


Implementação do jogo Dominó Monetário utilizando os algoritmos de Busca Gulosa e A*

1° Amanda Rodrigues Vieira
Juquiá, Brasil
amavie@alunos.utfpr.edu.br

2° Francisco Correa Neto
São José da Boa Vista, Brasil
franet@alunos.utfpr.edu.br

3° Júlio César Guimarães Costa
São José dos Campos, Brasil
julioesargcosta123@gmail.com

4° Luis Gustavo Mazzetto Campregher
Indaiatuba, Brasil
campregher@alunos.utfpr.edu.br

5° Plinio Villas Boas
São Paulo, Brasil
 <https://orcid.org/0000-0003-4631-4956>

Abstract—The present work has the goal to demonstrate a game of monetary dominoes with the application of two searching algorithms, the Greedy Search and the A* search.

Index Terms—Dominoes, intelligent agents, a* search, greedy search

I. INTRODUÇÃO

Dominó é um jogo de mesa com peças que tem uma das faces marcadas por pontos indicando valores numéricos, sendo que esta face pode possuir apenas dois valores numéricos, separados pela metade da peça. Na forma clássica do jogo, são sete números (de zero a seis) combinados entre si, o que resulta em 28 peças.

O jogo de dominó monetário segue as mesmas regras do dominó, com a diferença de que os valores numéricos das peças são definidos por cédulas e moedas. Dessa forma, como existe uma gama maior possibilidades quando se trata de dinheiro, o número total de peças se torna 91, definido pela fórmula de combinação:

$$C(13, 2) + 13 = C(14, 2) = 91 \quad (1)$$

O jogo começa com cada jogador recebendo 13 peças e o restante das peças fica no banco, onde o jogador pode comprar caso não tenha a peça para jogar. Se o jogador da vez possui uma peça com valor correspondente a um dos lados das peças da mesa, ele a joga. Se não possuir, compra peças no banco até ter a que precisa. O jogo acaba quando um jogador esgota suas peças, e uma pontuação é atribuída a ele baseada na soma das peças do adversário.

O objetivo do presente trabalho é demonstrar a implementação de dois algoritmos com agentes inteligentes que realizam as chamadas Busca Gulosa e Busca A*. A primeira toma decisões pensando no contexto local, ou seja, na melhor escolha para o momento. Já o algoritmo A* visualiza a situação geral para tomar uma escolha.

II. DESCRIÇÃO DO PROBLEMA

O problema inicial do projeto é a confecção de agentes inteligentes com a capacidade de jogar corretamente o jogo de Dominó Monetário. A partir disso, foram escolhidos dois algoritmos de busca, sendo eles: Busca Gulosa e Busca A*.

A. Busca Gulosa (Greedy Search)

O algoritmo de Busca Gulosa utiliza de meios locais para a resolução do problema apresentado. Em cada execução, o algoritmo busca o melhor caminho utilizando somente uma heurística simples e imediata.

Fazendo um paralelo entre busca de melhor caminho nos grafos, o Busca Gulosa buscaria somente pensando em minimizar o caminho naquele instante de execução, o que muitas vezes não o leva à solução ótima daquele problema.

B. Busca A* (A Star Search)

O algoritmo de Busca A* é amplamente utilizado em grafos a fim de encontrar o caminho ótimo entre dois pontos. Diferentemente do Busca Gulosa, o A* usa métodos de busca que consideram não somente o estado atual, mas também uma heurística que, utilizando os dados atuais, projeta caminhos futuros procurando não somente o melhor local, mas procurando encontrar a solução ótima àquele problema.

Utilizando o mesmo paralelo com grafos, ele utiliza não somente a menor distância entre o vértice atual e o próximo vértice, ele também utiliza de uma heurística que projeta uma linha reta entre os grafos possíveis de serem usados e o vértice final, assim maximizando as chances desses grafos serem mais próximos do fim.

