

Jogos: Tabuleiro de Números

Enzo C. da S. Barbosa

UEFS – Universidade Estadual de Feira de Santana

Av. Transnordestina, s/n, Novo Horizonte

Feira de Santana – BA, Brasil – 44036-900

caua7uefs@gmail.com

Abstract. *The games league of the Institute of Electrical and Electronic Engineers at the State University of Feira de Santana, in search of new members, challenged those entering the Computer Engineering course to develop a game using Python. The game to be developed consists of a board of $N \times N$ spaces, where 2 players alternately place a number in a space until one of the players reaches a drawn sequence before starting the game. The game will also feature a ranking of victories, option to enable a special match and saving matches in progress. The system must be attractive and easy to use to attract maximum new participants to a gaming league.*

Resumo. *A liga de jogos do Instituto de Engenheiros Elétricos e Eletrônicos da Universidade Estadual de Feira de Santana na busca por novos integrantes, desafiaram os ingressantes do curso de Engenharia de Computação a desenvolver um jogo utilizando Python. O jogo a ser desenvolvido consiste em um tabuleiro de $N \times N$ casas, onde 2 jogadores de forma alternada, colocam um número em uma casa até um dos jogadores atingir uma sequência sorteada antes de iniciar a partida, o jogo também contará com um ranking de vitórias, opção de permitir uma jogada especial e salvamento de partidas em progresso. O sistema deverá ser atrativo e de fácil uso para atrair o máximo de novos participantes para a liga de jogos.*

1. Introdução

A liga de jogos do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) da Universidade Estadual de Feira de Santana (UEFS) propôs um desafio de desenvolvimento para os alunos ingressantes no curso de Engenharia de Computação, com o objetivo de atrair novos integrantes para a liga. Para isso foi solicitado a criação de um jogo, que consiste em um tabuleiro de $N \times N$ casas, onde 2 jogadores de forma alternada colocam um número entre 1 e N^2 , ganha o jogador que preencher uma linha, coluna ou diagonal com uma sequência de números. O jogo também deve possuir um sistema de Ranking, salvamento de partidas e opções de jogadas especiais.

O objetivo deste relatório é descrever o desenvolvimento de um jogo de tabuleiro na linguagem de programação Python, como solução para o problema descrito acima. O jogo serve para promover a cultura dos games e atrair novos integrantes para a liga de jogos do IEEE UEFS. O programa consiste numa matriz que representa o tabuleiro, onde o jogador escolhe uma casa e em seguida um número que preencherá essa casa, a cada jogada o jogo verifica se uma sequência foi completada e vence quem completar a sua sequência objetivo primeiro. O jogo conta ainda com sistema de Ranking, seleção de dificuldade e salvamento de partidas.

Para solucionar este problema, foi necessário solucionar algumas questões: 1) Como criar o tabuleiro e exibi-lo no terminal?; 2) Como sortear os objetivos?; 3) Como verificar se um dos jogadores atingiu o objetivo?; 4) Como salvar o estado do jogo e como manter registrado o número de vitórias por jogador?; 5) Como diferenciar as cores dos jogadores?; 6) Como alternar qual jogador inicia o jogo?.

A solução está organizada da seguinte forma:

1. Utilização de Matrizes;
2. Utilizando a biblioteca “Random”;
3. Comparando as casas do tabuleiro com as sequências definidas;
4. Armazenando em arquivos externos;
5. Adicionando os códigos de cor a cada jogada;
6. Utilizando a biblioteca “Random”.

2. Metodologia

Nesta seção, será descrito os requisitos e funcionalidades do sistema, de acordo com o que foi solicitado pela liga para atender atrair novos integrantes, as questões e decisões abordadas durante as sessões tutoriais, a descrição do algoritmo desenvolvido, a ordem de codificação e as ferramentas utilizadas na elaboração do projeto.

2.1. Requisitos e Funcionalidades

1) Tabuleiro: O sistema deve possuir um tabuleiro de $N \times N$ casas com números entre 1 e N^2 disponíveis para jogar, sendo divididos inicialmente em: Fácil (3 X 3, 1 a 9), Médio (4 X 4, 1 a 16) e Difícil (5 X 5, 1 a 25). 2) Objetivos: O sistema deve sortear os objetivos de cada jogador, sendo eles formar uma sequência em linha, podendo ser na horizontal, vertical ou diagonal e seguindo o padrão de leitura da esquerda para a direita e de cima para baixo, as sequências podem ser ascendente, descendentes, pares ou ímpares. 3) Jogabilidade: O jogo deve ser jogado em turno no mesmo computador, durante o sorteio, os jogadores não poderão saber o objetivo do outro, também é necessário que haja diferença de cor entre a jogada de cada um dos jogadores. 4) Jogada Especial: O sistema deve ter uma opção de ativar uma jogada especial no antes de começar o jogo, que permita o jogador limpar uma linha ou coluna do tabuleiro e em seguida realizar uma jogada normal. 5) Interface: O

sistema deve possuir um interface, em modo texto, que seja atraente, cores e sons são opcionais, no entanto a sonorização não pode depender de arquivos externos. 6) Ranking: O sistema deve manter um registro com nome dos jogadores e quantidade de vitórias. 7) Salvamento: O sistema deve permitir os jogadores de parar o jogo a qualquer momento da partida e continuar em outro momento, salvando o estado da partida. 8) Modularização: 6) O código do sistema deve ser dividido em funções e permitir atualizações futuras.

