

Projeto de Desenvolvimento de Software

Dungeons of Latserolf

Alan Araújo dos Reis - 5096

Álvaro Gomes da Silva Neto - 5095

Arthur Ataíde de Melo Saraiva - 5070

Gabriel Rodrigues Marques - 5097

Marcos Biscotto de Oliveira - 4236

INTRODUÇÃO.....	3
DESENVOLVIMENTO.....	3
• database:.....	4
• display:.....	4
• entity:.....	4
• graphics:.....	5
• map:.....	5
• system:.....	5
RESULTADOS.....	6
CONCLUSÃO.....	8
REFERÊNCIAS.....	9

INTRODUÇÃO

Como trabalho final da disciplina de Programação Orientada a Objetos, foi solicitado aos grupos que desenvolvessem uma aplicação que dispusesse dos conceitos abordados em sala de aula e que fosse criativo. Com isso em mente, uma noção de tempo não muito condizente com a realidade e muito amor por jogos, nosso grupo teve a ideia de desenvolver um jogo do zero, cheio de desafios e oportunidades.

Eis então que surge Dungeons of Latserolf, um jogo de temática roguelike, com labirintos e monstros perigosos que prometem desafiar até mesmo os mais competitivos jogadores. Será que você é capaz de encontrar as chaves necessárias que destrancam esse pesadelo interminável de salas e tesouros?

Infelizmente, nosso primeiro escopo teve de sofrer algumas alterações para que pudesse ser finalizado dentro do prazo estipulado. Abordaremos isso mais para frente.

“Conhecimento é poder? PODER É PODER!” - Latserolf.

DESENVOLVIMENTO

Inicialmente, após muitas reuniões e discussões, foi decidido o escopo principal do projeto: um jogo de exploração de labirinto com temática/aspectos de jogos como Dungeons & Dragons e Munchkin. O projeto e o repositório foi organizado da seguinte forma:

doc/ - Documentação.

lib/ - Dependências.

res/ - Recursos.

src/ - Código-fonte.

Adentrando a pasta que contém o código fonte do projeto é possível identificar os seguintes pacotes:

database/ - Sistema de banco de dados.

display/ - Interface gráfica.

entity/ - Entidades do jogo.

graphics/ - Recursos gráficos.

map/ - Sistema de mapa.

system/ - Sistema de controle.

De maneira geral, a implementação do projeto foi feita em 4 grandes partes, o sistema de mapa gerado proceduralmente, as entidades do jogo, os sistemas que controlam o fluxo de execução e o display responsável por toda parte de interfaces de usuário. Como parte auxiliar, foi implementado uma parte de recursos gráficos, responsável por armazenar e manipular as imagens utilizadas no projeto.

É possível observar também que há uma parte referente ao sistema de banco de dados no repositório, que faria toda conexão com o banco de dados, armazenando os usuários e suas pontuações máximas de jogo. Porém, apesar de ter sido devidamente implementado, acabou não sendo possível integrá-lo ao sistema principal, por enquanto, devido a algumas dificuldades enfrentadas e ao curto período de tempo.

- **database:**

- **DatabaseManagerSystem:** gerencia a interação da aplicação com um banco de dados MySQL. A classe oferece métodos para realizar operações em um banco de dados, como inserir um novo usuário, fazer/verificar login, exibir todos os usuários ordenados por pontuação, atualizar a pontuação do usuário e excluir todos os usuários, também inclui métodos auxiliares para obter e definir o usuário atual.
- **UserEntity:** representa a entidade de usuário atual na aplicação.

- **display:**

- **BattleDialog:** representa uma caixa de diálogo de batalha, exibindo a descrição do monstro encontrado, permitindo que o jogador decida se deseja batalhar ou não.
- **BattleFrame:** representa a janela de batalha, proporcionando uma interface gráfica interativa dos eventos da batalha, exibindo as rolagens de dados, a vida do jogador e do monstro e o resultado do combate.
- **DungeonFrame:** representa o painel onde o mapa do jogo e o jogador são exibidos, controlando a visibilidade do mapa com base na posição do jogador e no que foi explorado.
- **GameInterface:** representa a interface gráfica principal do jogo, exibindo o mapa e o jogador, além de fornecer informações de pontuação e quantidade de chaves coletadas.

