



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**MISAEAL AUGUSTO SILVA DA COSTA**

**THE CODE: UTILIZANDO LÓGICA DE PROGRAMAÇÃO PARA DESENVOLVER UM JOGO DE  
RACIOCÍNIO LÓGICO**

**CAMPINA GRANDE - PB**

**2021**

**MISAEEL AUGUSTO SILVA DA COSTA**

**THE CODE: UTILIZANDO LÓGICA DE PROGRAMAÇÃO PARA DESENVOLVER UM  
JOGO DE RACIOCÍNIO LÓGICO**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**Orientador: Professor Dr. Marcelo Alves de Barros**

**CAMPINA GRANDE - PB**

**2021**



C837t Costa, Misael Augusto Silva da.  
The Code: utilizando lógica de programação para desenvolver um jogo de raciocínio lógico. / Misael Augusto Silva da Costa. - 2021.

12 f.

Orientador: Prof. Dr. Marcelo Alves de Barros.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Programação orientada a objetos. 2. Raciocínio lógico. 3. Lógica de programação. 4. The code. I. Barros, Marcelo Alves de. II. Título.

CDU:004(045)

**Elaboração da Ficha Catalográfica:**

Johnny Rodrigues Barbosa  
Bibliotecário-Documentalista  
CRB-15/626

**MISAEEL AUGUSTO SILVA DA COSTA**

**THE CODE: UTILIZANDO LÓGICA DE PROGRAMAÇÃO PARA DESENVOLVER UM  
JOGO DE RACIOCÍNIO LÓGICO**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Marcelo Alves de Barros  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Carlos Eduardo Santos Pires  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 20 de Outubro de 2021.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

At the beginning of the journey of learning to program, the first step presented is to practice programming logic. That is, the ability to think and develop algorithms capable of solving computational problems. Learning to program means developing this kind of thinking, which can be used to create solutions to real-world problems that can become software products. In this work, the objective is to develop a software product using programming logic and object-oriented programming, as main concepts. The result of this work is a 2D game called The code, in which the player must use logical reasoning to guess a combination of numbers to pass the level. It is expected to show with the development of this project, that it is possible to develop a software product from the basic knowledge acquired in the first year of the Computer Science course.

# The code: utilizando lógica de programação para desenvolver um jogo de raciocínio lógico

Trabalho de Conclusão de Curso

Misael Augusto Silva da Costa  
[misael.costa@ccc.ufcg.edu.br](mailto:misael.costa@ccc.ufcg.edu.br)  
Campina Grande, Paraíba, Brasil

Marcelo Alves de Barros  
[mbarros@computacao.ufcg.edu.br](mailto:mbarros@computacao.ufcg.edu.br)  
Campina Grande, Paraíba, Brasil

## RESUMO

No início da jornada de aprender a programar, o primeiro passo apresentado é o de praticar a lógica de programação. Isto é, a capacidade de pensar e desenvolver algoritmos capazes de resolver problemas computacionais [1]. Aprender a programar, significa desenvolver esse tipo de raciocínio, o qual pode ser utilizado para criar soluções para problemas do mundo real, que podem se tornar produtos de software.

Neste trabalho, o objetivo é desenvolver um produto de software utilizando lógica de programação e programação orientada a objetos, como principais conceitos. O resultado deste trabalho é um jogo 2D chamado The code, no qual o jogador deve utilizar o raciocínio lógico para adivinhar uma combinação de números para passar de nível.

Espera-se mostrar com o desenvolvimento desse projeto, que é possível desenvolver um produto de software a partir dos conhecimentos básicos adquiridos no primeiro ano do curso de Ciência da Computação.

## PALAVRAS-CHAVE

The code. Lógica de Programação. Programação Orientada a Objetos. Raciocínio Lógico.

