A Mimir avanços 4

- Menu Aprimorado (botões selecionáveis).
- Sprites do Player, Botões e Texturas.
- Tela de Game Over mais interativa!

Menu novo (uau!)

*menu alterado (tecla enter retirada)

defs.h

```c
typedef enum {
    MENU_BUTTON_PLAY,
    MENU_BUTTON_OPTIONS,
    MENU_BUTTON_QUIT,
    MENU_BUTTON_COUNT // Usado para saber quantos botões temos
} MenuButton;

typedef enum {
    END_BUTTON_PLAYAGAIN,
    END_BUTTON_MENUAGAIN,
    END_BUTTON_COUNT
} EndButton;

#endif // DEFS_H
```

menu.c

```c
#include "menu.h"
#include <SDL2/SDL_image.h>
#include <stdio.h>
#include "text.h"

// Texturas
static SDL_Texture* texFundoMenu = NULL;
static SDL_Texture* texSplashNome = NULL;
static SDL_Texture* texMenuTitulo = NULL;

// Texturas para o estado NORMAL
static SDL_Texture* texBotaoJogar = NULL;
static SDL_Texture* texBotaoOpcoes = NULL;
static SDL_Texture* texBotaoSair = NULL;

// Texturas para o estado SELECIONADO
static SDL_Texture* texBotaoJogar_H = NULL;
static SDL_Texture* texBotaoOpcoes_H = NULL;
static SDL_Texture* texBotaoSair_H = NULL;

static MenuButton botaoSelecionado = MENU_BUTTON_PLAY;

// Posições
static SDL_Rect rectFundoMenu;
static SDL_Rect rectSplashNome;
static SDL_Rect rectMenuTitulo;
static SDL_Rect rectBotaoJogar;
static SDL_Rect rectBotaoOpcoes;
static SDL_Rect rectBotaoSair;
```

```c
void menu_init(SDL_Renderer *renderer) {
    // Carrega Asset do Fundo ---
    texFundoMenu = carregarTextura("assets/fundo_menu_sprite.png", renderer);

    // O rect de fundo cobre toda a tela
    rectFundoMenu.x = 0;
    rectFundoMenu.y = 0;
    rectFundoMenu.w = SCREEN_WIDTH;
    rectFundoMenu.h = SCREEN_HEIGHT;

    // Carrega assets da Splash
    texSplashNome = carregarTextura("assets/splash_nome_jogo.png", renderer);

    // Carrega assets do Menu
    texMenuTitulo = carregarTextura("assets/menu_titulo.png", renderer);
    texBotaoJogar = carregarTextura("assets/menu_botao_jogar.png", renderer);
    texBotaoOpcoes = carregarTextura("assets/menu_botao_opcoes.png", renderer);
    texBotaoSair = carregarTextura("assets/menu_botao_sair.png", renderer);

    texBotaoJogar_H = carregarTextura("assets/menu_botao_jogar_H.png", renderer);
    texBotaoOpcoes_H = carregarTextura("assets/menu_botao_opcoes_H.png", renderer);
    texBotaoSair_H = carregarTextura("assets/menu_botao_sair_H.png", renderer);

    // --- Define Posições (usando o tamanho real das texturas) ---
    int texW, texH;
```

