

# Trabalho Final Algoritmos e Programação 2022/1

**Afonso Hafner de Britto, Diego Hommerding Amorim**

Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)

[afonsobritto@hotmail.com](mailto:afonsobritto@hotmail.com), [diegoufrgs2004@gmail.com](mailto:diegoufrgs2004@gmail.com)

**Abstract:** This article describes the process of creation and operation of the game made as the final project of the Algorithms and Programming discipline of the 2022/1 semester. The article is divided in (i) the defined lore of the game; (ii) developed architecture of the application: modules, structures and utilized functions; (iii) organization of the project in the duo; (iv) challenges encountered and knowledge acquired (v) references.

**Resumo:** Este artigo descreve o processo de criação e o funcionamento do jogo feito como trabalho final da disciplina de Algoritmos e Programação do semestre 2022/1. O artigo é dividido em (i) enredo definido para o jogo; (ii) arquitetura da aplicação desenvolvida: módulos, estrutura e funções utilizadas; (iii) organização do trabalho na dupla; (iv) desafios enfrentados e aprendizados adquiridos; (v) referências.

**Sumário:**

- 1) Capa
- 2) Sumário
- 3) I) Enredo definido para o jogo
- 4) II) Arquitetura da aplicação desenvolvida
- 7) III) Organização do trabalho na dupla
- 8) IV) Desafios encontrados e conhecimentos adquiridos
- 9) V) Referências

## **I) Enredo definido para o jogo**

O personagem principal do jogo é um cavaleiro chamado Wilhem.

Wilhem era um nobre guerreiro, sentinela do castelo da dinastia Wong, detentores do trono de Wongloo, um império conhecido por sua ferocidade em batalhas, acumulando inúmeras vitórias e conquistas.

O nosso cavaleiro, durante a sua possível última batalha, foi traído por um de seus companheiros, resultando na derrota de seu império. Wilhem foi considerado culpado pela derrota de seu povo e foi condenado por seu rei a ir para a temida “Masmorra Escura”, de onde Wilhem terá de escapar para provar seu valor novamente a seu povo.

Dentro dessa masmorra, o cavaleiro encontrará diversos obstáculos: baús com armadilhas, enigmas para resolver, mas também poderá encontrar muitos tesouros que poderão aumentar sua glória perante seu povo após fugir da masmorra.

Muitos desacreditam do guerreiro, pois, até hoje, todos que foram sentenciados à masmorra não foram mais vistos. A missão de Wilhem é difícil, porém, ele terá de provar seu valor para encontrar a luz novamente e poder trazer à tona e expor o verdadeiro traidor de seu povo, retomando seu lugar como sentinela e recebendo o perdão de seu imperador.

## II) Arquitetura da aplicação desenvolvida

Módulos criados:

- 1) arquivo.h
- 2) bau.h
- 3) coordenadas.h
- 4) inputs\_jogador.h
- 5) jogador.h
- 6) mapa.h
- 7) menu.h
- 8) movimentos.h

Módulo.1: Módulo criado para importar funções que lidam com os arquivos necessários para o funcionamento do jogo.

Módulo.2: Módulo criado para suportar a estrutura baú.

Módulo.3: Módulo criado para suportar a localização de um jogador dentro do mapa.

Módulo.4: Módulo criado para importar a função input\_jogador.

Módulo.5: módulo criado para suportar as estruturas Mapa e Texturas, além de importar todas as funções relacionadas ao mapa do jogo.

Módulo.6: módulo criado para importar funções utilizadas no menu do jogo.

Módulo.7: módulo criado para importar funções utilizadas na movimentação do jogador.

Funções utilizadas:

void arq\_salva\_jogo(Mapa mapa);

Função criada para salvar o progresso do jogo.

void arq\_recupera\_jogo(Mapa \*mapa);

Função criada para recuperar o progresso salvo do jogo.

void arq\_carrega\_mapa(Mapa \*mapa, char \*nomeArq);

Função criada para carregar um mapa salvo em arquivo texto.

Localizacao localizacao\_cria(int linha, int coluna);

Função criada para criar uma localização dentro do mapa, sendo linha a posição y e coluna a posição x no plano cartesiano.]

void input\_jogador(Mapa\* mapa);

Função criada para receber o *input* do jogador e definir a ação dentro do jogo.

Jogador jog\_inicializa(int x, int y);  
Função criada para inicializar um jogador dentro do jogo.

void inicializa\_baus(Mapa \*mapa);  
Função criada para gerar os baús dentro do mapa.

char gera\_gema(Mapa \*mapa, int gema);  
Função criada para gerar uma gema dentro de um baú.

void abre\_bau(Mapa \*mapa);  
Função criada para verificar se um baú foi aberto.

void mapa\_carrega(Mapa \*mapa, int nivel);  
Função criada para carregar uma fase do jogo.

void mapa\_muda\_nivel(Mapa \*mapa, int \*nivel);  
Função criada para mudar o nível da fase.

void mapa\_reseta\_player(Mapa \*mapa);  
Função criada para colocar o jogador na posição inicial da fase.

void mapa\_diminui\_vida(Mapa \*mapa);  
Função criada para diminuir a vida do jogador.

void mapa\_movimenta\_vertical(Mapa \*mapa, int direcao);  
Função criada para lidar com a movimentação vertical do jogador.

void mapa\_localiza\_jogador(Mapa \*mapa);  
Função criada para definir a localização do jogador.

void mapa\_busca\_porta(Mapa mapa, char porta, Localizacao \*local\_porta);  
Função criada para buscar onde estão as portas do mapa.