2.2. Questões e Soluções

1) Como criar o tabuleiro e exibi-lo no terminal?: O tabuleiro é criado por meio de uma matriz de dicionário, onde cada casa é representada por uma chave e o número associado a casa equivale ao valor da chave no dicionário. Para exibir o tabuleiro foi utilizado uma função para formatar a exibição no terminal, fazendo com que ela seja exibida no formato de um tabuleiro.

2) Como sortear os objetivos?: Os objetivos são sorteados por meio da função “choice” da biblioteca “Random”, um dos objetivos é sorteado entre os 4 disponíveis e é exibido para o primeiro jogador, em seguida o terminal é limpo e o objetivo do próximo jogador é sorteado.

3) Como verificar se um dos jogadores atingiu o objetivo?: O código utiliza uma lista de sequências de casas que poderiam formar uma sequência, para percorrer o tabuleiro e criar um string com cada sequência dessas no tabuleiro, em seguida verifica se alguma das sequências está presente na string de objetivo.

4) Como salvar o estado do jogo?: O jogo é salvo em um arquivo .pkl, criado por meio da biblioteca “pickle”. Quando o jogador seleciona a opção de salvar, o arquivo é aberto e a matriz do tabuleiro é escrita no arquivo, essa função também é utilizada para salvar o ranking ao final de cada partida.

5) Como diferenciar as cores dos jogadores?: A realizar um jogada, o número escolhido pelo jogador é concatenado com duas strings, uma referente ao código da sua cor e outra referente ao código de cor do tabuleiro, é adicionado também a quantidade de espaços necessários para centralizar o valor no tabuleiro. Ao final a string concatenada é associada a chave da casa somente no dicionário do tabuleiro de visualização, a string final fica da seguinte forma: “\033[32m N \033[m”

6) Como alternar qual jogador inicia o jogo?: Os nomes dos jogadores são adicionados a uma lista e utilizando a função “shuffle” da biblioteca “Random” os nomes são embaralhados, aleatorizando quem inicia o jogo.

2.3. Desenvolvimento do Sistema

A construção do algoritmo iniciou-se com a criação dos dicionários que seriam usados como referência para os 3 tamanhos de tabuleiro e as 3 sequências de verificação. Na função de criação do dicionário de tabuleiros, o tamanho é recebido como parâmetro e de acordo com ele é definido o dicionário, sendo as chaves e os valores a letra de cada casa. Já na função de criação do dicionário de sequências, o tamanho também é recebido como

parâmetro, porém as chaves são códigos referentes a sequência e o valores são listas com todas as casas que formam uma sequência para linha, coluna ou diagonal.

A criação do menu principal foi a etapa seguinte, a função do menu principal possui um laço de repetição “while”, onde estão todas as opções do menu, esse laço só é encerrado quando o usuário seleciona a opção de encerrar o programa, que finaliza o programa e informa ao usuário que o programa foi encerrado. A partir do menu principal é possível iniciar um novo jogo, carregar um jogo novo, visualizar o ranking, visualizar regras e encerrar o jogo.

A opção de iniciar novo jogo chama a função executar jogo, que exibe um menu com as dificuldades disponíveis para o usuário escolher, a dificuldade define os parâmetros de tamanho, além de definir os tabuleiros de visualização e de validação. O parâmetro de tamanho é utilizado para definir as casas e os números disponíveis. Em seguida é chamada a função de sortear os objetivos, o primeiro jogador informa seu nome, o programa sorteia um dos quatro objetivos para ele e em seguida limpa o terminal, repetindo o mesmo processo para o segundo jogador, ao final pergunta se a jogada especial será ativada. Após o sorteio, a ordem de jogadas é aleatorizada utilizando a função “shuffle” da biblioteca Random.

O jogo inicia após a dificuldade ser selecionada, os nomes serem registrados, os objetivos sorteados e a ordem ser aleatorizada. A execução ocorre toda dentro de um laço de repetição While, primeiro o tabuleiro é exibido utilizando uma função para formatar o dicionário no formato de um tabuleiro, em seguida é chamada a função de realizar jogadas. Durante a realização das jogadas, o jogador informa uma casa do tabuleiro, se for informado os números ‘45’ ou ‘53’, chama a função de jogada especial e de salvamento, respectivamente, esses números representam o valor hexadecimal das letras ‘E’ e ‘S’. Ao final da jogada, o número selecionado é associado a casa selecionada no tabuleiro de validação e no de exibição é adicionado o número concatenado com o código de cor do jogador e com os espaços vazios necessários para centralizá-lo.