render.c

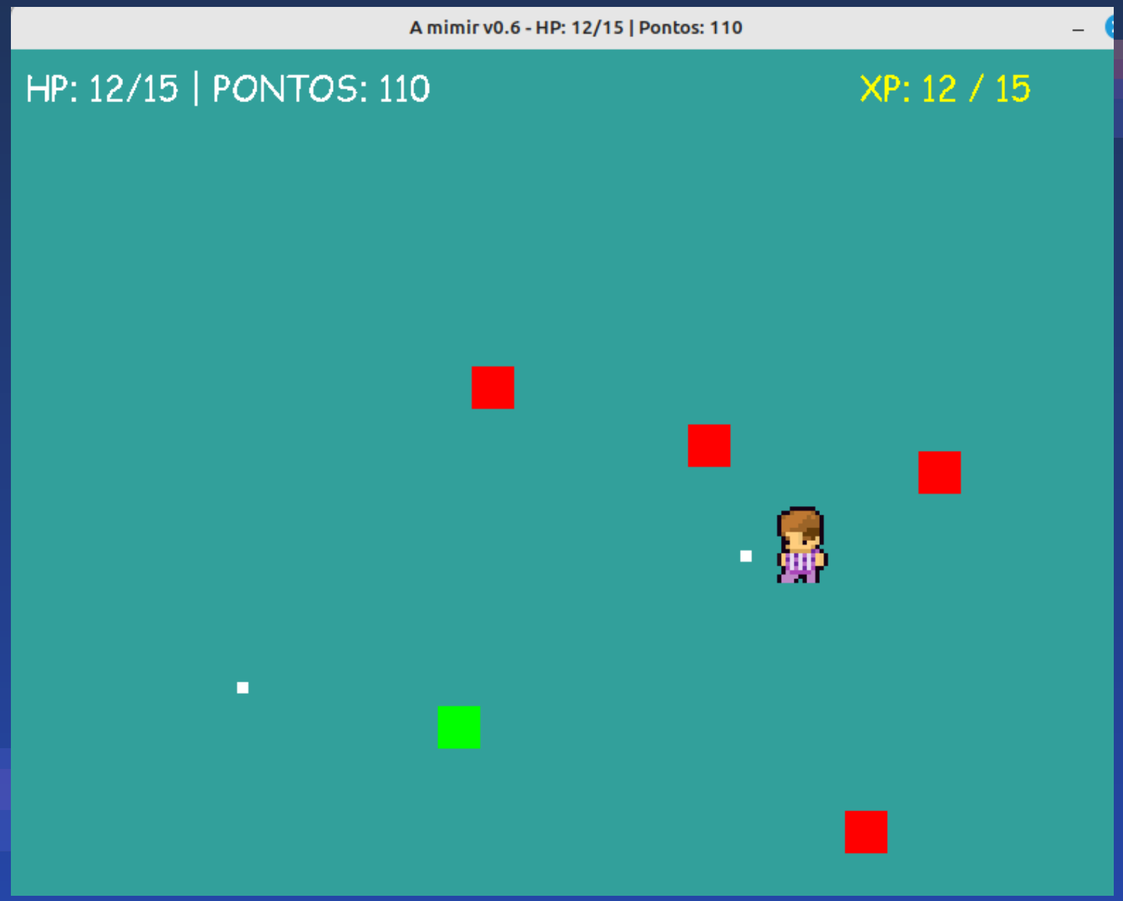
```c
// Desenha o Jogo (STATE_PLAYING ou STATE_LEVELUP) ---
if (currentState == STATE_PLAYING || currentState == STATE_LEVELUP) {

  // Limpa a tela (fundo azul)
  SDL_SetRenderDrawColor(renderer, 50, 161, 157, 255);
  SDL_RenderClear(renderer);

  // Desenha o Jogador
  if (player.texture != NULL) {
      Uint32 now = SDL_GetTicks();
      SDL_Rect srcRect;
      srcRect.x = player.frameX;
      srcRect.y = 0; // sheet horizontal
      srcRect.w = 32;
      srcRect.h = 32;

      if (player.invulnerable) {
          // lógica de piscar
          if ((now / 200) % 2 == 0) {
              SDL_RenderCopy(renderer, player.texture, &srcRect, &player.rect);
          }
      } else {
          SDL_RenderCopy(renderer, player.texture, &srcRect, &player.rect);
      }
  } else {
      // Fallback: Se o sprite não carregou, desenha o retângulo roxo padrão
      SDL_SetRenderDrawColor(renderer, 128, 0, 128, 255);
      SDL_RenderFillRect(renderer, &player.rect);
  }
```

```c
// --- 2. Desenha o Menu (STATE_MENU) ---
else if (currentState == STATE_MENU) {
  menu_render(renderer);

  if (menu_is_in_splash()) { // função auxiliar em menu.c/h
    Uint32 now = SDL_GetTicks();
    if ((now % 1000) < 500) { // pisca a cada 1 segundo (500ms on, 500ms off)
      const char *flashText = "Pressione qualquer tecla para continuar";
      drawText(renderer, app->font, flashText, 200, 400, white);
    }
  }
}

  // --- 3. Desenha o Fim de Jogo (STATE_END) ---
else if (currentState == STATE_END) {
  fim_render(renderer, player, app);
}
```

