

RELATÓRIO TRABALHO FINAL (O VIAJANTE)

João Vitor Melo Fontenele José Lucas Vasconcelos de Lucena

June 27, 2023

Abstract

Resumo

1 Introdução

Nosso objetivo foi criar um jogo de simulação de logística baseado nos conceitos estudados ao longo do curso. Neste projeto, buscamos aplicar os conhecimentos teóricos adquiridos de forma prática, desenvolver habilidades de programação e trabalho em equipe, além de proporcionar uma experiência divertida e desafiadora aos jogadores.

A logística desempenha um papel fundamental em diversos setores, como transporte, comércio e cadeia de suprimentos. Através deste jogo, queríamos oferecer aos jogadores a oportunidade de vivenciar as complexidades e desafios enfrentados na gestão de uma rede de transporte de recursos. Ao simular situações reais, os jogadores serão desafiados a tomar decisões estratégicas, avaliar rotas de transporte, trocar recursos, tudo isso visando maximizar o fluxo de recursos, continuar vivo e atingir seu objetivo principal, que é chegar em Nargumun e se tornar Rei da Cidade.

2 Visão Geral

O objetivo deste projeto foi desenvolver um jogo de simulação de logística em linguagem Java, baseado em conceitos de estrutura de dados. O jogo permite aos jogadores gerenciarem uma rede de transporte de recursos, tomando decisões estratégicas para maximizar o fluxo de recursos e minimizar os custos de transporte. Para isso, utilizamos uma representação de grafo para modelar as cidades e as conexões entre elas.

3 Objetivos

- Implementar estruturas de dados para representar a rede de transporte de recursos.
- Criar eventos e desafios no jogo para que os jogadores tomem decisões estratégicas.
- Proporcionar uma experiência de jogo completa, equilibrada, divertida e desafiadora.
- Apresentar um documento explicando a lógica do design, a descrição do jogo e o código-fonte.

4 Implementação

Utilizamos a linguagem de programação Java para implementar o jogo, aproveitando seus recursos de orientação a objetos e estruturas de dados disponíveis. Ao longo do desenvolvimento, trabalhamos em equipe, dividindo tarefas e garantindo uma colaboração eficiente.

Para representar a rede de transporte de recursos, utilizamos uma estrutura de dados baseada em grafo. Cada cidade foi representada como um nó do grafo, e as conexões entre as cidades foram representadas por estradas, onde cada estrada tinha um peso associado que representava o custo de transporte dos recursos através dela. [GG12]

No jogo, criamos eventos e desafios para os jogadores, como questionários de mercadores e missões a serem aceitas.

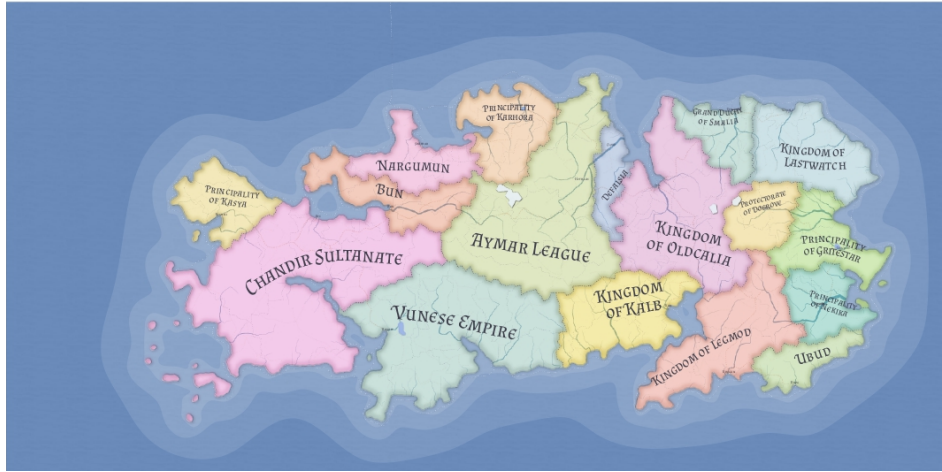


Figure 1: Mapa do Jogo

5 Resultados

Após realizar uma série de testes extensivos no jogo, temos o prazer de compartilhar os resultados positivos obtidos. Todos os testes foram bem-sucedidos e não foram encontrados erros significativos no código-fonte. A implementação das estruturas de dados, algoritmos de busca e fluxo de rede funcionou conforme o esperado.

