

PITCH CONCEPT

NOME DO JOGO: A Fuga de Teseu

VERSÃO: 1.0

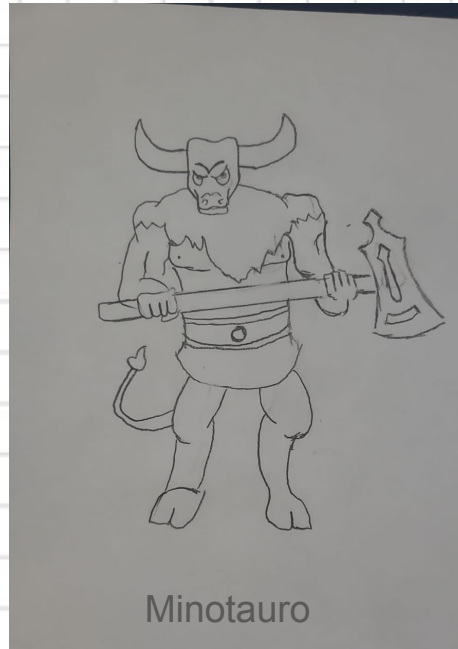
SINOPSE

O jogo se baseia em um labirinto completamente fechado de onde o jogador deve escapar enquanto foge, também, de um minotauro maligno. O mapa será repleto de itens mágicos que ajudam o jogador a escapar. Não seja pego pelo minotauro!

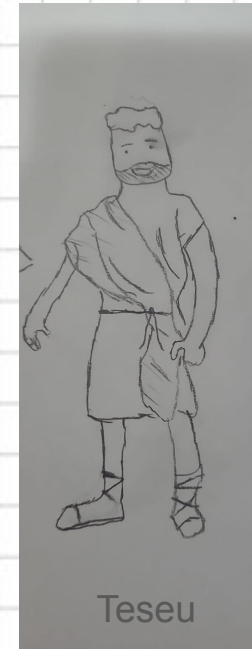
COMO JOGAR

O jogador, se movimentando com as teclas W A S D deve fugir do labirinto enquanto é perseguido por um monstro minotauro, podendo utilizar de itens gerados ao longo do caminho para facilitar sua saída.

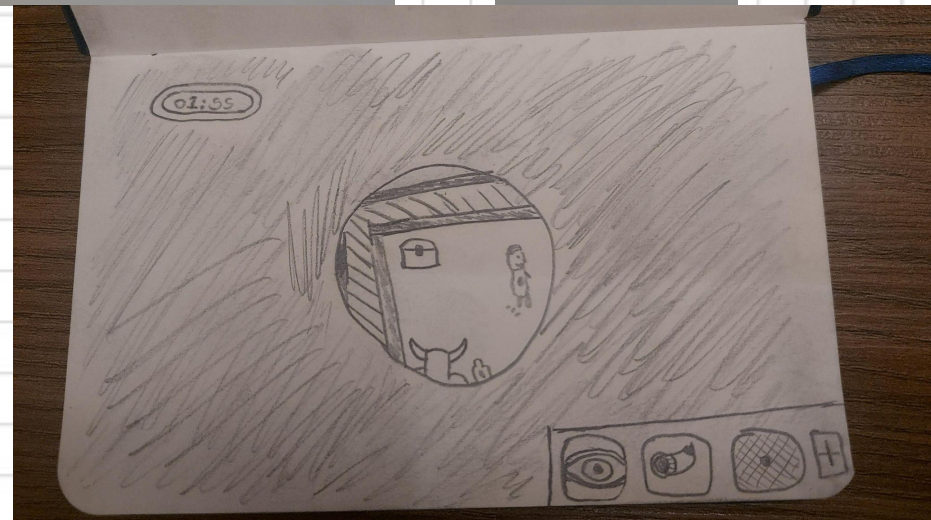
O jogador é derrotado caso o minotauro o atinja.



Minotauro



Teseu



GRUPO(integrantes): Fábio Antônio Caldeira, Gabriel Samarane, Guilherme Gomes de Brites, João Madeira, João Lucas Curi, Lucas Alkmim Barros, Pedro Marçal, Pedro Ribeiro

REFERÊNCIAS

Referências de jogos / gêneros / mecânicas/
estéticas / universos imagéticos / histórias



O mapa foi baseado em labirintos.



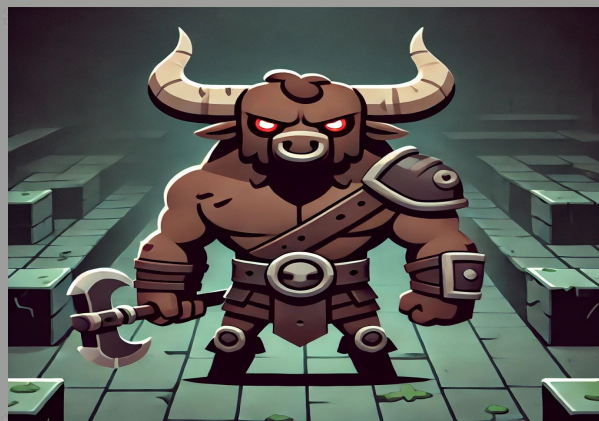
O inimigo será baseado no minotauro da mitologia grega.



