

2º Avaliação de Estrutura de Dados

1st Tiago José Cardoso Leite
Engenharia de Software
ICEV
Teresina, PI
tiago.leite@somosicev.com

2nd Gustavo Félix do Rego
Engenharia de Software
ICEV
Teresina, PI
gustavo.rego@somosicev.com

3rd Renato César dos Santos Junior
Engenharia de Software
ICEV
Teresina, PI
renato.junior@soomosicev.com

Abstract—Neste projeto construímos um jogo em que o protagonista, Maxwell, viaja de sua cidade inicial, Ubud, até Nargumun levando uma joia de poder. Dentre seus objetivos, o jogo deve ser feito usando os conhecimentos da disciplina de Estrutura de Dados e deve ser divertido e desafiador. Apesar de, nos resultados, o jogo ficou mediantemente desafiador, ainda sim é divertido de explorar o que ele tem a oferecer pelo esforço que os integrantes do projeto tiveram de deixá-lo com um bom toque de humor.

Index Terms—Jogo; Inteligência artificial; Estrutura de dados.

I. INTRODUÇÃO

Nosso Projeto foi iniciado como uma prova pelo Coordenador e Professor Dimmy em sua matéria de Estrutura de Dados, a fim de motivar os alunos a usar os conceitos aprendidos em sala para desenvolver um jogo, mostrando assim as capacidades de sua matéria e despertando mais o interesse dos alunos à importância do seu conteúdo ministrado.

O tempo foi de 50 dias para a entrega do projeto, pois os alunos teriam acesso e tempo para aprenderem a usar as estruturas ensinadas em sala e aplica-las em seu projeto. Nosso grupo foi planejado desde o início do projeto, mas só começamos o planejamento com 25 dias restantes, para que todos os membros pudessem ter uma noção e tempo de estudo para fazer e aplicar o que foi visto em sala.

Com tudo isso em “jogo”, estudamos Java, Estrutura de Dados, e um pouco de interface gráfica para a realização do nosso Game.

II. DESCRIÇÃO DO PROJETO

Como já dito anteriormente, a finalidade do game é fazer com que Maxwell saia de Ubud e vá até Nargumun são e salvo (e que de preferência não vire servo quando chegar lá). O projeto deve se utilizar das estruturas de dados aprendidas em sala de aula, como grafos, árvores de decisões e listas. Então juntamos os Requisitos:

- As cidades serão alocadas em um grafo;
- O mercador terá uma árvore de decisões;
- Maxwell deve conseguir viajar até Nargumun com um pouco de dificuldade;
- A joia deve ter seu poder e limiar alterados conforme o andar do jogo;
- O jogador pode ou não aceitar as missões;
- Só uma missão pode ser feita por vez;
- Maxwell pode abandonar a missão quando preferir;
- Nargumun deve ser o destino final para zerar o jogo.



Fig. 1. Print do sistema em funcionamento.

III. DESCRIÇÃO DA EQUIPE

Depois de duas reuniões, decidimos separar as atribuições de cada integrante de certa maneira:

- Tiago(Volturifox): Responsável pela Classe Cidade (City) e pela construção do Grafo de cidades para a transição entre elas, além de um pouco de interface. Segundo responsável pela programação em pares, usada no projeto.
- Renato: Responsável pelas classes de sistema de Quests e Mercador, utilizando árvore de decisão.
- Gustavo: Responsável pela criação de Maxwell e sua interação com o restante do jogo e todas as suas funções. Principal responsável pela programação em pares, usada no projeto.

IV. METODOLOGIA

Neste trabalho foi usado a metodologia ágil XP para a rotina, ou seja, programação em pares, pequenas versões, propriedade coletiva, integração contínua, testes, cliente presente (foi chamado Lucas Moura, amigo de um dos integrantes do grupo para testar o jogo), projeto simples e reuniões diárias.

Para a parte do Código, foi usado Java como linguagem, como bibliotecas foram usadas: TreeMap, Swing, Scanner, ArrayList e seria usado java Sound.

V. DESCRIÇÃO DAS ESTRUTURAS DE DADOS E ALGORITMOS

Neste projeto, foram usadas três das estruturas de dados estudadas pelos integrantes: lista, grafo e árvore de decisão.

As listas foram usadas para armazenar dados como a posição dos vizinhos que uma cidade tem.

```
private Integer graphPosition;  
public ArrayList<Integer> neighborsPosition = new ArrayList<Integer>();
```

Fig. 2. o exemplo acima está na linha 15 da classe City.

