

Lucas Moré e Henrique Martins  
INF01202

Prof: Marcelo Walter



**INFORMAÇÕES GERAIS(GAMEPLAY):**

O código é formado por várias estruturas.

Uma estrutura contém todas as características de um objeto.

```
struct estruturaElemento{
    Vector2 posicao;
    Vector2 posicaoInicial;
    int direcao;
    float tamanho_X;
    float tamanho_Y;
    Texture2D texturaAtual;
    Texture2D textura0;
    Texture2D textura1; //Facada
    Texture2D textura2; //Atirando
    Texture2D textura3; //Morrendo
    Texture2D textura0_0; //Esquerda
    Texture2D textura0_1; //Direita
    Texture2D textura0_2; //Cima
    Texture2D textura2_0; //Atirando - esquerda
    Texture2D textura2_1; //Atirando - direita
    Rectangle retangulo;
    Rectangle retanguloFacada;
    int situacao;
};
```

estruturaElemento

Todo objeto ou elemento tem seu retângulo, que é sua área de influência.

Para dois objetos se tocarem, seus retângulos devem sobrepor-se.

O uso do retângulo, apesar de não ser estritamente necessário em alguns casos, facilita o entendimento e a escrita do código.

A textura do jogador e dos inimigos é carregada na função inicialização.

Cada textura é formada a partir de um elemento imagem, reutilizado várias vezes.

Para o jogador, a textura 0 é sua figura normal, a 1 andando e a 2 atirando.

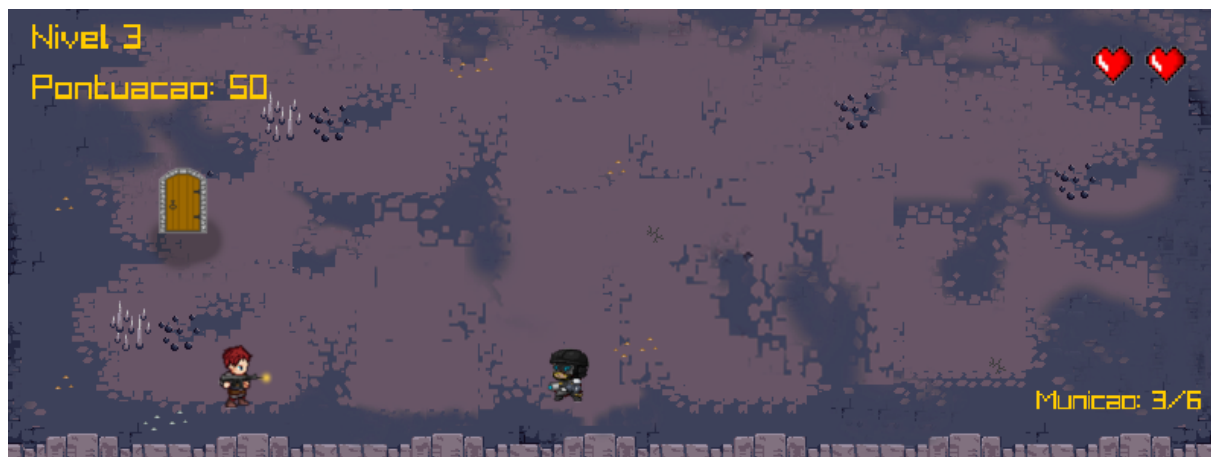
Para os inimigos, a textura 0 é a figura normal.

O movimento funciona em duas funções diferentes: para o jogador e para o inimigo.

Para o jogador, depende do pressionar das teclas direcionais.

Os inimigos se movem somente se houver diferença na posição horizontal ou vertical em relação ao jogador.

Além disso, o movimento inimigo somente ocorre em determinado intervalo de tempo.



Na função principal, é detectado se as teclas SPACE ou X estão pressionadas.

Se não, a textura normal é exibida.

As velocidades de movimento podem ser alteradas.

```
#define VELOCIDADEDOJOGADOR 1  
#define VELOCIDADEDOINIMIGO 0.5
```

As colisões (faca-inimigo, jogador-inimigo, jogador-porta) são analisadas pela função CheckCollisionRecs.

No caso das colisões jogador-inimigo, foi criada uma função colisoes(), pois neste caso o código necessário é mais longo e fica melhor para não poluir a função main().

## INFORMAÇÕES GERAIS(TELA):

O jogo é controlado pela utilização da “variável” estadoTela:

```
#define INTRO_TELA 0  
#define MENU_TELA 1  
#define NOVO_JOGO_TELA 2  
#define CARREGAR_JOGO_TELA 3  
#define RANKING_TELA 4  
#define TUTORIAL_TELA 5  
#define JOGO_TELA 6  
#define INPUT_NOME_TELA 7  
#define FIM_DE_JOGO_TELA 8  
#define VOCE_VENCEU_TELA 9
```

Que na verdade são definições, para controlar justamente o estado da tela:

O estado da tela muda dependendo das ações do usuário, um clique num botão do menu por exemplo, o leva a outra página e assim por diante.

Inicialmente o estado da tela é = 0, ou seja, INTRO\_TELA (tela de introdução):

A introdução que se vê no início do jogo foi inspirada por exemplos da biblioteca Raylib, e utiliza de um contador de frames, um contador de letras e também conta com posições e com as bordas dos retângulos a serem animados.

```
//Variáveis Intro-----
int logoPosicaoX = screenWidth/2 - 200;
int logoPosicaoY = screenHeight/2 - 120;

int framesIntroCont = 0;
int letrasCont = 0;

int bordaSuperiorComprimento = 16;
int bordaEsquerdaAltura = 16;
int bordaInferiorComprimento = 16;
int bordaDireitaAltura = 16;

int estadoIntro = 0;
float alpha = 1.0f;
float alpha2 = 1.0f;
```

Através da variável estadoIntro, a função configuracaoIntro faz suas tarefas por partes, com o final de cada etapa levando à próxima por meio da alteração dessa variável.

Dentro do loop principal essa função é chamada, para que seja atualizada.

Posteriormente, a função animacaoIntro, utiliza da variável estadoIntro definida e ajustada conforme a função configuraçãoIntro para funcionar.

A função animacaoIntro desenha então os retângulos e as letras sucessivamente, também utilizando de diversas variáveis já primeiramente usadas em configuracaoIntro para executar corretamente (como por exemplo as variáveis alpha, que controlam o efeito de “fade” das texturas e letras).

É chamada dentro da parte de desenho, para que possa cumprir seu dever.

A partir daí, quando o estado da introdução chega ao final, o estado da tela muda para 1, ou seja, MENU\_TELA:



Onde o jogador escolhe a opção através de um clique no seu mouse.

Os botões do jogo foram feitos como um conjunto de C e PhotoShop (para a parte visual).

Criamos retângulos na Raylib que possuem a posição e as dimensões dos retângulos mostrados na tela, o funcionamento do clique desses retângulos depende novamente do estado da tela. E os botões em si, alteram esse estado.



Exemplo: tela do tutorial e a configuração de seu botão

```
void configuracaoTutorial(Sound fxButton){  
    if(btnAcao && CheckCollisionPointRec(mousePosicao, btnVoltar))  
    {  
        PlaySound(fxButton);  
        estadoTela=MENU_TELA;  
    }  
}
```

As outras funções do jogo atuam de forma análoga às explicadas acima. Em geral sendo do tipo void, e sendo chamadas dentro do loop e da parte de desenho para que possam ser atualizadas e exibidas.