

Versão I -Anamnese

Huandy C. Silva, Letícia V. Santos, Luara V. Perilli, Luís E. Damasceno

Instituto de Matemática e Computação – Universidade Federal de Itajubá (UNIFEI)

Itajubá – MG – Brasil

huandysilva@gmail.com, leticiavsantoos@gmail.com,
luudvperilli@gmail.com, dududamasceno5@gmail.com

Resumo. Criar um jogo utilizando a lógica e as estruturas de programação. A partir dos aprendizados adquiridos e pesquisas, o jogo “Anamnese” foi criado com o fito de auxiliar na melhor compreensão dos ensinamentos adquiridos, uma vez que todas as pesquisas, práticas e escrita do código foram realizadas pelos discentes que compõem este grupo. Desse modo, o conhecimento obtido e aprofundado, a partir das explorações para execução deste trabalho, auxiliarão em outros estudos, uma vez que a compreensão desse conteúdo acarreta impactos positivos na vida dos indivíduos e do corpo social no qual estão inseridos.

Palavras-chaves: jogo, acessibilidade, cronograma, desenvolvimento.

1. INTRODUÇÃO

No decorrer da disciplina de Fundamentos de Programação (XDES01), ministrada pela Prof^ª. Dr^ª. Elisa de Cassia Silva Rodrigues, para os ingressantes do curso de Sistemas de Informação, na Universidade Federal de Itajubá (UNIFEI), no primeiro semestre do ano de 2022, foram iniciados estudos acerca dos fundamentos de programação. Bem como: lógica de programação, estrutura sequencial, condicional e de repetição, entre outros.

Nessa perspectiva, foi proposta a atividade em questão a fim de que os educandos coloquem em prática as teorias que foram lecionadas. Além disso, devido ao fato de ser uma pesquisa e programação em grupo, o trabalho também incentiva o desenvolvimento das soft skills necessárias no meio laboral, o qual seremos inseridos com o decorrer dos anos na universidade.

Neste documento, portanto, serão relatados todos os passos e implementações realizadas durante a execução do desenvolvimento do jogo, o qual é um jogo da memória numérica, intitulado *Anamnese*. Ademais, um cronograma foi criado com o intuito de facilitar a compreensão da ordem em que as atividades foram realizadas. Por fim, esse documento e o código em questão são referentes a “Versão I” do jogo da memória numérico, com o decorrer do semestre será executada uma parte final, que dará continuidade no projeto, a “Versão II”.

2. OBJETIVOS DO JOGO

Um dos principais objetivos do jogo *Anamnese* é desenvolver a capacidade de foco e concentração, uma vez que é necessário deixar de lado as distrações para que a sequência numérica seja corretamente reproduzida e o jogador possa avançar para o próximo nível. Além disso, o jogo também desenvolve a autoconfiança quando o indivíduo memoriza e acerta a sequência correta.

Ainda, o jogo também tem por objetivo estimular a memória, mantendo o jogador atento ao raciocínio. Cabe ressaltar que o aperfeiçoamento da capacidade de memorização traz inúmeros benefícios para a vida acadêmica, visto que exercita as habilidades de raciocínio e interpretação textual, permitindo a absorção de novos conhecimentos.

3. REGRAS DO JOGO

No jogo *Anamnese*, o jogador deve manter-se atento à sequência de números mostrada e, em seguida, reproduzi-la na ordem correta. Caso o jogador não acerte ou cometa um equívoco, o jogo termina e pode ser reiniciado.

É importante lembrar que a quantidade de números mostrada se torna maior - sendo incrementada de um em um - conforme o jogador acerta a sequência, aumentando a dificuldade do jogo. Dessa forma, aquele que memoriza os 30 números de maneira correta é o vencedor.