O Grafo foi feito para armazenar as cidades e dar uma liberdade maior de locomoção no jogo. Está localizado na classe CityGraph.

```
private void initialize() {  
    neighborhood = new ArrayList<City>();  
    neighborhood.add(new City(name:"Ubad",powerAmount10,questExist:false,graphPosition0));  
    neighborhood.add(new City(name:"Principality of nekish",powerAmount11,questExist:false,graphPosition1));  
    neighborhood.add(new City(name:"Principality of Gritesth",powerAmount12,questExist:false,graphPosition2));  
    neighborhood.add(new City(name:"Kingdom of lastwatch",2,questExist:false,graphPosition3));  
    neighborhood.add(new City(name:"Grand Duchy of Smalia",powerAmount11,questExist:false,graphPosition4));  
    neighborhood.add(new City(name:"Kingdom of Legood",powerAmount12,questExist:false,graphPosition5));  
    neighborhood.add(new City(name:"Protectorate of Dogrove",powerAmount13,questExist:false,graphPosition6));  
    neighborhood.add(new City(name:"Kingdom of Oldcalia",powerAmount14,questExist:false,graphPosition7));  
    neighborhood.add(new City(name:"Kingdom of Kalb",powerAmount12,questExist:true,graphPosition8));  
    neighborhood.add(new City(name:"Vunese Empire",powerAmount10,questExist:true,graphPosition9));  
    neighborhood.add(new City(name:"Principality of Kaszy",2,questExist:false,graphPosition10));  
    neighborhood.add(new City(name:"Aymar League",powerAmount11,questExist:false,graphPosition11));  
    neighborhood.add(new City(name:"Defalsia",-3,questExist:true,graphPosition12));  
    neighborhood.add(new City(name:"Chandir Sultanate",powerAmount11,questExist:false,graphPosition13));
```

Fig. 3. Classe cityGraph.

A árvore foi feita com a ajuda da biblioteca TreeMap para servir de estrutura de decisão ao mercador. Está localizada na classe Tree.

```
package app;  
import java.util.ArrayList;  
import java.util.TreeMap;  
public class Tree {  
    //this class contains functions that use CDT(coinDistanceTrade) and return C(coinJewelthreshold)  
    public CJ TreeMap<Integer, Integer> choices = new TreeMap<>();  
    //coins  
    choices.put(key:20, value:5);  
    //distance  
    choices.put(key:10, value:3);  
    choices.put(key:30, value:3);  
    //trade  
    choices.put(key:5, value:1);  
    choices.put(key:15, value:1);  
    choices.put(key:25, value:1);
```

Fig. 4. Classe Tree.

VI. RESULTADOS

Teste de Usuário: Foi convidado Lucas Moura, um amigo próximo de um dos integrantes do grupo, para que fosse feito a testagem do Jogo. Inicialmente, Lucas teve dificuldade para entender o funcionamento mostrando o quanto é necessário uma reavaliação e mudança de interface para o jogo ficar mais convidativo. Porém, em questão de gameplay, Lucas demonstrou bastante animação ao tentar realizar uma das missões e receber a “Glória dos Retornados”. Infelizmente, Lucas gostou do jogo mas comentou bastante sobre não ser

intuitivo o suficiente, que será levado em consideração para futuras atualizações. Equipe: A equipe ficou muito animada por ter feito um jogo e usado as estruturas de dados, apesar das dificuldades todos ficaram satisfeitos com o resultado e ansiosos para implementarem futuras atualizações no game. Programa: O jogo consegue rodar e cumpre com seus requisitos, além de ter um final.

VII. CONCLUSÃO

Com base nos resultados e nas metodologias apresentadas, destacam-se os benefícios da abordagem adotada, como o desenvolvimento rápido e iterativo, a melhor adaptabilidade às mudanças, o foco na qualidade do produto e a interface gráfica. Além disso, são discutidas possíveis direções para pesquisas futuras, explorando outras bibliotecas ou metodologias para aprimorar ainda mais o desenvolvimento de jogos em Java.

REFERENCES

- [1] Redescobindo o Swing toolkit - Revista easy Java Magazine 20 - Parte 1.” [Online]. Available: devmedia.com.br. [Accessed: Junho, 2023].
- [2] Java Swing: JTextField e JFormattedTextField.” [Online]. Available: devmedia.com.br. [Accessed: Junho, 2023].
- [3] GOODRICH, Michael T.; TAMASSIA, Roberto. Estruturas de Dados e Algoritmos em Java. Bookman Editora, 2013.
- [4] Canal Bro Code. ”Java GUI: Full Course (FREE)” [Online]. Available: <https://youtu.be/Kmgo00avvEw>. [Accessed: Junho, 2023].
- [5] w3schools.com. ”Java LinkedList.” [Online]. Available: https://www.w3schools.com/java/java_linkedlist.asp. [Accessed: Junho, 2023].
- [6] Canal Henrique Lima. ”Grafos - Conceitos e Implementação Simples (Java)” [Online]. Available: <https://youtu.be/Drd8rzVTUVo>. [Accessed: Junho, 2023].