UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA

Trabalho realizado por
Bruna Schmidt e
Larissa Furtado.

Desenvolvimento de Jogo com Raylib

Space Spelunker

Algoritmos e Programação

Professor: Joel Luis Carbonera

1 RESUMO

O projeto consiste em um jogo de computador implementado em linguagem C com auxílio da biblioteca Raylib em que um explorador espacial controlado pelo jogador deve minerar em cavernas misteriosas enquanto evita ser atacado por toupeiras mutantes.

2 DESENVOLVIMENTO

2.1 Elementos do jogo

Figura 1:Mineirador espacial (jogador).

Figura 2: Toupeira mutante.

Figura 3: Esmeralda.

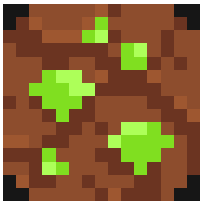

Figura 4: Ouro.

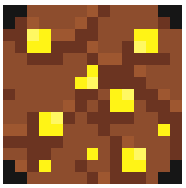

Figura 5: Powerup.

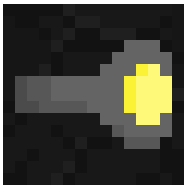

Figura 6: Pedra.

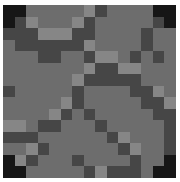

Figura 7: Terra.

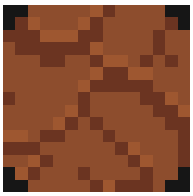

Figura 8: Projétil

2.2 Pastas

O jogo inclui uma pasta contendo o código, os mapas e os jogos salvos e uma pasta contendo os sprites responsáveis pelos elementos gráficos.

2.3 Elementos estáticos

O mapa foi implementado por meio de uma função chamada "desenhaMapa", que lê os elementos de uma uma matriz de caracteres previamente preenchida pela função "leMapa" a partir de um arquivo de texto, e desenha-os sequencialmente utilizando a função *Drawtexture* e dois laços *for* aninhados. Já o menu é ativado quando a variável "menu" se torna verdadeira, pausando simultaneamente o jogo, o qual encontra-se em um laço *while*, executando enquanto a variável for falsa.

2.4 Elementos animados

A animação do jogador foi implementada por meio da função pressionaTecla, que a partir da tecla pressionada atualiza os campos sentido e deslocamento em x e y da struct jogador, e da função podeMoverJ, que verifica se a nova posição corresponde a uma área livre ou bloqueada, retornando 1 ou 0, respectivamente. Já as toupeiras movem-se de acordo com a função podeMoverT, na qual um laço *for* checa se é possível o deslocamento de cada toupeira, e a função moveToupeira, em que a cada deslocamento um contador contaPassos é incrementado até chegar a 5, momento no qual outra direção é definida randomicamente pela função toupeiraRand.

.

2.5 Funções

2.5.1 carregaJogo

Recebe como parâmetros jogador, número de toupeiras, toupeira, número do mapa, número de esmeraldas e velocidade e uma matriz mapa. Carrega as informações de jogos previamente salvos em arquivos de texto.

2.5.2 salvaJogo

Recebe como parâmetros jogador, número de toupeiras,toupeira, número do mapa, número de esmeraldas e velocidade e a matriz mapa. É responsável por salvar o jogo em arquivo de texto.

2.5.3 pressionaTecla

Recebe como parâmetros jogador, tiro e menu. Detecta quando teclas de seta, tiro ou tab são pressionadas e implementa suas respectivas funcionalidades.

2.5.4 funcionaMenu

Recebe como parâmetro menu. É responsável por detectar quando teclas do menu são pressionadas e retornar seus respectivos caracteres.

2.5.5 tiroColisao

Recebe como parâmetros tiro, número de toupeiras, pontos, o vetor toupeiras e a matriz mapa. Verifica a posição do tiro em relação ao mapa e detecta se este colidiu com áreas soterradas ou toupeiras, atualizando-as para deixarem de ser exibidas e retornando 1 caso o tiro tenha encontrado um alvo.

2.5.6 sentidoTiro

Recebe como parâmetros tiro e jogador. É responsável por determinar a direção em que o tiro deverá se deslocar, tomando como base o sentido do último deslocamento realizado pelo jogador.

2.5.7 moveToupeira

Recebe como parâmetros toupeira e pode. Realiza o movimento aleatório das toupeiras com base em um contador de passos.

2.5.8 move

Recebe como parâmetros posição. É responsável por alterar a posição de uma toupeira ou jogador tendo como base o deslocamento recebido.

2.5.9 podeMoverT

Recebe como parâmetros posição, largura, altura e a matriz mapa. Checa se a posição ocupada por cada toupeira após o deslocamento será um bloco livre (terra ou área soterrada) ou um bloco ocupado (pedra), retornando 1 caso o deslocamento seja possível.

2.5.10 podeMoverJ

Recebe como parâmetros jogador, largura, altura, número de toupeiras,o vetor toupeiras e a matriz mapa.Checa se a posição ocupada pelo jogador após um deslocamento será um bloco livre e retorna 1 em caso positivo.

2.5.11 leMapa

Recebe como parâmetro jogador, número de toupeiras, número de esmeraldas, os vetores nome do arquivo e toupeiras, e a matriz mapa. É responsável por ler o conteúdo de um arquivo de texto e preencher a matriz mapa.

2.5.12 toupeiraRand

Recebe como parâmetro posição. Atribui um valor de -1 a 1 ao deslocamento de cada toupeira.

2.5.13 desenhaMapa

Recebe como parâmetro textura da toupeira, textura da pedra, textura da terra, textura do fundo, textura do powerup, textura da esmeralda e textura do ouro, e a matriz mapa. É responsável por desenhar o mapa na tela, já aplicando as respectivas texturas de cada caracter.

5 CONCLUSÃO

O jogo foi implementado com sucesso, visto que todas as mecânicas funcionam apropriadamente, possuindo jogabilidade adequada, além de apresentar texturas desenvolvidas manualmente para um enriquecimento da experiência do jogador.