C. Designação das heurísticas

A partir da teoria acerca dos algoritmos, foram estabelecidos paralelos entre as heurísticas comuns desses algoritmos com o jogo de Dominó. As principais heurísticas utilizadas foram:

- 1) Jogar as peças duplas primeiro;
- 2) Jogar as peças de valor mais alto;
- 3) Analisar as peças na mesa e jogar peças que já foram utilizadas com frequência;

Essas heurísticas foram baseadas em regras do dominó e também com relação à regra da poupança. As regras 1 e 2 foram baseadas na ideia de que ganha o jogo aquele que tenha a maior poupança. Logo, ao jogar peças que contenham o maior valor da soma entre seus lados, o jogador diminui o valor total ao final do jogo, e mesmo que perca, ele minimiza o valor da poupança do jogador adversário.

Já na regra 3, utilizada pelo A*, ele analisa as peças da mesa, para assim jogar as peças que tem a maior chance de travar o inimigo. Isso por conta da chance do outro jogador ter aquela peça em suas mãos ser menor e também pois como a chance daquela peça ser usada diminui, talvez o algoritmo tenha menos chances de utilizá-la em um momento futuro do jogo, assim impedindo que o próprio algoritmo se trave.

III. IMPLEMENTAÇÃO

A. Metodologia da Implementação

Para a implementação do jogo foi utilizada a IDE Pycharm em conjunto com a linguagem de programação Python, assim como a biblioteca *open source* Arcade [1]. Esta utiliza-se da API OpenGL para construção gráfica dos objetos em tela. No início do desenvolvimento foi definida uma ferramenta para organização de pessoal de cada integrante, chamada *Anaconda Environments*, que serve para o isolamento do projeto, das bibliotecas e tecnologias. Foi definida também uma ferramenta de organização para todo o grupo chamada Github, servindo para o compartilhamento das atualizações desenvolvidas no decorrer do projeto entre os integrantes.

B. Fluxograma do Projeto

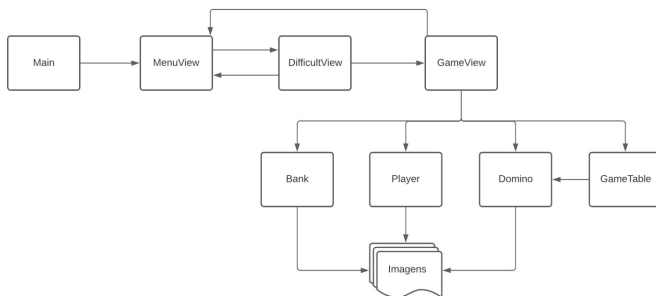


Fig. 1. Fluxograma

C. Fases da Implementação

- Criação das Imagens:
 - Criação dos dominós.
 - Criação do plano de fundo.
 - Criação dos marcadores de posição (*placeholders*).
 - Criação da logo e identidade.
- Implementação dos Estrutura Principal (*Main*):
 - Implementação das constantes.
 - Implementação da janela de visualização.
 - Adição da Licença de Uso.
 - Adição do Leia-me.
- Implementação dos Objetos (Objetos em Tela):
 - Implementação do objeto Banco (*Bank*).
 - Implementação do objeto Domino.
 - Implementação do objeto Mesa (*GameTable*).
 - Implementação do objeto Jogador (*Player*).
- Implementação dos Telas (*Views*):
 - Implementação da tela de Menu (*MenuView*).

- Implementação da tela de Dificuldade (*DifficultView*).
- Implementação da tela do Jogo (*GameView*).

D. Implementação dos Agentes Inteligentes

A implementação dos agentes inteligentes foi realizada dentro do objeto Jogador (*Player*) e se baseou em uma heurística de prioridades de escolha onde, através de uma analogia feita entre os algoritmos (Busca Gulosa e Busca A*) aplicados em grafos e o problema da melhor jogada proposto, foram definidas as seguintes relações:

- Os vértices dos grafos seriam as jogadas feitas pelo agente no decorrer da partida;
- O caminho percorrido nos grafos seriam as jogadas antecedentes e sua adaptação;
- O objetivo de alcançar um vértice específico do grafo seria relacionado à vitória da rodada.