## REPOSITÓRIO

<https://github.com/MisaelAugusto/the-code>

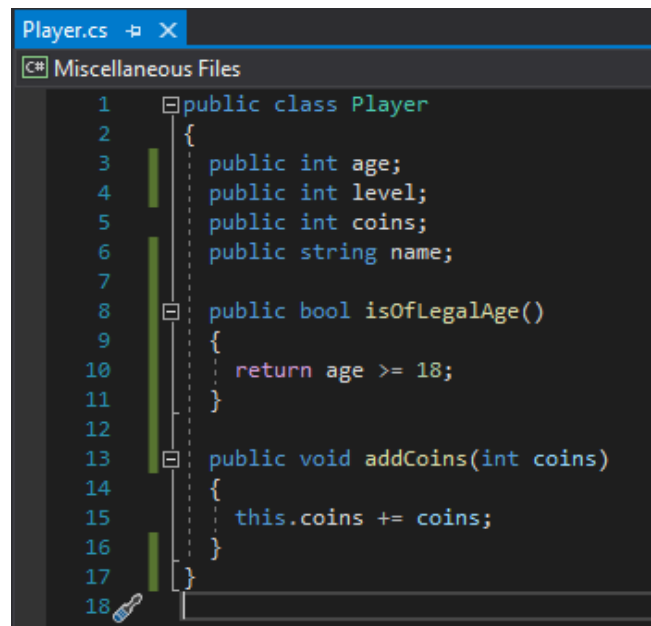
## 1. INTRODUÇÃO

Desde o início da jornada na graduação do curso de Ciência da Computação, são apresentadas as disciplinas práticas de programação. São nelas onde aprende-se a programar e adquirir o raciocínio para

resolver os problemas computacionais, mais conhecido como lógica de programação.

A partir da lógica de programação, o novo desafio é aprender a programação orientada a objetos. É nela onde é possível criar estruturas chamadas de Classes. Uma classe define um conjunto de objetos com características e comportamentos similares [2]. Por exemplo, ao definir uma classe chamada *Player*, é possível definir as características que qualquer objeto desta classe irá possuir. Neste caso, um objeto dessa classe poderia ter como características: name, age, level e coins. Para definir os comportamentos, as classes possuem métodos, os quais podem manipular as próprias informações dos objetos da classe.

No exemplo de código abaixo, tem-se a classe *Player*:



```
Player.cs [X]
Miscellaneous Files
1 public class Player
2 {
3     public int age;
4     public int level;
5     public int coins;
6     public string name;
7
8     public bool isOfLegalAge()
9     {
10        return age >= 18;
11    }
12
13    public void addCoins(int coins)
14    {
15        this.coins += coins;
16    }
17 }
18
```

Figura 1: código de exemplo de uma classe na linguagem de programação C#

Neste exemplo, a classe possui os atributos já mencionados e dois métodos: o *isOfLegalAge* retorna *true*, caso o jogador seja maior de idade aqui no Brasil, do contrário, retorna *false*; o *addCoins* recebe uma quantidade de moedas como parâmetro e adiciona-as às moedas do jogador. De forma simples tem-se uma classe que ao criarmos objetos a partir dela, é possível manipular esses e outros objetos de diferentes classes e desenvolver um jogo simples e divertido.

Este trabalho tem como objetivo apresentar o jogo The code, resultado de um processo de desenvolvimento utilizando essencialmente conhecimentos de lógica de programação e programação orientada a objetos. Utilizando esses conhecimentos, é possível aplicá-los para construir um produto em nível profissional que pode ser comercializado nas lojas de jogos e aplicativos.

## 2. PROJETO

Neste trabalho, o projeto objeto de estudo é o The code, um jogo 2D de raciocínio lógico desenvolvido utilizando essencialmente os conhecimentos de lógica de programação e programação orientada a objetos. O objetivo é mostrar com um projeto simples a importância de, como aluno da graduação, aplicar os conhecimentos aprendidos nas disciplinas para desenvolver um produto de software.