4. CRONOGRAMA

DATA	PROCESSO	DETALHAMENTO
Até 24/05/2022.	Definir qual jogo será criado.	O grupo se reuniu por meio de chamadas de vídeo no aplicativo <i>Google Meet</i> e pessoalmente, junto à professora, para que fosse definido um jogo acessível e de apenas um jogador. O jogo escolhido foi o jogo da memória numérica, o qual o grupo optou pelo nome de <i>Anamnese</i> .
Até 25/05/2022.	Criar um menu inicial para o jogo.	O grupo se reuniu durante a monitoria, para criar uma interface inicial, a qual contém as opções: Jogar, Ranking, Instruções, Créditos e Sair. O detalhamento do menu também foi criado, realizando a escrita das instruções e dos créditos.

Até o final do semestre letivo com a Versão II.	Criação do jogo.	O grupo se reuniu para analisar o que poderia ser feito, mas chegou à conclusão de que o jogo em si só poderia ser realizado após a obtenção de novos aprendizados.
---	------------------	---

5. DOCUMENTAÇÃO

5.1 Definição do Jogo

A equipe juntou-se por chamada de vídeo para afunilar as ideias e estabelecer algumas opções de jogos a fim de iniciar o projeto. Após a reunião, as sugestões apresentadas por cada integrante do grupo foram levadas à Prof^a. Dr^a. Elisa de Cássia Silva Rodrigues com o intuito de definir a melhor escolha para a criação de um jogo acessível e funcional. Após isso, o jogo da memória numérico, *Anamnese*, foi selecionado para ser construído, em decorrência da fácil acessibilização, bem como boa jogabilidade.

5.2 Definição do Editor de Texto e Bibliotecas

Com o fito de ampliar a inclusão, para todos os membros da equipe, no processo de elaboração do código fonte, foi escolhido o editor Visual Studio Code (popularmente conhecido como VS Code), afinal apresenta uma interface intuitiva e, portanto, possui simples usabilidade. Além disso, é dotado de boa compatibilidade com o software leitor de tela, o que o torna acessível a deficientes visuais.

Propôs-se para a execução do jogo a utilização da linguagem C, visando colocar em prática os conceitos de fundamentos da programação aprendidos em aulas, realizando, assim, a fixação do conteúdo. Ademais, foram implementadas as seguintes bibliotecas:

- ***stdio.h***: Possui definições de sub rotinas relacionadas a entrada e saída de dados, normalmente, sempre está presente nos códigos realizados na linguagem C;

Após a conclusão do código do jogo, este foi disponibilizado em uma IDE on-line. Foi escolhido o REPLIT, pois possibilita a execução do programa sem a necessidade de fazer download do código ou bibliotecas. Entretanto, devido a baixa acessibilidade dessa IDE, foi escolhido, também, um repositório on-line. O grupo optou pelo GitHub, devido ao acesso inclusivo e popularidade entre os desenvolvedores.

5.3 Etapa de Desenvolvimento

Primordialmente, foi adicionada a biblioteca *stdio.h*, a qual possui funções de entrada e saída de comandos. Após essa implementação criou-se a função principal, *int main*, que contém um *return 0*, dentro dessa função que será desenvolvido todo o código.

5.3.1 Criação do Menu

Dentro da função supracitada, inicialmente, duas variáveis foram criadas: *escolha* do tipo caractere, a qual é iniciada sem nenhum caractere correspondente, e *sair* do tipo inteiro, a qual é inicializada com o valor 1, ambas foram criadas para armazenar as escolhas do jogador. Ademais, de acordo com as instruções recebidas para o trabalho prático, o menu deveria conter as opções para que o jogador pudesse iniciar o jogo e sair deste. Por conseguinte, o grupo também optou por adicionar instruções, créditos e ranking.

Para que o menu principal fosse construído com todas as opções necessárias, foi utilizada a estrutura de repetição *do while*, no qual as instruções contidas dentro da chave são executadas em looping até que o usuário opte por sair do jogo. A primeira função escrita dentro do laço foi a *system("cls")* que é utilizada quando o compilador está relacionado com o sistema operacional Windows. Quando o código foi adicionado ao REPLIT houve a mudança para *system("clear")*, pois essa IDE utiliza compiladores relacionados ao sistema operacional Linux, ou seja, a função é diferente de acordo com o sistema operacional utilizado. Outrossim, a função *system("cls")* é inicializada sempre que o loop se repete, com o fito de limpar o terminal, tornando a interface mais agradável ao usuário.