- **entity:**

- **MonsterCategory:** representa uma categoria de monstros. A classe permite a definição e personalização de diferentes tipos de monstros no jogo.
- **MonsterEntity:** representa um monstro, fornecendo informações básicas sobre o mesmo.
- **MonsterGenerator:** responsável por gerar instâncias aleatórias e diversificadas de monstros com base em categorias predefinidas.
- **PlayerComponent:** representa os componentes individuais do jogador, fornecendo informações sobre sua pontuação, total de chaves coletadas, posição, vida, ataque e defesa.
- **PlayerController:** responsável por gerenciar as entradas do teclado relacionadas ao controle do jogador.
- **PlayerEntity:** representa a entidade do jogador no jogo, fornecendo informações básicas sobre o mesmo, além de métodos para realizar diversas ações.

- **graphics:**
 - **AssetLibrary:** responsável pelo gerenciamento e carregamento de ativos gráficos, como imagens de personagens, monstros e outros elementos visuais do jogo. Ela utiliza uma estrutura de HashMap para associar nomes de arquivos a imagens correspondentes, permitindo o acesso eficiente aos ativos durante a execução do jogo.
- **map:**
 - **TileTypeEntity:** representa, através de uma classe abstrata, um tipo genérico de tile no mapa do jogo, o bloco básico fundamental para construção de todo o jogo. A classe é essencial para a representação diversificada dos diferentes tipos de terrenos e objetos no mapa do jogo.
 - **Chest:** representa um baú.
 - **Door:** representa uma porta.
 - **Exit:** representa a porta de saída.
 - **Floor:** representa o chão.
 - **Wall:** representa uma parede.
 - **MapData:** representa os dados associados a um mapa específico. A classe inclui informações como o tamanho do mapa, o número de chaves necessárias e as probabilidades de portas, monstros e baús. Esses dados são parâmetros fundamentais utilizados para configurar e gerar mapas no jogo,
 - **MapEntity:** representa a entidade mapa e encapsula a lógica e os dados associados aos dados e geração do mapa de jogo.
 - **MapGeneratorSystem:** a classe mais importante de todo o Dungeons of Latserolf. O sistema responsável por gerar todo o mapa do jogo. Ele utiliza um algoritmo baseado em Backtracking e define a posição dos elementos do mapa, como paredes, portas, saída, baús e monstros, com base nas probabilidades predefinidas. O tamanho do mapa é determinado por um número primo aleatório, entre 7 e 107, e a distribuição de elementos é controlada por diversas funções.
- **system:**
 - **BattleSystem:** sistema de batalha do jogo. É responsável por gerenciar as batalhas entre o jogador e os monstros. Utiliza rolagem de dados para determinar ataques e iniciativa e a lógica é baseada em turnos, alternando entre as ações do jogador e do monstro até que um deles seja derrotado.
 - **GameSystem:** sistema de controle de jogo, interagindo com a interface gráfica do jogo e controla eventos relacionados ao jogador, como movimento e interações com objetos no mapa.