E. Lista de prioridades dos Agentes

Das peças possíveis de serem jogadas na mão do agente, ele irá priorizar:

- Busca Gulosa:
 - 1) Jogar as peças duplas primeiro;
 - 2) Jogar as peças de valor mais alto.
- AStar (A*):
 - 1) Analisar as peças na mesa e jogar peças que já foram utilizadas com frequência;
 - 2) Jogar as peças duplas;
 - 3) Jogar as peças de valor mais alto.

IV. RESULTADOS

A partir destes levantamentos, o jogo foi desenvolvido aplicando tais táticas, com base nos algoritmos Busca A* e Busca Gulosa. O jogo foi batizado como Dominó Monetário, contando com:

- Uma tela de menu, para início e saída do aplicativo;
- Tela de escolha de dificuldade, com Fácil (Busca Gulosa) e Difícil (Busca A*);
- Tela de carregamento;
- Tela do jogo.

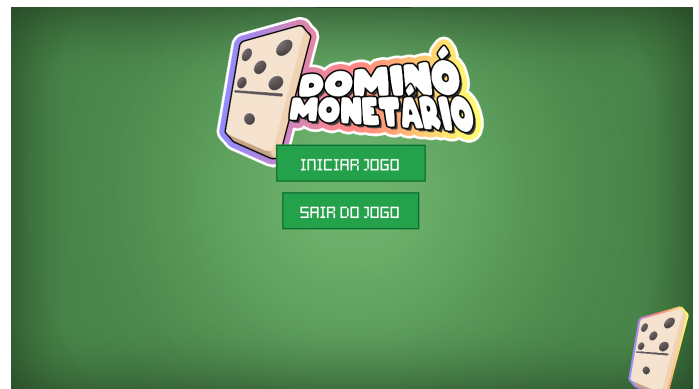


Fig. 2. Tela de Menu

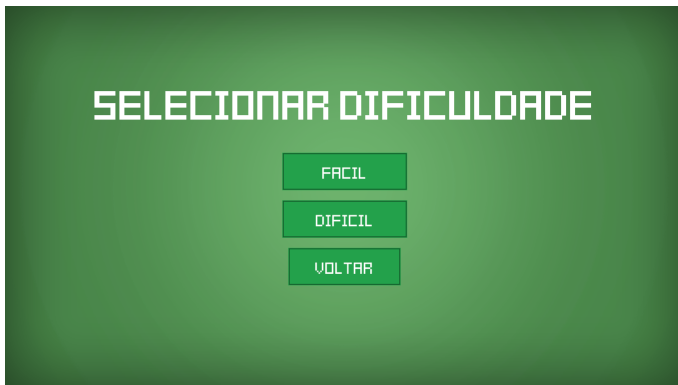


Fig. 3. Escolha de Dificuldade

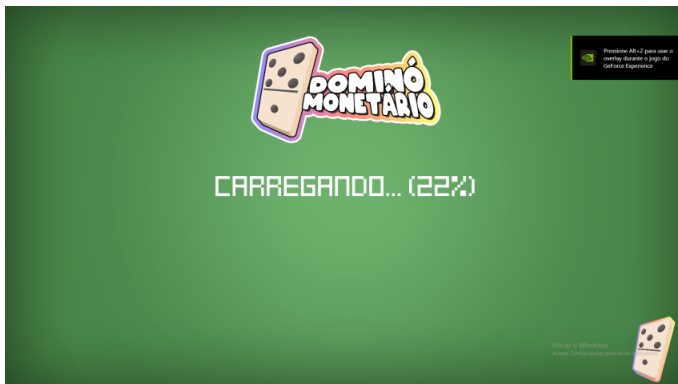


Fig. 4. Tela de Carregamento

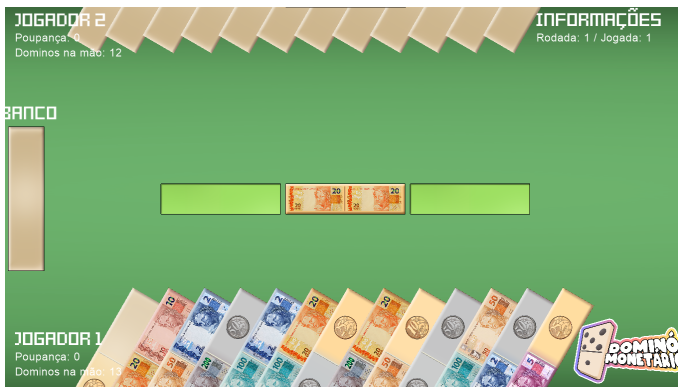


Fig. 5. Início de Jogo

A. Interações com a tela

As interações possíveis com a tela são: a de clique, onde é possível clicar nas opções do menu e dificuldade e também nas peças a serem jogadas; a de *scroll*, onde é possível ir para a esquerda e para a direita do tabuleiro e visualizar todas as peças que já foram jogadas; e o botão `esc`, que fecha o jogo quando ele acaba.

B. Situações de jogo

A primeira peça a ser jogada deve ser uma dupla. Assim, o jogador que tiver a maior dupla inicia o jogo. Se o jogador da vez possuir uma peça jogável, ele a joga no respectivo lado. Se não possuir, ele deve comprar peças no banco até possuir uma peça jogável.

Se o jogador continuar comprando peças até o banco se esgotar, o jogo é considerado "fechado" e vence aquele que possuir menos peças em mãos. Quando isso acontece, o ganhador recebe como pontuação a soma das peças do adversário e este valor vai para a poupança.