A interface no console, apesar de simples, provou ser intuitiva e fácil de usar. Ela fornece aos jogadores informações claras sobre as cidades, conexões, custos de transporte e opções disponíveis. A interação com o jogador foi bem implementada, permitindo que eles aceitem ou recusem missões, escolham cidades de destino.

Durante os testes, observou-se que as decisões tomadas pelos jogadores têm um impacto significativo no fluxo de recursos e nos custos de transporte. As escolhas estratégicas são importantes para maximizar o fluxo e minimizar os custos, proporcionando aos jogadores um desafio interessante e motivador.

O tempo de execução desses algoritmos foi rápido o suficiente para garantir uma jogabilidade fluída, sem atrasos perceptíveis.

6 Descrição do Jogo

6.1 Classe: Caminho

Representa as arestas que vão formar as estradas entre cidades que possuem fronteira entre si. Cada caminho possui uma cidade de destino, que representa para onde Maxwell vai, e um custo que é representado por Moeda de Transporte, que é o que vai ser gasto por cada percorrimento no mapa. Métodos:

Getters e Setters e Construtor.

6.2 Classe: Cidade

Representa as características de cada cidade em que Maxwell percorre: Seu nome, a lista de caminhos que a cidade possui para outras cidades, O poder que a cidade vai afetar na joia que Maxwell leva e se a cidade possui missões.

Métodos: Possui o método para addEstrada() (Representa a criação de Caminhos que Maxwell pode percorrer para outras cidades).

Getters e Setters e Construtor.

6.3 Classe: Jogatina

Representa como vai ser a jogabilidade do usuário, quais as sequências de acontecimentos. Desde a saída de Ubud, todas as escolhas que o usuário pode fazer(Aceite de missões e escolha de estradas que quer percorrer) para chegar na cidade de Nargumun a salvo, e como Rei. Além de toda configuração da interface que será mostrada para o usuário.

Métodos: Dentro dessa classe que ocorre toda a criação do mapa, com as cidades e as estradas que elas possuem e todas as missões que existem em cada cidade, através da instanciação de todos esses objetos, além da chamada de métodos de outras classes para o bom funcionamento do jogo, que funciona em um While(True).

6.4 Classe: Display

Funcionará como uma modularização da classe Jogatina, para tornar o jogo mais reutilizável, dividido por funções que aparecerão na classe Jogatina

6.5 Classe: Missão

Representa a configuração de cada missão que Maxwell pode se deparar, na mesma podemos encontrar: Cidade de Origem(Cidade a qual a missão estará disponível no mapa), Missão(A explicação da missão para o usuario), Cidade Final(Cidade a qual o usuário deve levar maxuel para conseguir concluir a missão), Aceitar Missão(Forma com que o jogo pode identificar se maxuel está dentro de uma missão), Prêmio por Aceitar(Representa a premiação que maxwell receberá por ter aceitado fazer a missão), PremioTerminarF(Representa o prêmio em moedas de transporte que ele receberá ao concluir a missão), PremioTerminarS(Representa o premio em poder no limiar da joia que maxwell receberá).

Métodos: AceitarMISSAO()(Método que ativará uma missão para Maxwell, ou seja, ativará o atributo “Aceitar Missão” e entregará o “Prêmio por Aceitar” a Maxwell). VerificarMissão()(Método que informará que Maxwell conseguiu chegar a cidade final e poderá receber todos os prêmios da missão). ConcluirMissao()(Método que será ativo através do método VerificarMissão(), ele retornará a maxwell todos os prêmios). AbandonarMissao()(Sempre que maxwell mudar de cidade com uma missão ativa ele poderá abandonar a mesma, sem ter que voltar a cidade de ativação da missão).

Getters e Setters e Construtor.