“Tudo o que um dia respirou, se curvará perante a mim.” - Latserolf

RESULTADOS

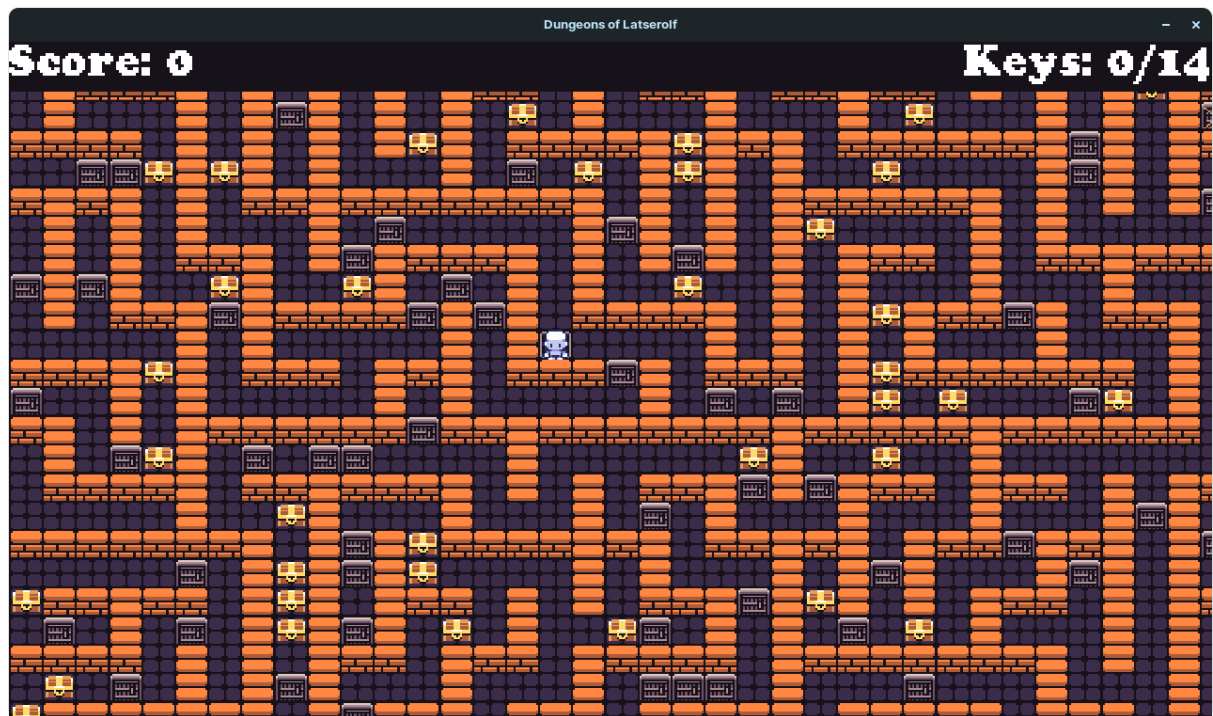


Imagem 1: Mapa grande gerado aleatoriamente.



Imagem 2: Mapa pequeno gerado aleatoriamente.