### 2.1 DESCRIÇÃO

No The code, o objetivo principal do jogador é adivinhar qual a combinação numérica (código) de cada nível. Para isso, o jogador pode fazer uma quantidade pré-definida de tentativas por nível. A cada tentativa o jogo irá indicar, através das bolinhas amarelas, quais dos números digitados pelo jogador fazem parte do código do nível, mas não estão na posição correta. E, através das bolinhas verdes, indicar quais fazem parte do código e também estão na posição correta.

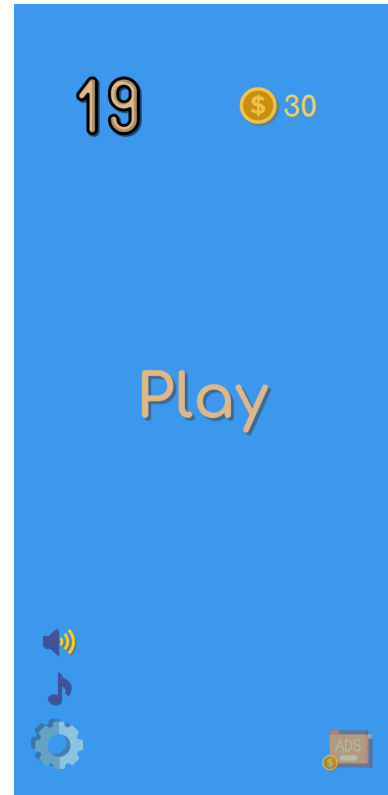


Figura 2: tela inicial do The code

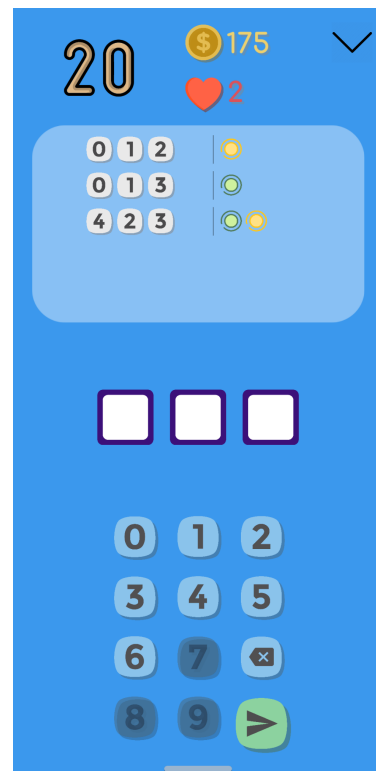
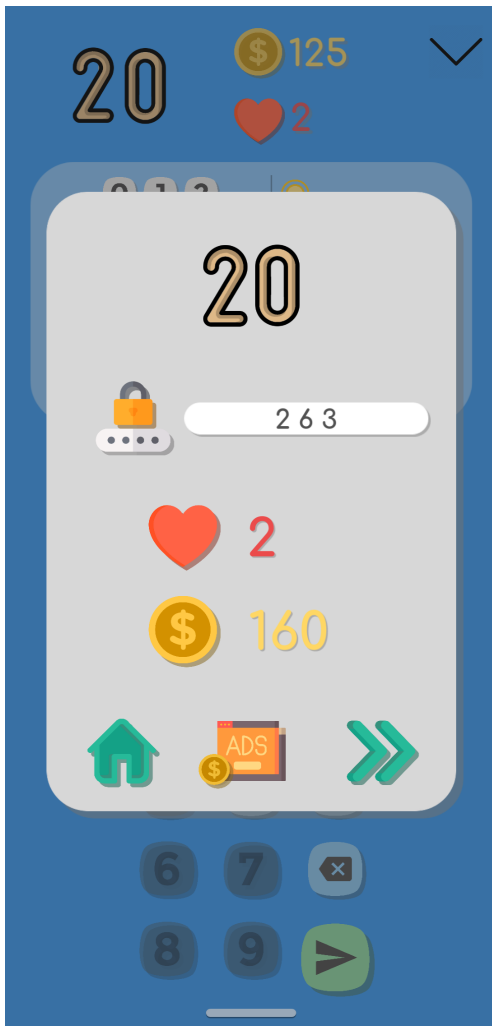