Ao final de três rodadas, vence o jogador com maior valor acumulado na poupança.

C. Comparação dos algoritmos

Ao jogar o jogo algumas vezes, foi possível notar que o agente inteligente ganhava mais vezes no modo Difícil do que no Fácil. Assim, concluímos que o desempenho do algoritmo de Busca A* foi melhor do que o de Busca Gulosa.

V. CONCLUSÃO

Este trabalho apresentou duas implementações de agentes inteligentes para serem usados no jogo desenvolvido, "Dominó Monetário". O primeiro deles, um algoritmo de Busca A* (esse referente a nível de dificuldade "Difícil"), considera a peças da mesa, a frequência em que foram utilizadas e prioriza peças duplas e depois as de maior valor. O segundo, um algoritmo de Busca Gulosa, joga peças duplas primeiro e em seguida peças de maior valor. Além disso, foi desenvolvido de forma completa o jogo de "dominó monetário" em linguagem Python, utilizando a biblioteca *Arcade*. Este jogo apresenta interface, níveis de dificuldade, peças baseadas no sistema monetário brasileiro, logotipo, múltiplas telas e mesa de jogo. Durante o desenvolvimento, alguns *bugs* foram encontrados e eliminados conforme apareceram.

Como resultado final, o jogo que foi entregue junto deste trabalho já pode ser disponibilizado e usado como o esperado pela atividade proposta, porém, ainda podem haver melhorias visuais e de apresentação.

REFERÊNCIAS

- [1] Vincent Craven, Paul. "How to create a 2D game with Python and the Arcade library", 24 de Abril de 2018. Disponível em: <https://opensource.com/article/18/4/easy-2d-game-creation-python-and-arcade>. Acesso em: 20 mar. 2022.
- [2] Silva, A. G. "Sistemas Inteligentes". Disponível em: <http://www.inf.ufsc.br/alexandre.goncalves.silva/courses/14s2/ine5633/slides/aula07.pdf>. Acesso em: 28 mar. 2022.