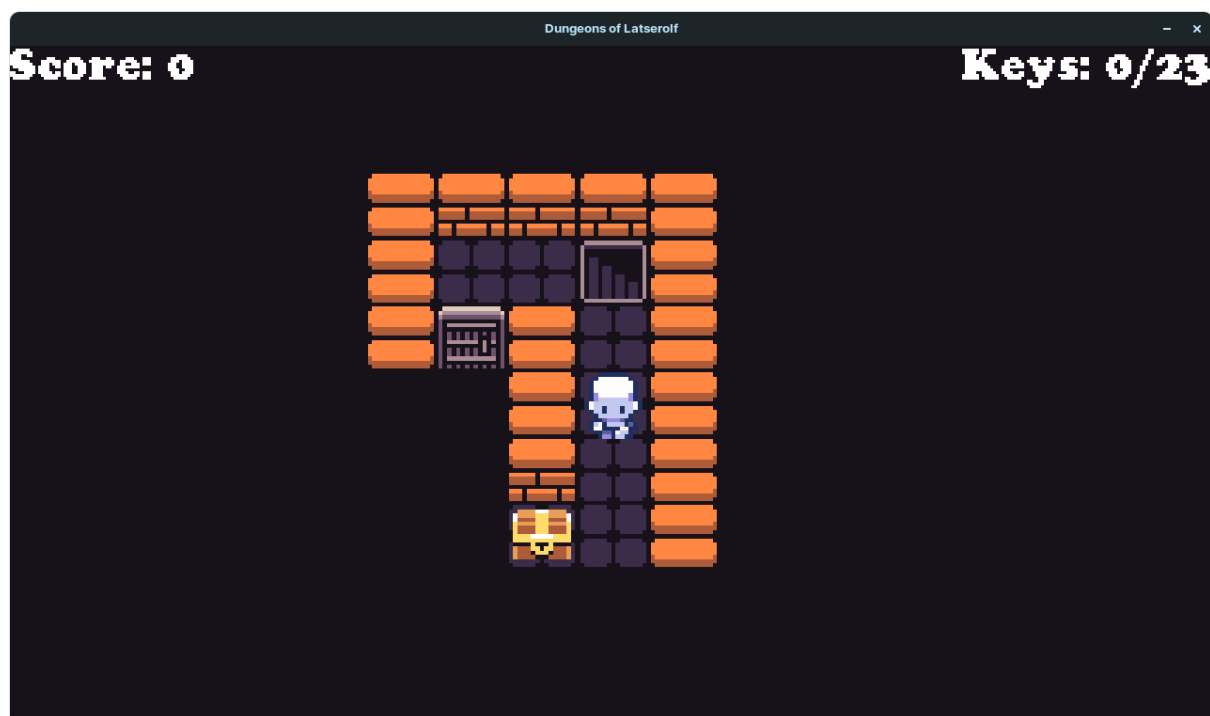


Imagem 3: In-game footage do personagem se aventurando na masmorra

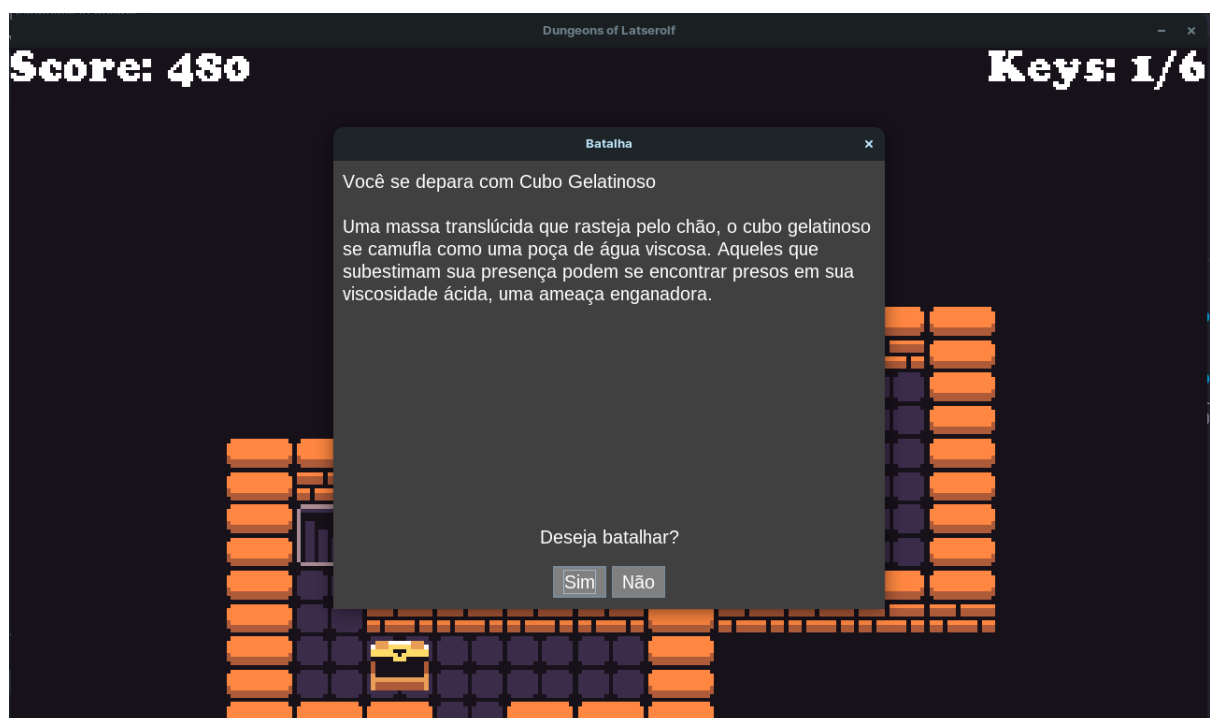


Imagem 4: Personagem abre a porta e encontra um inimigo. Deseja batalhar?