6.6 Classe: Mercador

Representará o personagem “Mercador” no jogo, toda cidade que Maxwell passar, terá a escolha de conversar com o Mercador para conseguir trocar moedas de Transporte por limiar da Joia, mas ele pode se dar mal ou se dar bem. Tudo dependerá da quantidade de recursos que ele tenha e de suas escolhas.

Métodos: conversaMercador(): Como funcionará a dinâmica com o mercador, as perguntas que serão feitas a Maxwell e as recompensas que ele receberá, de acordo com suas respostas.

6.7 Classe: Maxwell

Representará o player que estará jogando, contendo suas características e recursos, através dessa classe que conseguimos puxar as informações necessárias para a dinâmica do jogo. Contém o número de Moedas de Transporte disponíveis para conseguir viajar, o limiar da joia, o poder atual da joia, a Cidade atual que Maxwell se encontra, e o booleano Vivo, para verificação de sua vida. Metodos: atualizarLimiar(): Faz a atualização do limiar total da joia, de acordo com as escolhas tomadas por Maxwell durante a Jogatina. verificarPoderAtual(): Faz a verificação do valor do Poder Atual da jóia e sempre que ele for menor que 0, se transforma em 0.

6.8 Classe: Verificações

Representará a classe que terá alguns métodos de verificação dentro do jogo, como : Se o player digitar uma informação válida, e em vez de quebrar o jogo, ignora aquela informação. Faz verificação de uma das regras do jogo, que a Joia que Maxwell carrega não pode ter o seu Poder Atual maior que o Limiar

Total dela, se não Maxwell morre, e fim do jogo. Além de fazer a verificação quando Maxwell não possuir mais moedas de transporte, e morrerá se acabar tentando viajar sem moeda nenhuma.

7 Equipe

João Victor Melo : Desenvolvimento do Jogo em Linguagem Java, Arquitetura do Software e Testes.

José Lucas Vasconcelos de Lucena: Desenvolvimento do Jogo em Linguagem Java, Produção de Relatório Técnico, Arquitetura do Software e Testes.

8 Conclusão

Em conclusão, o projeto de desenvolvimento do jogo de simulação de logística foi concluído com sucesso e proporcionou valiosos aprendizados ao longo do processo. Durante o desenvolvimento do jogo, pudemos aplicar os conceitos de estruturas de dados e algoritmos de busca e fluxo de rede, consolidando nosso conhecimento teórico e ganhando experiência prática na implementação desses conceitos em um cenário real.

Ao longo do projeto, enfrentamos alguns desafios significativos. Um deles foi o gerenciamento eficiente do tempo, garantindo que todas as partes do jogo fossem implementadas dentro do prazo estabelecido. A organização e o planejamento adequados foram fundamentais para superar esse desafio e entregar o projeto dentro das expectativas.

Como sugestões para futuras melhorias, consideramos a possibilidade de aprimorar a interface do usuário, explorando opções mais visualmente atraentes e intuitivas. Além disso, adicionar elementos lúdicos adicionais, como eventos aleatórios ou missões secundárias, pode tornar o jogo ainda mais envolvente. A implementação de competidores virtuais controlados por inteligência artificial também pode adicionar desafios extras e aumentar a longevidade do jogo. Por fim, expandir o escopo do jogo para incluir mais recursos, redes de transporte complexas e a opção de jogar em modo multiplayer online pode atrair um público mais amplo e proporcionar uma experiência mais imersiva.

Em suma, o desenvolvimento do jogo de simulação de logística foi uma experiência enriquecedora que nos permitiu aplicar meus conhecimentos de programação e estruturas de dados de maneira prática. Através desse projeto, aprimoramos nossas habilidades de resolução de problemas, trabalho em equipe e planejamento de projetos. Estamos satisfeitos com os resultados alcançados e acredito que o jogo atingiu seus objetivos de ser completo, equilibrado, divertido e desafiador para jogar.

References

- [GG12] Marco Goldbarg and Elizabeth Goldbarg. *GRAFOS: conceitos, algoritmos e aplicações*. Elsevier, 2012.