

**UNIVERSIDADE DO MINDELO**

**DEPARTAMENTO DE ENGENHARIA INFORMÁTICA**

**CURSO DE LICENCIATURA EM**

**Engenharia Informática e Sistemas Computacionais**

**TRABALHO DE INTRODUÇÃO Á PROGRAMAÇÃO**

**Ano letivo 2023/2024 – 1º Ano**

**Tema: Implementação do Jogo REVERSI em Python (Relatório Técnico)**

**Autor: Kelly Fortes, N.º 5727, Ricardo Pires, N.º 5731**

**Orientador: Mcs. Emanuel Vieira**

**Mindelo, 2024**

**Introdução**

No âmbito da unidade curricular Introdução a Programação foi-nos solicitado a implementar o jogo Reversi em python. Ao longo deste trabalho será apresentado o passo a passo da implementação do projeto Reversi.

O Reversi surgiu na Inglaterra por volta de 1880, também é conhecido como Othello. É um jogo de tabuleiro para dois jogadores, há apenas um único tipo de peça, a qual possui duas faces, normalmente de duas cores diferentes. Uma vez em que a peça foi colocada em uma das casas do tabuleiro, ela não poderá ser retirada, apenas poderá ser modificado a cor da peça, basta cercá-la com duas peças de cor oposta. A vitória será dada para o jogador que tiver mais peça da sua cor.

O Reversi é um jogo que requer um alto desenvolvimento de ideias lógicas, táticas e estratégias. Infelizmente em Cabo Verde, tal jogo é pouco difundido e conhecido, porém em outros países há inúmeros clubes, fundações e até mesmo campeonatos internacionais.

O objetivo é implementar o jogo Reversi em python, utilizando conceitos de programação orientada a objetos e algoritmos de busca. A implementação compreende a criação de um tabuleiro, a definição das regras do jogo, e a criação de uma interface gráfica para facilitar a interação com o jogo.

**Objetivos do Jogo Reversi**

O jogo Reversi é jogado num tabuleiro de 8x8 em que dois jogadores têm peças ou “pedras” de cores diferentes.

O objetivo do jogo é acabar com o maior número de peças da cor escolhida.

A única jogada permitida é a que “ensanduícha” uma ou mais peças do adversário entre duas peças suas. Quando coloca uma peça, todas as peças “ensanduichadas”. Do adversário ficam da sua cor. Se não puder colocar uma peça, você tem de passar a vez.

O jogo termina quando não há mais espaços no tabuleiro ou quando ambos os jogadores passam a vez na mesma jogada. O vencedor é aquele que, no fim, tiver mais peças no tabuleiro.

**Algumas regras do Jogo Reversi**

* O preto sempre faz o movimento primeiro;
* Se um jogador não puder flanquear e virar ao menos um disco do oponente, ele perde a vez e seu oponente joga outra vez;
* Um disco pode flanquear vários discos em uma ou mais linhas em várias direções ao mesmo tempo – horizontal, vertical ou diagonal. Uma “linha” é definida como um ou mais discos em uma linha reta contínua;
* Um jogador não pode passar por cima do disco de sua cor(s) para flanquear um disco oposto a ele;
* O(s) disco(s) deve(m) ser flanqueado(s) como resultado direto de uma jogada sendo virado(s) na linha do disco posicionado;
* Todos os discos flanqueados em qualquer movimento devem ser virados, mesmo que fosse uma vantagem para o jogador não virá-los;
* Uma vez que um disco é colocado em um quadrado, ele nunca poderá ser movido para outro quadrado mais tarde no jogo.
* Quando não há mais possibilidade de movimento para ambos os jogadores, o jogo termina. Os discos são contados e o jogador com a maioria dos discos da sua cor é o vencedor da partida.

**Estrutura do Código**

Essa estrutura de código que se segue abaixo, são apenas algumas partes do código que consideramos ser mais importante para que o jogo funcionasse.

1. **Inicialização e configuração do jogo**

O código começa importando as bibliotecas necessárias e definindo algumas constantes, como as cores usadas no jogo e o tamanho do tabuleiro. Em seguida, ele inicializa o Pygame e configura a janela do jogo. O tabuleiro do jogo é representado como uma lista de listas, onde cada elemento é uma célula do tabuleiro.

1. **Definição do tabuleiro**

O tabuleiro é inicializado com quatro peças no centro, duas brancas e duas pretas, em uma configuração específica.

1. **Função movimento\_válido (i, j, jogador)**

Esta função verifica se um movimento é válido para o jogador atual. Ela verifica se a célula está dentro do tabuleiro e está vazia, e então verifica todas as direções a partir dessa célula. Se houver uma peça do oponente na próxima célula em uma direção, ela continua seguindo essa direção até encontrar uma peça do jogador atual. Se encontrar, o movimento é válido.

1. **Função atualizar\_tabuleiro (i, j, jogador)**

Esta função atualiza o tabuleiro depois que um movimento é feito. Ela coloca a peça do jogador na célula e então verifica todas as direções a partir dessa célula. Se houver uma peça do oponente na próxima célula em uma direção, ela continua seguindo essa direção até encontrar uma peça do jogador atual. Quando encontra, ela muda todas as peças do oponente que estão entre a peça do jogador atual e a célula do movimento para a cor do jogador atual, efetivamente capturando as peças do oponente.

1. **Loop principal do jogo**

Este é o Loop principal do jogo, onde a maior parte da lógica do jogo acontece. Ele lida com os eventos do pygame, como o fechamento da janela do jogo. Se o evento for um clique do mouse, ele verifica se o movimento é válido e, em caso afirmativo, atualiza o tabuleiro.

**Funcionalidades Implementadas**

Visto que baseamos nos conhecimentos adquiridos na sala de aula, não foi implementado nenhuma outra funcionalidade além daquilo que foi aprendido durante as aulas.

Houve alguma dificuldade em algumas partes do código, por esse mesmo motivo não foi possível colocar um menu, ou uma imagem de fundo, entre muitos outros tipos de funcionalidades que poderia ser implementada.

**Conclusão**

Em conclusão, a implementação de jogo Reversi em python foi um desafio empolgante e enriquecedor. Ao longo do desenvolvimento, pudemos aplicar e aprimorar nossos conhecimentos em programação. Além disso, o projeto proporcionou a oportunidade de aprofundar nosso entendimento sobre as regras e as estratégias do jogo Reversi.

Este trabalho permitiu expandir nossas habilidades de programação, bem como nos desafiou a encontrar soluções inovadoras para os problemas encontrados.

Em última análise, a implementação do jogo em python foi uma experiência gratificante que nos permitiu aplicar e expandir nossos conhecimentos em programação. Através deste trabalho, possível explorar e aplicar diversos conceitos de programação, ao mesmo tempo que se criou uma versão funcional e interativa do jogo Reversi.

Ao final, conseguimos produzir uma versão funcional e interativa do jogo, proporcionando uma experiência de aprendizagem valiosa e divertida.

**Melhorias Futuras**

Para o futuro espera-se cumprir todos os objetivos, visto que houve algumas complicações em implementar algumas partes essenciais do jogo, como por exemplo explorar a lógica por trás da implementação de uma inteligência artificial capaz de jogar contra o jogador.

Também em implementar um código para atualizar o jogo, toda vez que o mesmo terminasse. Também houve dificuldades em colocar um placar no jogo, o que daria uma nova interação com o jogo.

**Referências Bibliográficas**

[www.worldothello.org](http://www.worldothello.org)

<https://www.mathsidfun.com/games/reversi.html>

<https://www.ic.unicamp.br>

https://www.orac-games.com