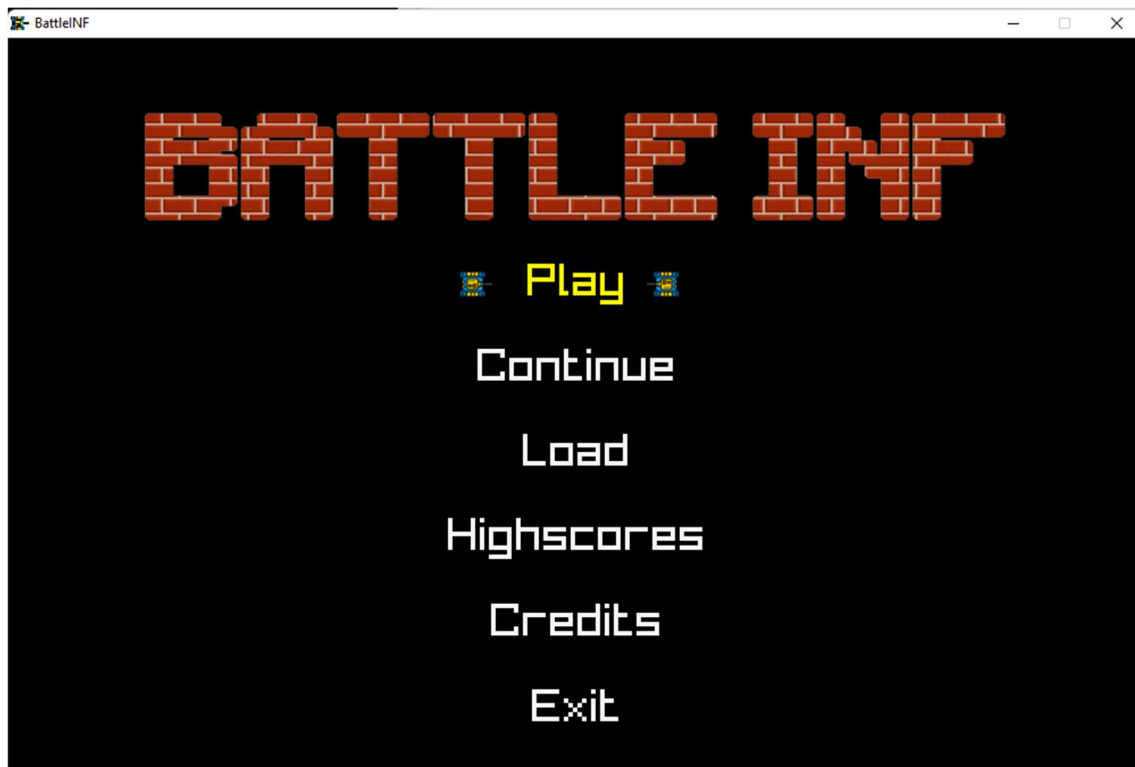


Trabalho Final - BattleINF
Algoritmos e Programação - UFRGS 2021/2
Marcos Luiz Kurth Reckers - 00315653

Como jogar:

Ao abrir o Executável a primeira tela é o menu principal:



Play – Abre o jogo

Continue – Continua de um jogo salvo (Caso não tenha um jogo salvo ele abre na primeira fase com 0 pontos)