PAC-MAN: inspiração pro sistema de itens que ajudam o jogador.



O jogador é perseguido por um monstro, como em Maze Runner.

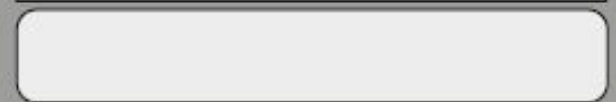
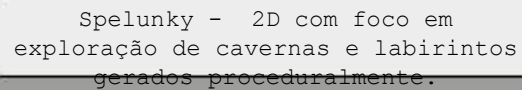
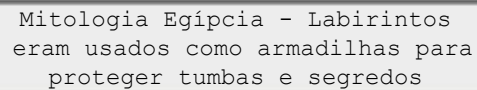
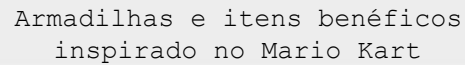
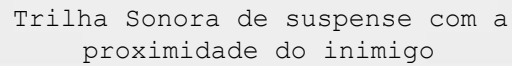


O mito do minotauro gira em torno de um labirinto.



Meize Rush: jogo do semestre passado com visão limitada.

Referências de jogos / gêneros / mecânicas/
estéticas / universos imagéticos / histórias



PITCH CONCEPT

Algoritmos de Grafos

Visão Geral - Mapeamento do labirinto para uso de itens guias (melhor caminho) e para o caminho tomado pelo inimigo.

Responde a pergunta: Como ele funciona e como ele implementado?

Vamos utilizar o Algoritmo A* com a heurística de Manhattan para calcular o melhor caminho possível entre dois pontos do grafo (labirinto), que no caso serão de Teseu para a saída (na utilização da Lã), ou do Minotauro para Teseu (Minotauro perseguindo Teseu), além de definir todo o labirinto como um grande grafo.

Algoritmos de IA

Visão Geral - Algoritmo de classificação para regular a frequência da geração de itens para o jogador ao longo da partida

Responde a pergunta: Como ele funciona e como ele implementado?

A IA após seu treinamento com dados fornecidos obtidos por pessoas jogando o jogo, irá por meio do Logistic Regression, classificar um jogador como "noob" ou "pro" no começo de cada fase, no qual um jogador pro terá sua porcentagem de geração de itens reduzida, enquanto o noob terá uma porcentagem maior, geração implementada por um script que manuseia a criação desses itens nas redondezas de Teseu.

Cronograma

Setembro:

- **Semana 1:** Criação do layout básico do labirinto, definindo o design dos níveis e a estética geral.
- **Semana 2:** Implementação inicial do sistema de movimentação do personagem, incluindo a lógica básica de controle.
- **Semana 3:** Desenvolvimento e integração do sistema de grafos para representar o labirinto e suas conexões.
- **Semana 4:** Revisão e ajustes no layout e no sistema de movimentação para garantir uma base sólida para as próximas etapas.

Outubro:

- **Semana 1:** Implementação dos itens no jogo, começando pelos itens básicos que o jogador pode coletar.
- **Semana 2:** Início da implementação do sistema de IA, focando na geração dinâmica de armadilhas e itens baseados no progresso do jogador.
- **Semana 3:** Aperfeiçoamento da IA, ajustando a dificuldade de acordo com as ações do jogador e a evolução dentro do labirinto.
- **Semana 4:** Integração do sistema de dificuldade, ajustando parâmetros como velocidade, quantidade de armadilhas, e efeitos dos itens.

Novembro:

- **Semana 1:** Testes iniciais de jogabilidade para identificar bugs, problemas de equilíbrio, e áreas para otimização.
- **Semana 2:** Balanceamento dos desafios no labirinto, ajustando a curva de dificuldade e a distribuição de itens e armadilhas.
- **Semana 3:** Implementação de ajustes finais com base no feedback dos testes.
- **Semana 4:** Testes finais e polimento do jogo, preparando-o para o lançamento ou para uma fase de beta testing.

Divisão

Design e arte:

- Lucas Alkmin
- Guilherme de Brites
- Pedro Ribeiro

Implementação IA:

- Fabio Antônio
- João Madeira
- Gabriel Samarane

Implementação Grafos:

- João Lucas
- Gabriel Samarane
- Pedro Ribeiro

Mecânicas:

- Lucas Alkmim
- Guilherme de Brites
- Pedro Marçal