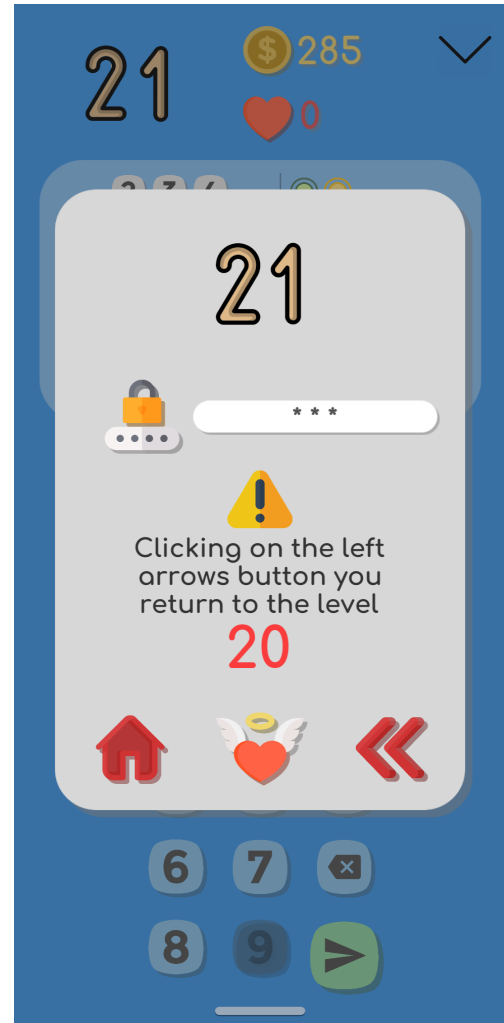
Figura 3: tela de jogo com três tentativas realizadas

Caso o jogador acerte a combinação antes de gastar todas as tentativas do nível, o jogo mostrará a janela abaixo, na qual o jogador pode: voltar para a tela inicial; assistir a um anúncio para dobrar a quantidade de moedas recebidas por concluir o nível ou avançar para o próximo nível.



**Figura 4:** tela de jogo com a janela de vitória

No entanto, caso o jogador utilize todas as tentativas do nível, mas não consiga descobrir o código, o jogo mostrará a janela abaixo, na qual o jogador pode: voltar para a tela inicial; assistir a um anúncio para tentar novamente o mesmo nível ou retroceder uma quantidade pré-definida de níveis.



**Figura 5:** tela de jogo com a janela de derrota

## 2.2 FUNCIONALIDADES

- Na tela inicial, o jogador pode assistir a um anúncio para ganhar entre 50 e 150 moedas por anúncio. Este valor é gerado aleatoriamente toda vez que o anúncio estiver disponível;
- O jogador pode ligar e desligar a música e os efeitos sonoros do jogo;
- O jogador pode fazer palpites em cada nível clicando nos botões de número e depois clicando no botão verde com a seta;
- O jogador pode comprar dicas, ou seja, um número do código é revelado, gastando uma quantidade pré-definida de moedas;



- O jogador pode comprar tentativas extras, até um máximo de 5, gastando uma quantidade pré-definida de moedas;
- O jogador pode assistir a um anúncio para dobrar a quantidade de moedas obtidas pelo nível concluído;
- O jogador pode assistir a um anúncio para tentar novamente o mesmo nível sem ter que retroceder para um nível anterior;
- O jogo pode mostrar todas as tentativas que o jogador fez para o nível no qual ele está;
- O jogo pode salvar e recuperar o progresso do jogador localmente no dispositivo no qual ele foi instalado.