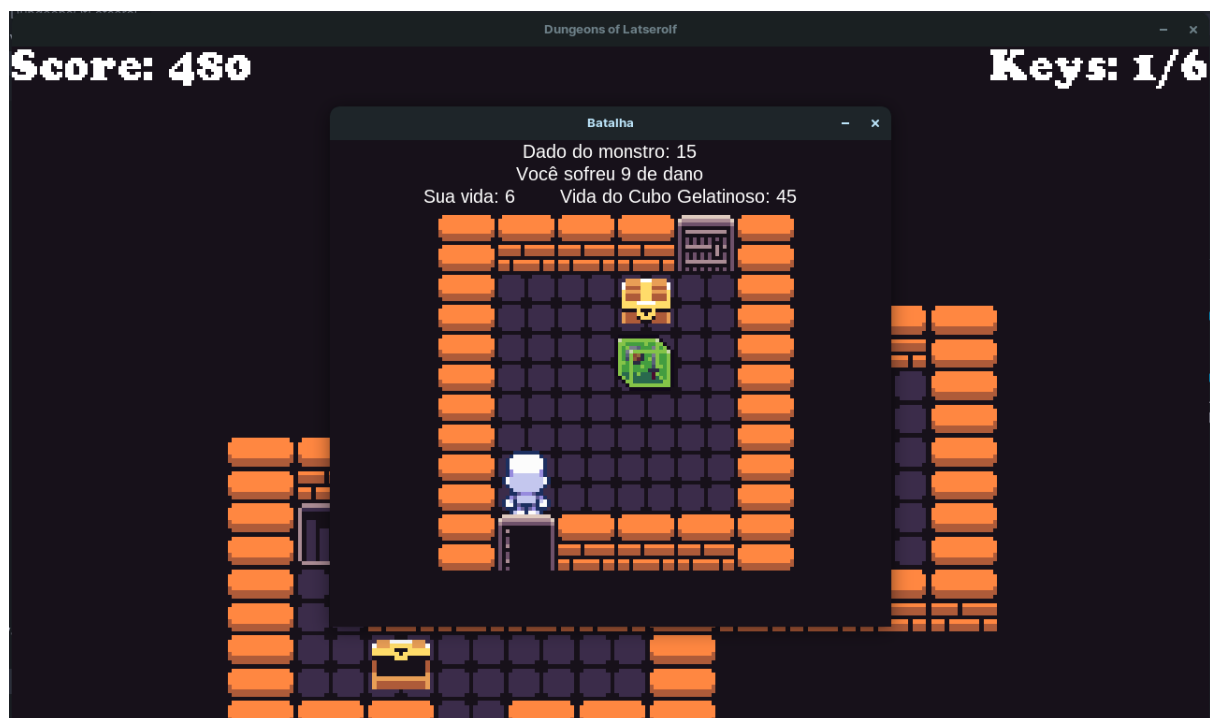


Imagem 5: Interface do combate, onde os dados dos combatentes são exibidos.

“Achou mesmo que uma mera magia de oitavo ciclo funcionaria em mim?” - Latserolf

CONCLUSÃO

Após horas dedicadas ao trabalho e com o término do mesmo, conseguimos entregar um projeto que, ao nosso ver, atende às expectativas geradas e aos critérios impostos. Acreditamos também que o trabalho foi de suma importância para o melhor entendimento e aprendizado em relação às competências da disciplina e aos principais conceitos da Orientação a Objetos e da linguagem Java. Com isso, apresentamos um jogo funcional, bem modularizado com código coeso, onde cada membro da equipe teve papel fundamental no seu desenvolvimento.

“Eu presenciei o Primeiro Cataclisma... Você jamais entenderia...” - Latserolf

REFERÊNCIAS

- [1] Introduction to Learning Java 2D Game Programming. Disponível em: <<https://youtube.com/playlist?list=PLfHzvwt11ruOl0h6nn72545ISqh6Y1aPb&feature=shared>>;
- [2] RogueBasin. Disponível em <https://www.roguebasin.com/index.php/Main_Page>;
- [3] 5etools. Disponível em <[5etools](#)>;
- [4] Github Copilot. Disponível em: <<https://github.com/features/copilot>>;
- [5] Github. Disponível em: <<https://github.com/gabridulol/DungeonsOfLatserolf>>;
- [6] itch.io. Disponível em: <<https://itch.io/>>;
- [7] DeepDiveGameStudio. Disponível em: <<https://deepdivegamestudio.itch.io/>>;
- [8] VEXED. Disponível em: <<https://v3x3d.itch.io/>>;

“Pergaminhos são fontes vastas de informações. Abuse deles.” - Latserolf