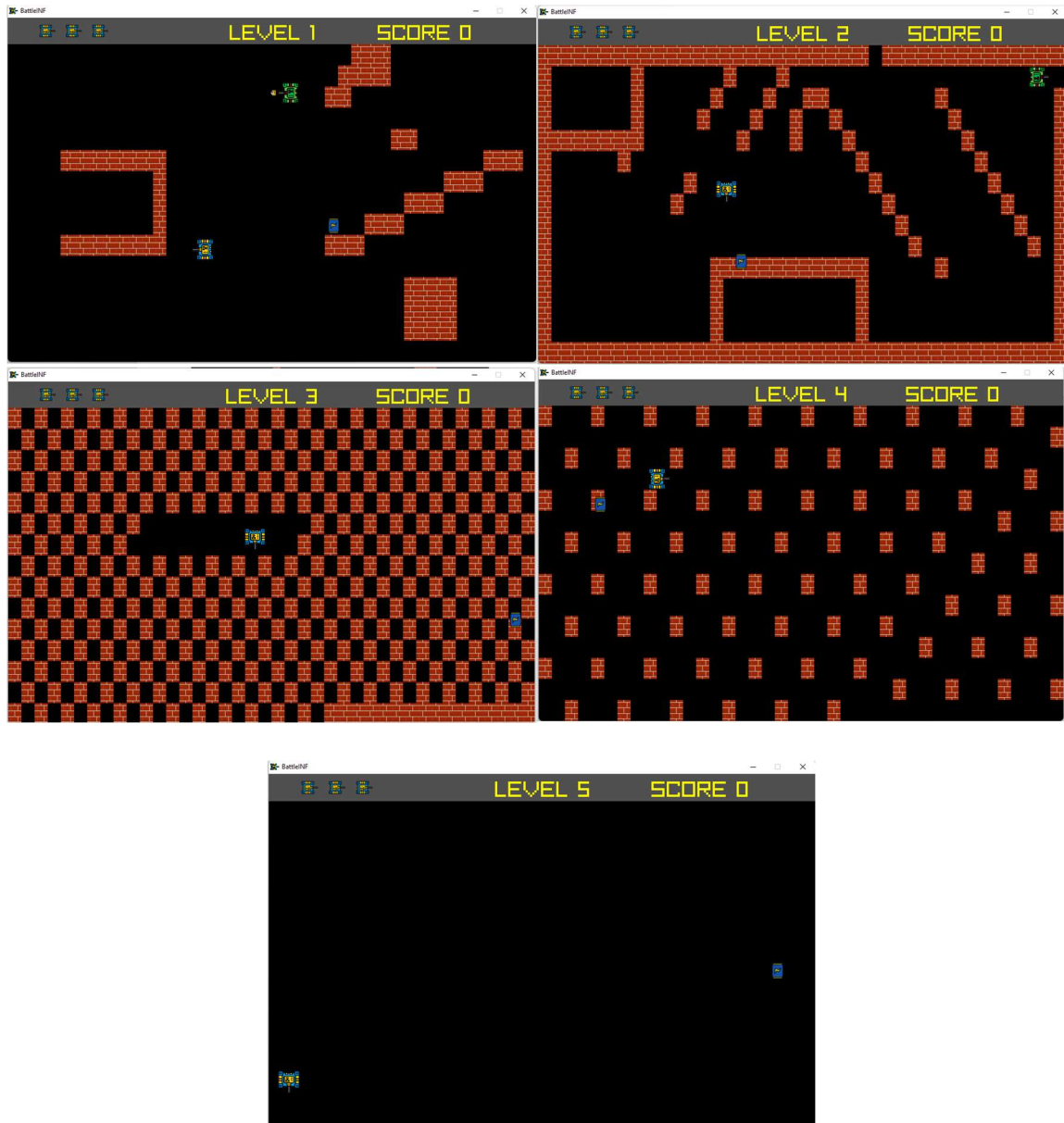
Load – Permite ao usuário carregar uma fase específica (1, 2, 3, 4 ou 5. Ideal para Treinos).

Highscores – Abre uma tela com as 5 melhores pontuações.

Credits – Abre uma tela de créditos ao desenvolvedor.

Exit – Fecha o jogo

FASES:



O design das fases foi feito de forma meio corrido, mas cada uma apresentando uma dificuldade diferente

Com muitos ou nenhum obstáculo.

A mudança de fase é feita quando matar todos os 10 inimigos do nível.

O jogo acaba quando o jogador não tiver mais vidas.

Jogabilidade:

O Tank Aliado é controlado pelas setinhas do teclado e no espaço para atirar. ele e os inimigos tem a mesma velocidade de disparo, porem o Tank Aliado tem maior velocidade de movimento.

Inimigos:

Os inimigos começam a aparecer após os 3 segundos de jogo.

a cada 4 segundos é feita uma verificação se não existem mais de 5 inimigos ao mesmo tempo na tela, caso retorne falso, um novo inimigo aparece.

eles se movimentam de forma aleatória até colidirem ou se alinharem, tanto verticalmente quanto horizontalmente, com o player.

Célula de energia:

Uma célula de energia aparece aleatoriamente pelo mapa (mesmo dentro de paredes), ao colidir com ela o Player recebe um aumento de velocidade de movimento e na velocidade das balas.