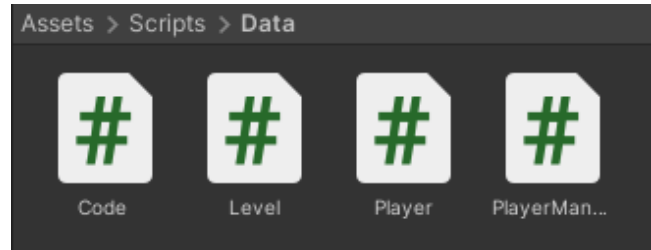
## 2.3 ARQUITETURA

O The code é um jogo simples e o seu desenvolvimento tem como principal diretriz utilizar as abordagens mais simples para implementá-lo. Portanto, todos os dados do jogador são salvos localmente em um arquivo binário, removendo a necessidade de um banco de dados.

É por esse motivo que o jogo possui uma arquitetura DDD, ou seja, Desenvolvimento Dirigido por Domínios. Neste projeto, todos os scripts são divididos pela sua funcionalidade e responsabilidade dentro do jogo, por exemplo: o *UIManager.cs* trata dos elementos de interface, já o *ADSUnity.cs* lida com a interação entre o jogador e os anúncios.

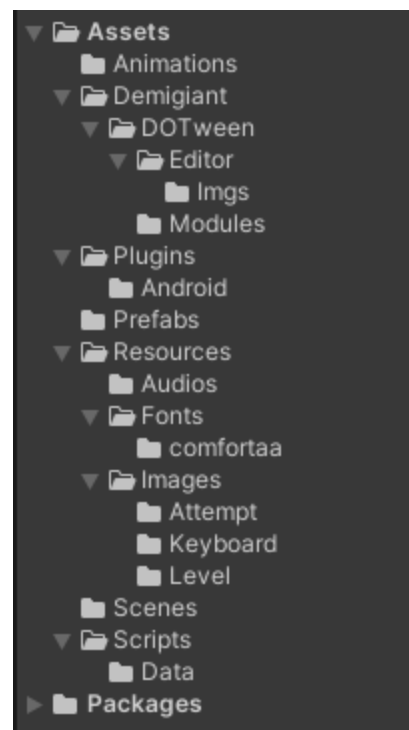


**Figura 6:** todos os scripts do The code



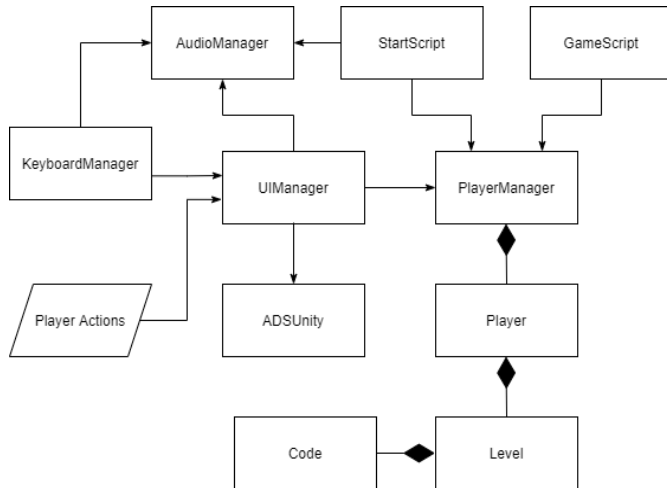
**Figura 7:** classes que armazenam os dados do jogo

Além disso, a estrutura de pastas do projeto também é dividida por responsabilidades e áreas do jogo.



**Figura 8:** pastas e subpastas do diretório do projeto

O diagrama abaixo mostra a comunicação entre os componentes do jogo. Nele observamos um baixo nível de acoplamento pois as entidades se comunicam de forma simples, sem dependência mútua (A depende de B e B depende de A) ou circular (A depende de B, B depende de C e C depende de A).



**Figura 9:** diagrama de componentes do jogo

Legenda: As setas indicam que o componente que toca a linha, acessa o componente que toca a cabeça de seta; as linhas com losango indicam composição, o componente de que toca o losango é composto pelo componente que toca a linha.

