Tabuleiro de números

Lucas Oliveira da Silva

Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, S/N. Feira de Santana, Novo Horizonte – BA, Brasil – 44036-900.

lucasoliveiradaailva@gmail.com

Resumo. *Os jogos ficam populares a cada dia que passa, o desenvolvimento de um jogo é uma tarefa bastante desafiadora. Tendo em vista esse desafio, os alunos da disciplina MI algoritmos, foram solicitados para desenvolver um jogo de tabuleiro com números. Dessa forma, este relatório traz uma breve explicação do processo de elaboração desse sistema: metodologia adotada, dificuldades ao longo da elaboração, bem como os resultados e possíveis melhorias.*

1. Introdução

Na sociedade atual, os jogos se tornam cada vez mais presente na cotidiano do público. Segundos dados da Pesquisa Game Brasil (PGB), principal levantamento do setor, 74,5% da população brasileira joga jogos eletrônico. Jogos de tabuleiro estimulam o desenvolvimento cognitivo e social das pessoas, uma vez que é necessário criar estratégias e analisar possibilidades em tempo real.

Nesse sentido, os alunos da disciplina MI Algoritmos da Universidade Estadual de Feira de Santana, UEFS, receberam uma solicitação do grupo da Liga de Jogos do IEEE UEFS para desenvolver, individualmente, um jogo de tabuleiro com números. O software deveria atender os seguintes requisitos básicos presentes na lista abaixo:

1.º. Salvar e carregar os jogos;

2.º. Salvar e carregar o ranking de pontuação;

3.º. Sortear os objetivos dos jogadores;

4.º. Verificar a sequência dos jogadores;

1.2 Descrição breve da solução

O jogo, desenvolvido na linguagem Python, foi projetado em módulos, sendo cada um responsável por uma tarefa específica. Ele tem como base as seguintes estruturas:

- Dicionários e listas para armazenar e processar os objetivos, ranking e outros dados.
- Condicionais e estruturas de repetição para gerenciar o funcionamento do jogo, além das verificações e o loop principal.

- Funções para permite uma reutilização posterior do código, além de Variáveis e outras estruturas básicas.
- Bibliotecas nativas para operações.

Ao iniciar o jogo, o usuário pode escolher entre 4 opções: carregar jogo, novo jogo, ranking e sair. Carregar jogo acessa os dados e retorna, caso exista, a partida que foi salva. Novo jogo configura o software para uma nova partida. Ranking, exibe as 10 melhores pontuações registradas até então, Por fim, sair finaliza a execução do jogo.

2. Metodologia

Durante as sessões, a equipe escolheu usar matriz como estrutura de dados para o tabuleiro do jogo. Surgiu a questão se utilizaria matrize esparsa ou tradicional, optei pela esparsa pois seria mais fácil de recuperar os dados. Além disso, a equipe debateu muito acerca do funcionamento do jogo, tentando entender a mecânica. Tivemos como escolhas de projeto: a pontuação específica dependendo da dificuldade escolhida, o usuário poder salvar somente uma partida e os jogadores não podem ter o mesmo objetivo.

Tabela 1. Esquema de pontuação adotado pela equipe.

	Fácil	Médio	Difícil
Vitória	3	4	5
Empate	1	2	3
Derrota	0	0	0

No contexto individual, optei por deixas as lógicas separadas, isto é, cada módulo trabalha de forma independente, funcionando como peças de um quebra-cabeça, sendo o módulo main responsável pela organização. Além disso, não achei necessário utilizar bibliotecas externas no projeto, fiz uso apenas das nativas sendo elas: json para ler e salvar os dados e ranking, os para limpar o terminal e random para escolher o objetivo de forma aleatória. Por fim, projetei o jogo para seguir a estrutura presente na Figura 1.

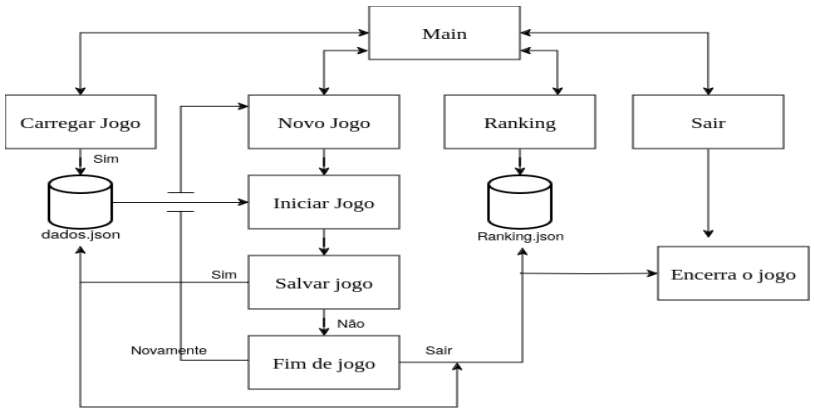

Figura 1. Estrutura do projeto.