Nessa perspectiva, após a utilização da função para limpeza do terminal, foi utilizada a função *printf* com o objetivo de exibir o nome do jogo, *Anamnese*. Seguidamente, serão exibidas, por meio de funções *printf*, as opções supracitadas para o jogador, onde a letra "J" representa Jogar, a letra "R" representa o Ranking, a letra "I" representa as Instruções, a letra "C" representa os Créditos e, por fim, a letra "S" representa Sair. Posteriormente, utilizando a função *scanf*, a opção do usuário é lida e armazenada na variável *escolha*.

Ainda, dentro do laço de repetição *do while*, foi adicionada a estrutura *switch case*, na qual o conteúdo da variável *escolha* é comparado aos caracteres estabelecidos, executando um comando caso haja equivalência entre os valores. Também, a utilização do laço *switch case* visa reduzir a complexidade de vários *if else* encadeados.

Cada um dos *case*, incluindo o *default*, possuem uma sequência de código a ser executadas, a saber:

- **Caso 1 - Jogar [J]:** Essa opção realiza a execução do jogo, o qual ainda não foi implementado, portanto, ao selecionar [J], entra-se em um *do while*, que será executado enquanto a variável *sair* for diferente de 1, no qual a função *system("cls")* será executada a fim de limpar o terminal. Sequencialmente, um *printf* exibirá a mensagem "*Jogo a ser implementado em versões futuras*". Em seguida, um novo *printf*, contendo a mensagem "*Digite [1] para voltar ao menu*", é utilizado com intuito de dar ao usuário a opção de voltar ao menu, - essa opção será lida e armazenada na variável *sair*. Caso o jogador digite 1, o loop se encerra.
- **Caso 2 - Ranking [R]:** Essa opção exibe o ranking do jogo, o qual ainda não foi implementado, portanto, ao optar por [R], um *do while* é inicializado, o qual será executado, também, enquanto a variável *sair* for diferente de 1. Dentro do laço de repetição, a função *system("cls")* será executada a fim de limpar o terminal. Sequencialmente, um *printf* exibirá a mensagem "*Ranking a ser implementado em versões futuras*". Em seguida, um novo *printf*, contendo a mensagem "*Digite [1] para voltar ao menu:*", é utilizado com o

mesmo intuito da opção anterior - essa opção será lida e armazenada na variável *sair*. Caso o jogador digite 1, o loop se encerra.

- **Caso 3 - Instruções [I]:** Ao optar por [I], um *do while* será inicializado, sendo a função *system("cls")* a primeira função a ser executada. Após, serão apresentadas as instruções do jogo, através de estruturas *printf*, com a seguinte mensagem:

“

INSTRUÇÕES

1. O jogador deve manter-se atento à sequência de números mostrada;
2. O jogador deve reproduzir a sequência de números na ordem correta;
3. O jogo termina caso o jogador cometa um equívoco;
4. A quantidade de números aumenta conforme o jogador acerta a sequência;
5. O jogador que memorizar a maior quantidade de números é o vencedor.”

Seguidamente, também será exibida a opção de retornar ao menu, e esta funcionará assim como nas demais opções, utilizando a leitura da variável *sair*.

- **Caso 4 - Créditos [C]:** Caso a opção [C] seja a escolhida, assim como nas demais, um laço de repetição, *do while*, será iniciado, e a primeira função a ser executada será a *system("cls")*. Posteriormente, serão apresentados os créditos do jogo, através de estruturas *printf*, com as seguintes mensagens:

“

CRÉDITOS

Este jogo foi criado em 2022 por: Huandy Calini de Camargo Silva, Letícia Vitória dos Santos, Luara do Val Perilli e Luís Eduardo Damasceno, alunos da disciplina de Fundamentos de Programação e deve ser utilizado como ferramenta educacional e de entretenimento.”. Em sequência, os mesmos passos serão executados visando finalizar o loop e voltar ao menu.