void mapa\_imprime(Mapa mapa, Texturas texturas);  
Função criada para imprimir o mapa na tela do jogador.

void mapa\_imprime\_info(Mapa mapa);  
Função criada para imprimir as informações da fase atual.

void mapa\_muda\_movimento(Mapa \*mapa, bool movendo);  
Função criada para animar o movimento do jogador no mapa.

void mapa\_muda\_movimento\_escada(Mapa \*mapa);  
Função criada para verificar a direção para a qual o jogador está se movendo.

void mapa\_inicializa\_texturas(Texturas \*texturas);  
Função criada para inicializar as texturas do jogo em uma estrutura.

void DesenhaMenu(int opcao);

Função criada para desenha o menu do jogo na tela.

void menuJogo(Mapa \*mapa, int \*nivel, int \*flag, Ranking ranking[], int \*posicoes);

Função criada para receber a opção que o jogador escolher do menu e definir a ação dentro do jogo.

void gameOver(Mapa \*mapa, int \*menu);

Função criada para desenhar *game over* na tela do jogador caso ele morra.

void gravidade(Mapa\* mapa);

Função criada para implementar a gravidade dentro do jogo.

void movimenta\_vertical(Mapa\* mapa, int direcao);

Função criada para lidar com o movimento vertical do jogador.

void movimenta\_horizontal(Mapa\* mapa, int direcao);

Função criada para lidar com o movimento horizontal do jogador.

Ranking cria\_ranking(char nome[NOMEMAX], int pontuacao);

Função criada para criar um elemento da estrutura ranking.

void salva\_rankings(Ranking entradas[VETORMAX], int ocupacao);

Função criada para salvar um vetor de rankings em um arquivo texto.

void carrega\_rankings(Ranking entradas[VETORMAX], int \*ocupacao);

Função criada para carregar o ranking salvos em um arquivo texto e contar a quantidade de jogadores no ranking.

void imprime\_rankings(Ranking entradas[VETORMAX], int ocupacao);

Função criada para imprimir o ranking do jogo na tela.

void adiciona\_ranking(Ranking entradas[VETORMAX], int \*ocupacao, Ranking novo\_jogador);

Função criada para adicionar um novo jogador ao ranking.

void recebe\_nome(Mapa \*mapa);

Função criada para receber o nome do jogador após ele completar o jogo.

### III) Organização do trabalho na dupla

O trabalho foi dividido em duas frentes: o *backend* e o *frontend*. Sendo o primeiro, o funcionamento por detrás do sistema do jogo, e o segundo a interface de interação com o usuário e a parte gráfica.

O *backend* foi responsabilidade do autor Diego Hommerding Amorim, enquanto o *frontend* foi desenvolvido pelo autor Afonso Hafner de Britto.

A divisão de tarefas não foi majoritária, havendo colaboração entre os autores para o desenvolvimento de cada frente.

Foram utilizadas diversas ferramentas para o auxílio na interação entre os autores. Para o compartilhamento de código foi utilizado um repositório criado pela dupla no *GitHub* e para a comunicação, a plataforma *Discord*.

O trabalho foi desenvolvido, integralmente, de forma remota por ambos os criadores, não havendo encontros de maneira presencial durante o desenvolvimento do jogo.

#### **IV) Desafios enfrentados e conhecimentos adquiridos**

Foram muitos os desafios encontrados para desenvolver a aplicação. Principalmente pelo fato de ter de aprender a lidar com uma nova biblioteca da linguagem C. A biblioteca *raylib* sendo algo novo para nós - nos forçando a aprender novos comandos e funções, os quais nunca tínhamos estudado - foi na própria programação na linguagem C que encontramos os maiores problemas. Tendo de lidar com situações inusitadas para nós dentro da programação e arrumando *bugs* que não conhecíamos, em uma linguagem tão rigorosa quanto essa se tornou um grande desafio, aliando esses problemas ao fato de termos de combinar a programação em C com a biblioteca *raylib* dificultou ainda mais o trabalho.

Ao final do projeto foram muitos os conhecimentos adquiridos. Aprendemos a usar novas funções e como lidar com diversos novos elementos - como ponteiros, estruturas, arquivos -, desenvolvendo ainda mais nossa lógica de pensamento após resolver tantos problemas e corrigindo diversos *bugs* encontrados durante o caminho. Além dos conhecimentos técnicos de computação adquiridos, um dos maiores benefícios durante esse percurso da criação dessa aplicação foi com certeza a experiência de um trabalho com um time. Os conhecimentos adquiridos durante o desenvolvimento da aplicação com certeza poderão ser aplicados no mundo profissional e/ou acadêmico, tendo sido uma ótima experiência para adquirir novos conhecimentos de forma leve e divertida.



## V) Referências

Assets utilizados:

2D SL Character by Szadi art.

Disponível em: <<https://szadiart.itch.io/2d-soulslike-character>>

2D Environment Platformer Asset Pack: Castle of Despair by brullov.

Disponível em: <<https://brullov.itch.io/2d-platformer-asset-pack-castle-of-despair>>

Pirate Bomb by Pixel Frog.

Disponível em: <<https://pixelfrog-assets.itch.io/pirate-bomb>>

Rogue Fantasy Catacombs by Szadi art.

Disponível em: <<https://szadiart.itch.io/rogue-fantasy-catacombs>>

Raylib - Inputbox

Disponível em: <<https://www.raylib.com/examples.html>>