2.1 Definição de requisitos

Para atender os requisitos básicos, foi necessário seguir algumas determinações:

1. Desenvolver um menu contendo: carregar jogo, novo jogo, ranking e sair.

2. Para atender ao 1º requisito, seria necessário desenvolver a leitura de dados de um arquivo json, para salvar a partida e ela ser recuperada posteriormente. Atrelado a isso, para atender o 2º requisito, também seria necessário ler e salvar um arquivo json, para desenvolver um ranking e o jogo ficar mais competitivo.

3. Já para atender o 3º requisito, o jogo deveria sortear um objetivo e diferenciar os jogadores, optei por utilizar cores, sendo como padrão: verde para o jogador 1 e amarelo para o jogador 2. Além disso, o jogo deveria funcionar em turnos, tendo uma vez de cada de jogar.

4. Por fim, para atender o 4º requisito, o jogo deveria considerar que os jogadores poderiam ter objetivos diferentes, além que o tamanho do tabuleiro seria proporcional a dificuldade escolhida. Ademais, os números disponíveis seriam calculados pelo tamanho do tabuleiro ao quadrado. Por fim, o jogador poderia vencer a partida caso completasse uma linha, coluna ou diagonal, principal ou invertida.

2.2 Descrição de alto nível

Para desenvolver esse jogo, foi necessário desenvolver o módulo tabuleiro, sendo ele o responsável por toda tarefa que diz respeito ao tabuleiro. Ele possui a função principal iniciar jogo, ela é responsável por fazer o jogo funcionar. Essa função recebe os dados, tanto novos quanto salvos, e realizar o desempacotamento para as variáveis que serão utilizadas. Logo após, entra no loop principal, que funciona enquanto existir números disponíveis ou nenhuma vitória tenha sido registrada. Dentro do loop, o jogo pede um número, caso ele seja -1 e o usuário tenha escolhido a jogada especial, o jogo entra no modo da jogada especial, por fim, caso o usuário tenha digitado 0, o jogo entra no modo de salvar. Logo após o jogador escolher um número válido, o jogo pede a posição que esse número será colocado. Por fim, o jogo verifica se o jogador ganhou a partida , analisando todas as possibilidades, caso não tenha vitória, o jogo passa a vez para o outro jogador, e esse loop se repete.

Logo após o desenvolvimento do módulo tabuleiro, desenvolvi o módulo ferramentas que serve para auxiliar os outros módulos. Suas funções principais são: carregar ranking, carregar jogo e novo jogo. A função carregar ranking, ler o arquivo “ranking.json”, caso tenha dados lá dentro, ela ordena os dados de forma decrescente, assim a maior pontuação fica no topo, cabe salientar que só serão exibidos os 10 melhores rankings. Já a função carregar jogo opera de modo semelhante, por meio da leitura do arquivo “dados.json”, a função verifica se existe algum jogo salvo e caso exista, ela pergunta o usuário se ele quer continuar a partida, caso queira, os dados da partida salva são enviados para a função iniciar jogo do módulo tabuleiro. Por fim, a função novo jogo, é responsável por gerar todos os dados necessários para partida: números disponíveis, pontuação, sequência, tabuleiro e outras coisas. Essa função é extremamente necessária pois ela permite que vários jogos aconteçam, e por consequência novos dados sejam gerados.

Após o desenvolvido do módulo ferramentas, desenvolvi o módulo gerenciar dados. Isso aconteceu por meio de uma refatoração nas funções carregar ranking e carregar jogo. Esse módulo basicamente possui duas funções principais: salvar dados e ler dados. Salvar dados possui como caminho padrão o arquivo “dados.json”, ela recebe os dados em forma de dicionário e com auxilio da biblioteca json, realiza a escritura no arquivo. Caso o caminho passado seja diferente do padrão, a função entra em modo de tratamento dos dados, pois ela reconhece que está trabalhando com o ranking, e este possui uma forma especial de tratamento dos dados. Após esse tratamento, a função entra no fluxo normal explicado anteriormente.

Por fim, desenvolvi o módulo main, o responsável pela execução do jogo. Esse módulo contém os menus, ele liga os outros módulos conectando suas funções principais. Basicamente, toda execução retorna ao main, ele permite que o usuário escolha as opções: carregar dados, novos dados, ranking e sair. Cada opção leva a execução a um módulo específico.

2.3 Ordem de codificação

O processo de desenvolvimento desse sistema durou cerca de 5 semanas. Na primeira semana realizei os estudos acerca da estrutura do projeto: montei o diagrama estrutural presente na figura 1 e estudei a biblioteca json.