- **Caso 5 - Sair [S]:** Essa opção é a única que finaliza o loop. Nesta, primeiramente, será executada a função *system("cls")* para limpar o terminal. Logo após, será exibida a mensagem “Fechando jogo...” por meio da função *printf*. Desse modo, quando o usuário fizer essa seleção, o laço de repetição do menu se encerra e, consequentemente, o jogo também.
- **Caso 6 - Default:** Caso seja selecionado uma tecla fora das opções válidas, um *do while* inicia, contendo, primeiramente, a função *system("cls")* e, seguida, através de um *printf*, “Opção inválida”. Posteriormente, será dada a opção de voltar ao menu, também com a exibição da mensagem “Digite [1] para voltar ao menu:”, por meio de um *printf*. Com a leitura da variável *sair*, essa sendo igual a 1, o loop se encerra voltando ao menu principal.

Nota-se, ainda, que foi adicionada aos casos 1, 2, 3, 4 e 6 a opção *sair*, visando proporcionar um maior tempo para o leitor entender o que há dentro de cada um dos casos. A título de exemplificação, tem-se as instruções, a qual contém as regras do jogo, que são fundamentais para a compreensão de como a *Anamnese* funcionará, por tanto necessitam ser analisadas cuidadosamente.

6. DIFICULDADES ENCONTRADAS

Durante a criação do jogo *Anamnese*, poucos obstáculos foram encontrados. Entre eles, é possível citar que, ao adicionarmos as instruções e créditos, foi necessária a utilização de várias estruturas *printf*, para que quando o código fosse lido e avaliado, não houvesse a necessidade de arrastar a tela para o lado. Por isso, para que não houvessem grandes mensagens em um único *printf*, a mensagem foi dividida em *printf* menores.

Também cabe mencionar que, ao executar o menu, foi identificado um erro no qual “opção inválida” sempre era exibida após o jogador realizar a escolha de um dos casos. Isso se deu pela falta de um espaço na leitura da variável do tipo caractere, *escolha*, ou seja, *scanf("%c", &escolha)*. Por conseguinte, o correto seria *scanf(" %c", &escolha)*. Isso ocorre devido ao *buffer* de teclado. Em outras palavras, ao digitar uma opção - [J], por exemplo - e apertar a tecla enter, o *buffer* de teclado armazena temporariamente aquilo que foi digitado - neste caso, a tecla enter como (\n) - como o próximo caractere atribuído ao menu. Assim, como o enter (\n) não estava entre as demais opções, “opção inválida” era exibido antes de voltar ao menu e dar ao usuário a oportunidade de escolher uma nova opção. Dessa maneira, quando adicionado um espaço antes de %c, tornando-se “ %c”, o espaço, enter ou tab, por exemplo, que pode estar no *buffer* é dispensado, uma vez que o espaço dado antes do %c realiza uma espécie de limpeza no *buffer*, possibilitando, assim, a escolha do usuário. Ademais, não foram encontradas outras dificuldades na criação do jogo de modo geral.

7. CONSIDERAÇÕES FINAIS

Com a realização desse trabalho, pode-se aprofundar os conhecimentos obtidos por intermédio das aulas ministradas pela Prof^a. Dr^a. Elisa de Cássia Silva Rodrigues. Isso se deve pela aplicação prática dos conceitos fundamentais de Fundamentos de Programação, como estrutura sequencial, condicional e de repetição, além da utilização de variáveis e funções básicas como *printf* e *scanf*. Outrossim, implementamos a biblioteca, normalmente, presente em programas escritos na linguagem C, a *stdio.h*.

Portanto, conclui-se que a execução do jogo *Anamnese* agregou novos aprendizados para os membros da equipe, visto que, além de desenvolvermos um jogo com a linguagem C, o que proporcionou a aplicação dos fundamentos aprendidos, também efetuamos a escrita de toda a documentação, a qual é de extrema importância quando se está desenvolvendo códigos.

8. LINK PARA ACESSAR O CÓDIGO

IDE Replit: <https://replit.com/@leticiasants/Versao-I#main.c>

Repositório GitHub: https://github.com/leticiasants/Versao_1_Jogo_Anamnese