaço, "KEY_SPACE", está apertada é capturado o valor do tempo com a variável iniciio_tiro, através da função **al_get_time(),** e a variável segurando_tecla é declarada como true. Quando está solta segurando_tecla é declarada como false, a variável duracao_tiro recebe o valor do tempo menos o valor de inicio_tiro e é chamada a função **atira()** passando nos parâmetros as variáveis descritas nesse parágrafo e os structs necessários.

Linha 1469 à 1483: Procedimentos de fim de jogo (fecha a tela, limpa a memória, etc).

Linha 1484: Fim do Programa.