Já na segunda semana, comecei o desenvolvimento na prática, desenvolvi o algoritmo inicial do tabuleiro, que era gerava uma matriz esparsa com uma determinada ordem que era passada como parâmetro. Além disso, desenvolvi um algoritmo para mostrar o tabuleiro e substituir os números no tabuleiro.

Na terceira semana desenvolvi o menu inicial para o jogo, atrelado a isso desenvolvi a geração dos objetivos dos jogadores, além de obter os nomes de cada jogador. Além disso, desenvolvi um algoritmo para salvar os dados do jogo, quando o usuário digitasse o número 0, o jogo entrava no modo de salvamento. Essa parte foi a mais complicada de desenvolver, pois o jogo não recupera a sequência, foi bastante complicado e cheio de erros, mas, felizmente, consegui resolver tudo. Por fim, desenvolvi a entrada de números no tabuleiro e alternar a vez do jogador.

Na quarta semana, organizei o código e refatorei algumas funções aos seus respectivos módulos. Logo após, desenvolvi a verificação da vitória, essa foi uma tarefa bastante complicada, pois inicialmente não fazia ideia de como fazer essa verificação, mas no final tudo deu certo e consegui desenvolver. Para completar desenvolvi o ranking do jogo, após as partidas adicionar a respectiva pontuação, além de ler e salvar o ranking. Neste ponto, o jogo estava praticamente desenvolvido, faltando alguns ajustes e refatorações.

Na quinta semana, tentei desenvolver um modo single-player no jogo, comecei do absoluto zero, pesquisei bastante e encontrei o algoritmo MinMax, que basicamente avalia os pontos que uma determinada jogada pode trazer. Consegui implementar metade do algoritmo, no entanto, parei por conta do tempo que estava apertado. Por fim, realizei algumas melhorias e comentei o código. Optei por remover o código do single-player do projeto, mas vou continuar desenvolvendo.

O sistema foi desenvolvido na IDE Visual Studio Code, com o interpretador Python 3.10.12, no sistema operacional Linux utilizando a distro Zorin OS. Além disso, como exposto anteriormente, o software foi desenvolvido utilizando as bibliotecas json, os e random, todas nativas da linguagem Python..

3. Resultados e Discussões

Após as cinco semanas de desenvolvimento, o jogo ficou pronto e funcionando perfeitamente, seguindo a estrutura projetada e aberto a futuras melhorias. Para jogar o tabuleiro de números, é necessário ter o Python na versão mais recente. O usuário deve executar o arquivo com nome “main”. Ele é o módulo principal do sistema, O menu inicial do jogo oferece três opções enumeradas: carregar jogo, novo jogo, ranking e sair.

A opção carregar jogo entra em um novo menu. Na verdade, ele carrega o arquivo “dados.json” e caso ele esteja vazio, uma mensagem informando é exibida na tela. Caso exista um jogo salvo, o jogo pergunta se o usuário deseja retornar a partida, e caso queira a partida é carregada do ultimo ponto. Quando o jogador finaliza a partida, automaticamente ela é apagada da memória, tendo em vista que já foi concluída.

A opção novo jogo permite uma nova geração de dados. Inicialmente, o jogo pergunta os nomes dos jogadores. Depois, um menu com as dificuldades é apresentado, e por fim, o jogo pergunta se os jogadores querem ou não ativar a jogada especial. Após essas configurações, o jogo inicia com o jogador 1. O jogo pede um número entre os disponíveis e a sua posição (as posições iniciam do 0).

Dentro do jogo é exibida uma mensagem em azul com os números disponíveis, é importante que o usuário digite um número que esteja contido no conjunto dos disponíveis, logo após, o jogo pede a posição no tabuleiro, as linhas e colunas começão do 1 e vai até o tamanho do tabuleiro. Para salvar o jogo, o usuário deve digitar 0, quando o jogo pedir para digitar um número. Caso queira utilizar a jogada especial, o jogador deve digitar -1 quando o jogo pedir para digitar um número, por fim, digitar se quer apagar linha ou coluna e depois a sua localização.

A opção ranking carrega os dados do arquivo “ranking.json”, caso o arquivo esteja vazio, uma mensagem informando é apresentada na tela. Caso existam dados, o jogo apresenta as 10 melhores posições em ordem decrescente. Cabe ressaltar, que um nome pode ser repetido diversas vezes no ranking. Por fim, a opção sair finaliza a execução do jogo.

3.1 Dados de entrada

O jogo possui alguns tratamentos de erros, mas ainda assim é importante digitar os dados corretos. Para navegar entre os menus e escolher opções é necessário digitar números inteiros, em outras palavras, cada menu tem uma opção associada a um número. Além disso, é necessário que o usuário digite um nome válido em forma de string, sem espaços em branco. Por fim, durante o jogo, o usuário deve digitar somente números, caso seja digitado algo diferente , o jogo vai fazer o devido tratamento.