## 2.4 TECNOLOGIAS

Referente às tecnologias, foi utilizada a *game engine* [3] Unity [4] para desenvolver o jogo e a linguagem de programação C# para implementar os scripts. Para a prototipação das telas e criação dos *assets* [5] do jogo utilizou-se a versão para navegador da ferramenta Figma [6]. Para a música e os efeitos sonoros utilizou-se um site que fornece áudios livres de direitos autorais, o Freesound [7], e para criar e/ou editar outros sons foi utilizado o software Audacity [8]. Por fim, o código fonte do jogo está armazenado em um repositório público do site Github [9], aliado a ele utilizou-se a ferramenta Git [10] para lidar com o versionamento do código.

## 3. METODOLOGIA

O primeiro passo no desenvolvimento do projeto foi escolher quais seriam as tecnologias que seriam utilizadas. Essa decisão foi bem simples e rápida, devido ao conhecimento prévio na *game engine* Unity e na linguagem de programação C#.

Para atender ao propósito deste trabalho, levantou-se um número mínimo de requisitos funcionais. Assim, seria possível desenvolver um

projeto completo e comercializável, utilizando os conceitos mais básicos de programação.

Todas as decisões do trabalho se basearam em manter as soluções simples e de rápida implementação. Um exemplo é a lógica para salvar os dados do jogador, que utiliza um arquivo binário salvo localmente, ao invés de um banco de dados ou uma integração com a Play Games.

Ademais, o processo de desenvolvimento foi dividido em iterações de implementação seguida de testes no próprio dispositivo móvel. Assim, a necessidade de testes unitários diminuiu no primeiro momento. Além disso, acelerou o processo de ajuste ao formato da tela do dispositivo, deixando o jogo sempre responsivo a cada iteração de desenvolvimento.

Por fim, na etapa de avaliação do produto final, utilizou-se um formulário [11] com questões específicas sobre o jogo. Abaixo segue as principais perguntas:

1. *Qual sua nota para o visual do jogo (cores, ícones, tipografia e animações) ?*
2. *Qual sua nota para a música e os efeitos sonoros do jogo ?*
3. *Com relação a dificuldade do jogo, você o considera: (Fácil; Normal; Difícil)*
4. *Com relação a progressão da dificuldade do jogo, você o considera: (Irregular; Fluida; Brusca)*
5. *Com relação ao desempenho do jogo, durante as sessões em que esteve jogando, com que frequência você experienciou algum travamento ou lentidão ? (Nenhuma vez; Algumas vezes; Todas as vezes que jogou)*

Um grupo de 6 pessoas se voluntariou a jogar e testar todos os recursos do jogo. Após isso, foi solicitado a cada participante que respondesse o formulário para avaliarem como foi a experiência de entretenimento ao jogar o The code. O objetivo é validar se o produto de software desenvolvido neste projeto atende aos requisitos mínimos de qualidade de um jogo.

## 4. RESULTADO

Como resultado da etapa de avaliação do The code, conclui-se que nessa primeira fase de testes os aspectos audiovisuais do jogo agradaram todos os jogadores.

Qual sua nota para o visual do jogo (cores, ícones, tipografia e animações) ?  
6 respostas

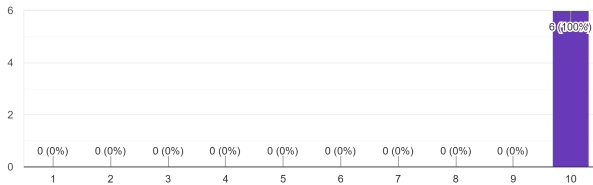


Figura 10: gráfico das respostas para a pergunta 1

Qual sua nota para a música e os efeitos sonoros do jogo ?  
6 respostas

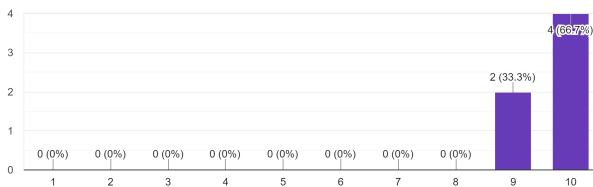


Figura 11: gráfico das respostas para a pergunta 2

Quanto à dificuldade do jogo, pode-se concluir que a dificuldade manteve-se normal para a maioria dos jogadores, mas há ainda a necessidade de melhoria para os que acharam o jogo difícil, tenham uma experiência mais facilitada.