PAUSE:

O Player pode dar pause a qualquer momento clicando na tecla “ESC”, abrindo o seguinte menu:



Resume – Volta ao jogo normalmente.

Save – Cria um save-game.

Main Menu – Volta para o menu inicial.

Exit – Encerra o jogo

Estruturas:

O jogo possui somente 2 estruturas subdivididas internamente:

```
typedef struct cfg
{
    // Game settings
    //-----
    int game_screen_width; // Largura da tela menos o cabeçalho
    int game_screen_height; // Altura da tela menos o cabeçalho
    bool pause; // Verificação de pausa
    bool exit_window; // Verificação de saída da janela
    int frames_counter; // Contador de frames
    int select; // Seleção de opção
    int pause_select; // Seleção de opção na tela de pausa
    Sound fx_select; // Som de seleção
    Texture2D right_tank; // Textura do tanque direita
    Texture2D left_tank; // Textura do tanque esquerda
    int level; // Nível atual
    //-----

    // Player
    //-----
    Texture2D player_texture; // Textura do tanque
    Vector2 player_texture_origin; // Origem da textura do
tanque
    Rectangle player_pos; // Posição do tanque
    float player_rotation; // Rotação do tanque
    bool player_collision_check; // Verificação de colisão do
tanque
    Vector2 player_collision_pos; // Posição de colisão do
tanque
    int player_score; // Pontuação do tanque
    int player_lives; // Vidas do tanque
    float player_speed; // Velocidade do tanque
    int player_bullet_time_shot; // Tempo de espera para atirar
    float player_bullet_speed; // Velocidade da bala do tanque
    Rectangle player_bullet_pos; // Posição da bala do tanque
    float player_bullet_rotation; // Rotação da bala do tanque
    double player_bullet_time_active; // Tempo em que a bala
esta no ar
    //-----

    // Enemy
    //-----
    Texture2D enemy_texture; // Textura do inimigo
```

```

    Vector2 enemy_texture_origin; // Origem da textura do
inimigo
    int enemy_amount; // Quantidade de inimigos
    int enemy_lives[10]; // Vidas dos inimigos
    float enemy_speed; // Velocidade dos inimigos
    float enemy_bullet_speed[10]; // Velocidade da bala dos
inimigos
    Rectangle enemy_pos[10]; // Posição dos inimigos
    float enemy_rotation[10]; // Rotação dos inimigos
    Rectangle enemy_bullet_pos[10]; // Posição da bala dos
inimigos
    float enemy_bullet_rotation[10]; // Rotação da bala dos
inimigos
    Vector2 enemy_collision_pos[10]; // Posição de colisão dos
inimigos
    bool enemy_collision_check; // Verificação de colisão dos
inimigos
    int enemy_time_spawn; // Tempo de spawn dos inimigos
    int enemy_counter; // Contador de inimigos
    int enemy_on_screen; // Quantidade de inimigos na tela
    int enemy_random_rotate[10]; // Rotação aleatória dos
inimigos
    int enemy_bullet_time_shot[10]; // Tempo de espera para
atirar
    double enemy_bullet_time_active[10]; // Tempo em que a bala
esta no ar
    int enemy_kills; // Quantidade de inimigos mortos

    //-----

    // Energy cell
    //-----
    Rectangle energy_cell_rec; // Posição da célula de energia
    Texture2D energy_cell_texture; // Textura da célula de
energia
    bool energy_cell_active; // Verificação de ativação da
célula de energia
    int energy_cell_time_active; // Tempo em que a célula de
energia esta ativa
    bool energy_cell_spawn; // Verificação de spawn da célula de
energia
    int energy_cell_time_spawn; // Tempo de spawn da célula de
energia

    // Bullets

```

```

//-----
Texture2D bullet_texture; // Textura da bala
Vector2 bullet_texture_origin; // Origem da textura da bala
//-----

// Map
//-----
Texture2D map_texture; // Textura do mapa
Rectangle map_pos[15][41]; // Posição dos blocos do mapa
//-----

// Save
//-----
bool save_game; // Verificação de salvar o jogo
//-----

//Load
//-----
char load_file[100]; // Nome do arquivo a ser carregado
bool load_game; // Verificação de carregar o jogo
//-----
} cfg;
typedef struct HIGHSCORE
{
    char name[10]; // Nome do jogador
    int score; // Pontuação do jogador
} HIGHSCORE;

```