3.2 Testes

Ao longo do desenvolvimento do jogo, realizei diversos testes nas funções para garantir o funcionamento adequado. Para testar convidei meu irmão para jogar comigo, iniciei um novo jogo, com os jogadores “Lucas” e “Luan”. Selecionei a dificuldade fácil e ativei o poder especial, Lucas ficou com a sequência par e Luan com a sequência descendente. A partida contou com o uso do poder especial, que funcionou de forma correta para ambos jogadores. Além disso, salvei a partida e continuei novamente, o jogo voltou do mesmo ponto e como os poderes já tinham sido usados, não habilitou novamente. Por fim, terminamos a partida que teve o jogador lucas como vencedor, a vitória refletiu no ranking de pontuação, associando cada jogador a sua respectiva pontuação.

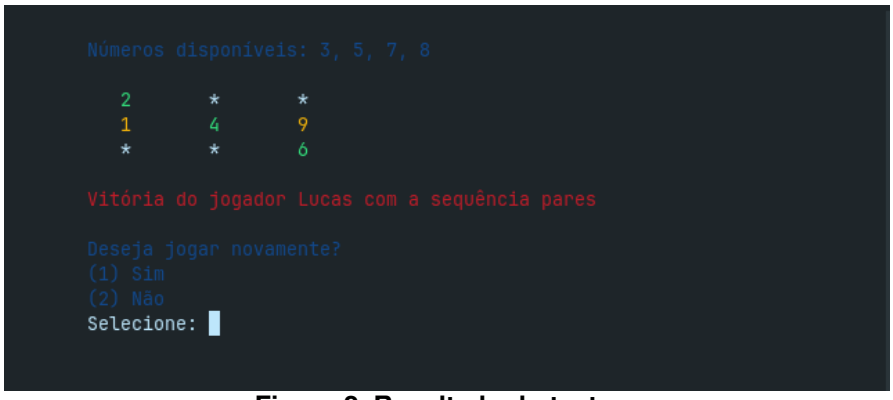

Figura 2. Resultado do teste.

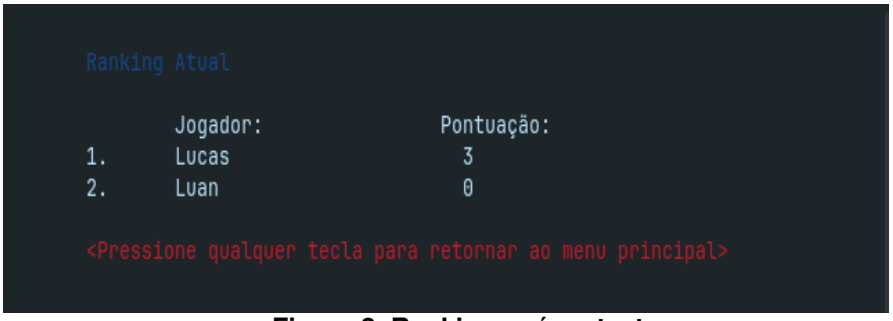

Figura 3. Ranking após o teste.

3.3 Erros

O jogo foi projetado para tratar eventuais erros de entrada. Durante o desenvolvimento não notei nenhum erro de execução, o jogo está funcionando perfeitamente. Identifiquei uma falha quando o usuário tenta colocar um nome em branco, infelizmente, o jogo permite. Além disso, caso o jogador passe uma posição já ocupada, o jogo passa a vez desse jogador, ao invés de permitir o mesmo jogar novamente. Por fim, os eventuais erros foram resolvidos e tratados de forma correta.

4. Conclusão

Em síntese, o desenvolvimento foi concluído com sucesso, foi um processo legal de desenvolver e testar. Ele alcançou todos os requisitos básicos e específicos, além de seguir a estrutura inicial que projetei. A possível melhoria é a organização de algumas linhas, que talvez ficaram muito complicadas de entender. Ademais, o jogo poderia ter

uma interface para o usuário, isso aumentaria a jogabilidade em 100% e deixaria o jogo ainda mais interessante. Em resumo, os resultados foram positivos e o jogo é bastante legal e divertido.

5. Referências

O cenário brasileiro de desenvolvimento de jogos eletrônicos. MeioeMensagem, 2022. Disponível em: <https://www.meioemensagem.com.br/proxxima/o-cenario-brasileiro-de-desenvolvimento-de-jogos-eletronicos> Acesso em 7 de junho de 2024.

WAZLAWICK, R. S. (2018). Introdução a Algoritmos e Programação com Python. Elsevier.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. (1996). Algoritmos: Lógica para Desenvolvimento de Programação. Érica,

Documentação Python. Disponível em: <https://docs.python.org/3/> Acesso em 27 de abril de 2024.