main.c

```
// Se acabamos de sair do menu e entrar no jogo...
if (currentState == STATE_PLAYING && (previousState == STATE_MENU || previousState == STATE_END)) {
    // ... inicializa/reseta o jogo
    initPlayer(&player, renderer);
    initEnemies(enemies);
    initBullets(bullets);
    initEnemyBullets(enemyBullets);
    lastEnemySpawnTime = SDL_GetTicks();
}

//Se saímos do Fim de Jogo/Placar e voltamos ao menu...
if ((previousState == STATE_END || previousState == STATE_SCOREBOARD) && currentState == STATE_MENU) {
    menu_init(renderer);
}

// 3. Atualizar Estado do Jogo
if (currentState == STATE_PLAYING) {

    handlePlayingInput_State(&player);

    update(&player, enemies, bullets, enemyBullets, &running, &lastEnemySpawnTime, &currentState);
}
```

Tela de Fim de Jogo



*botões provisórios para testes

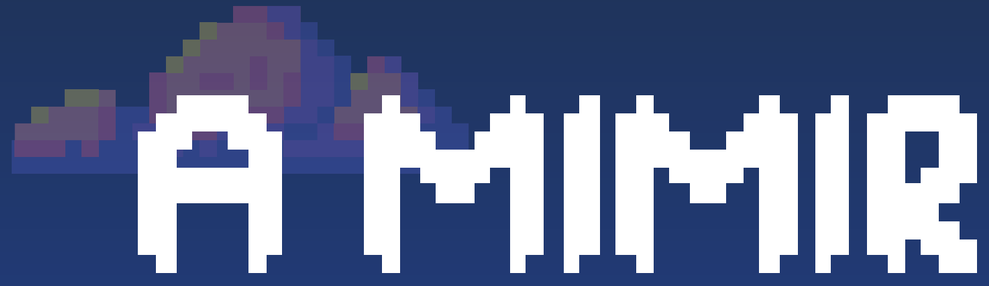
fim.c

```c
static void executar_acao_botao(GameState *currentState) {
    switch (botaoSelecionado) {
        case END_BUTTON_PLAYAGAIN:
            *currentState = STATE_PLAYING;
            break;
        case END_BUTTON_MENUAGAIN:
            *currentState = STATE_MENU;
            printf("Voltando ao Menu...\n");
            break;
    }
}
```

```c
void fim_render(SDL_Renderer *renderer, Player player, App *app) {
    // Desenha o Sprite de Fundo ---
    if (texFundoFim != NULL) {
        SDL_RenderCopy(renderer, texFundoFim, NULL, &rectFundoFim);
    } else {
        // Caso o sprite falhe, cor de fundo padrão para garantir que a tela limpe
        SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
        SDL_RenderClear(renderer);
    }

    // Desenha a tela de GameOver
    SDL_RenderCopy(renderer, texSplashGameOver, NULL, &rectSplashGameOver);

    // Desenha a pontuação atingida
    if (app -> font) {
        char hudText[150];
        int textW, textH; // armazena largura e altura do texto

        sprintf(hudText, "Voce fez %d pontos!", player.points);

        TTF_SizeText(app -> font, hudText, &textW, &textH);
        // tamanho que o texto ocupará na tela

        int xCentral = (SCREEN_WIDTH - textW)/2;

        drawText(renderer, app -> font, hudText, xCentral , 185, whitef);
    }
```

Sprites Implementados

- Menu Inicial



- Player

- GameOver

Sprites Implementados

- Botões

Sprites Implementados

- Mais botões

Fim