As jogadas são repetidas até que todas as casas sejam preenchidas ou que um dos jogadores atinja o seu objetivo. A validação de objetivo é feita utilizando um lista de com as casas que forma as sequências referentes às linhas, colunas e diagonais no tabuleiro, o código percorre cada lista de sequências, inicializa uma lista com os valores do tabuleiro, para cada casa na sequência, e adiciona o valor correspondente do tabuleiro de validação à essa lista, converte a lista de valores do tabuleiro em uma string onde os valores são separados por vírgulas e adiciona essa string à uma outra lista com as sequências do tabuleiro. Para cada sequência construída do tabuleiro, verifica se a sequência está presente no primeiro conjunto de objetivos, se estiver, retorna esse conjunto de objetivos. Se não estiver no primeiro conjunto, verifica no segundo conjunto de objetivos, se estiver, retorna esse conjunto de objetivos. Se nenhuma sequência corresponder a um dos objetivos, retorna False. Quando um objetivo é completado, o jogo é encerrado, o nome do ganhador é exibido na tela e a vitória é adicionada ao ranking.

A opção visualizar ranking chama a função de leitura e exibição do dicionário contendo o ranking, utilizando a função “open” e a biblioteca Pickle o dicionário é lido de um arquivo no formato “pkl” e em seguida exibe uma lista numerada com os 10 jogadores com mais vitórias. Já a função visualizar regras, exibe todas as regras do jogo.

Todo o desenvolvimento do projeto foi feito no sistema operacional Windows 11 Home Versão 23H2, na versão 3.12 do Python, utilizando a IDE Visual Studio Code v1.89.

3. Resultados e Discussões

O jogo desenvolvido para a liga de jogos do IEEE UEFS foi dividido em 4 partes principais: O menu principal, o Ranking, a execução do jogo e o carregamento da partida salva. A partir do menu principal é possível iniciar um novo jogo, continuar um jogo salvo, visualizar o ranking, exibir as regras ou encerrar o programa.

A opção iniciar novo jogo, solicita o nome dos jogadores, sorteia os objetivos de cada um, pergunta se deseja ativar a jogada especial e em seguida inicia o jogo. Na opção de continuar jogo, deveria ser possível carregar um jogo salvo em um arquivo e continua-lo de onde parou, no entanto houveram problemas na implementação dessa funcionalidade e não foi possível de implementá-la, então é exibido para o usuário a seguinte mensagem: “Infelizmente essa função não foi implementada ainda :(”. A opção visualizar ranking lê um arquivo com os nomes no ranking e exibe os 10 jogadores com mais vitórias. Já a função exibir regras, mostra uma lista com todas as regras do jogo.

Em todas as entradas do código é feita uma validação que só permite que o programa avance quando o usuário inserir uma entrada válida, caso seja inválida o programa exibe uma mensagem de erro e solicita que a entrada seja inserida novamente. São consideradas entradas inválidas letras nas entradas de numero, espaços vazios e números que estejam fora da faixa de opções disponíveis.

Os testes foram realizados com objetivos e jogadas predefinidas para verificar o funcionamento das verificações e em seguida foram feitos testes com o código normal para testar o comportamento em diversas situações. Os principais erros encontrados foram durante as validações e na exibição do terminal. Nas validações ocorreram diversos erros devido a sequências que poderiam ocorrer em mais de um objetivo ao mesmo tempo ou em situações onde ambos os jogadores completavam as suas sequências ao mesmo tempo. Para solucionar os erros de validação foi necessário adicionar vírgulas nas sequências usadas como referência para a validação, evitando com que houvesse sequências de um objetivo dentro do outro. Na exibição houveram erros relacionados a diferenciação de cor, uma vez que ao adicionar os códigos de cor, o alinhamento por meio de f-string deixava de funcionar. Para solucionar a exibição foi necessário adicionar um cálculo que verifica o tamanho da numero informado e adicionasse a quantidade de espaços vazios necessários para centralizá-lo.

4. Conclusão

O sistema desenvolvido atende todos os requisitos mínimos, exceto o salvamento e carregamento de partida, solicitados pela liga de jogos do IEEE UEFS para atrair novos participantes, permitindo a criação de um tabuleiro de $N \times N$ casas, sorteio de objetivos, jogadas alternadas, diferenciação visual dos jogadores, possibilidade de jogadas especiais, e registro do ranking de vitórias, além de ser estruturado em funções permitindo a visualização clara do código-fonte e facilitando possíveis atualizações futuras. Futuramente o programa poderia ser melhorado adicionando o salvamento e carregamento de partidas, a possibilidade de tabuleiros maiores e outras habilidades especiais.