

TRABALHO DE ALGORITMOS E PROGRAMAÇÃO – ANA LAURA E LÉO HARDT

Descrição do jogo:

O jogo começa pelo menu, onde o usuário pode escolher uma dentre as opções: “Novo Jogo”, “Continuar”, “Carregar mapa”, “Ranking”, “Ajuda”, “Sobre” e “Sair”.

As três primeiras opções levam o usuário ao jogo, entretanto, de formas diferentes. “Novo jogo”, como o nome diz, começa um novo, mas antes de começar a jogar o jogador pode informar seu nome. “Continuar” permite que o jogador continue jogando um jogo salvo anteriormente. “Carregar mapa” permite que o jogador digite o nome do arquivo .txt com o mapa que quer jogar.

O jogo em si é baseado no jogo “MarioBros.”, de 1983, da Nintendo e o jogador pode controlar o Mario utilizando as setas do teclado, e também a tecla ‘D’ para pular. É possível salvar o jogo e sair apertando a tecla ‘A’, o que permite que o jogador continue de onde parou quando voltar a jogar. O jogo chega ao fim quando o Mario matou todos seus inimigos ou quando o Mario morreu. Ao fim de um jogo, que pode terminar em duas fases ou numa única fase se o jogador tiver carregado um mapa, é possível ver um ranking com as melhores pontuações obtidas pelos usuários ao jogar.

Esse ranking pode ser acessado direto do menu ao selecionar a opção “Ranking”. No menu também é possível acessar uma página de ajuda, que mostra as instruções de jogo, e outra de informações sobre o jogo.

ESTRUTURAS

```
typedef struct Highscore {  
    char nome[ 1 + HIGHSCORE_NAME_SZ  
];  
    int score;  
} Highscore;
```

```
typedef struct Vector2f {  
    double x,y;  
} Vector2f;
```

```
typedef struct Moeda {  
    Vector2f pos;  
    Vector2f vel;  
    bool coletada;  
} Moeda;
```

```
typedef enum Tipolnimigo {  
    T_TARTARUGA,
```

T_CARANGUEJO

```
} Tipolnimigo;
```

```
typedef struct Inimigo {  
    Tipolnimigo tipo;  
    bool vivo;  
    Vector2f pos;  
    Vector2f vel;  
    Vulnerabilidade vulnerabilidade;  
} Inimigo;
```

```
typedef struct Mario {  
    Vector2f pos;  
    Vector2f vel;  
    int score;  
    int vidas;  
} Mario;
```

```
typedef struct Fase {
```

```

    char mapa[FASE_ALTURA +
1][FASE_LARGURA + 1];

    Mario mario;

    int n_moedas;

    Moeda moedas[NMAX_MOEDAS];

    int n_inimigos;

    Inimigo inimigos[NMAX_INIMIGOS];

    int n_mapa;

    int n_caranguejos, n_tartarugas, delay;
} Fase;

typedef struct TelaJogoInfo {
    Fase fase;
}TelaJogoInfo;

typedef struct Jogo {
    char        janela_titulo[256];

    int          janela_largura;

    int          janela_altura;

    Font         fonte_menu;

    Sound        sons[N_SONS];

    Texture      spritesheet;

    Tela         tipo_tela;

    // Número de 1/60 segundos passados;
    int ticks;

    //Pontuação
    int pontos;

    //Nome do jogador
    char nome_jogador[50];

    //Nome do arquivo da fase
    char nome_fase[50];

    //Número da fase
    int num_fase;

    //Núm de vezes que o botão power foi usado
    int num_power;

    // Se o jogo deve fechar no próximo loop.

```

```

    bool sair;

    // Qualquer informação que a tela atual
    queira guardar ficaria aqui.

    union {

        TelaMenuInfo *
        tela_menu;

        TelaJogoInfo *
        tela_jogo;

        TelaHighscoreInfo *
        tela_highscore;

        TelaCarregaInfo * tela_carrega;

        TelaNomeInfo*      tela_nome;

        TelaInformacoesInfo *
        tela_informacoes;

    };

} Jogo;

/* Todas as informações que são pertinentes somente
a uma tela*/

struct TelaCarregaInfo {
    char arquivo[TAM_MAX];
    int nChar;
};

struct TelaHighscoreInfo {
    Highscore * scores;
    int n_highscores;
};

struct TelaInformacoesInfo {
    char texto[3][TAM_MAX];
};

struct TelaMenuInfo {
    Color cores[2];
    int options[10];
    TelaMenuOption selecionada;
};

struct TelaNomeInfo {
    char nome[TAM_MAX];
    int nChar; };

```