Com relação a dificuldade do jogo, você o considera:  
6 respostas

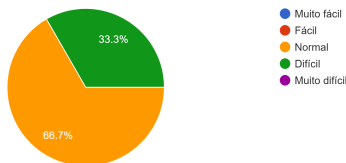


Figura 12: gráfico das respostas para a pergunta 3

Relacionado à questão acima, um ponto de melhoria levantado pela maioria foi a necessidade de um tutorial, que, devido ao tempo, não coube no escopo deste projeto. Um nível inicial no qual o

jogador é instruído sobre como realizar cada ação do jogo e o que ele deve fazer para passar de nível.

A implementação do tutorial, como também as sugestões de melhorias e novas funcionalidades dos usuários, podem servir de base para um trabalho futuro.

Na questão 4, todos os jogadores que avaliaram o The code, consideraram a experiência de jogo fluida, ou seja, a dificuldade dos níveis aumenta de forma perceptível, porém não há uma mudança brusca entre níveis consecutivos.

Com relação a progressão da dificuldade do jogo, você a considera:  
6 respostas

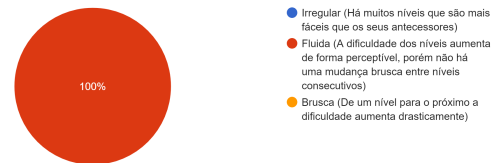


Figura 13: gráfico das respostas para a pergunta 4

Por fim, todos jogadores que testaram o jogo, confirmaram que não tiveram nenhuma experiência de travamento ou lentidão em seus dispositivos.

Com relação à performance do jogo, durante as sessões em que esteve jogando, com que frequência você experienciou algum travamento ou lentidão ?  
6 respostas

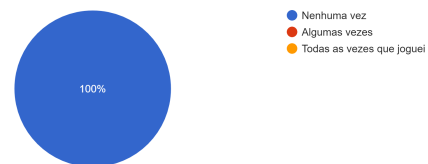


Figura 14: gráfico das respostas para a pergunta 5

## 5. CONCLUSÕES

Com o desenvolvimento desse projeto, ficou claro que é possível, com o tempo necessário, desenvolver qualquer produto de software sem necessariamente investir tempo e esforço estudando todas as áreas da Ciência da Computação.

É necessário buscar o conhecimento mínimo necessário para iniciar. Neste projeto é a lógica de programação e programação orientada a objetos. Os

demais conhecimentos podem ser adquiridos no processo de desenvolvimento.

Portanto, se uma ideia surge, e desenvolvê-la traz ânimo e sentido para o seu dia a dia, não há motivos para não fazer. O importante é fazer o melhor que você consegue com as condições que tem hoje, até um dia ter condições melhores para fazer melhor ainda.

Durante a graduação em Ciência da Computação, é muito importante, principalmente para os que desejam se tornar desenvolvedores de software, aproveitar as ideias que surgem e utilizar o conhecimento adquirido nas disciplinas para criar soluções em formato de produto de software. O desenvolvimento do The code mostra que é possível criar, a partir de uma ideia e de conhecimentos de disciplinas iniciais do curso, um projeto de entretenimento para clientes reais, neste caso, jogadores casuais da Play Store.

## 5.1 DESAFIOS E LIMITAÇÕES

No desenvolvimento do The code, houveram desafios que pausaram o processo de implementação por várias semanas. O notebook utilizado para programar o jogo sofria para renderizar as cenas do jogo após cada alteração nos scripts, devido à baixa quantidade de memória ram e por possuir um HD ao invés de um SSD, o qual é bem mais rápido para manipulação de arquivos e execução de programas mais pesados.

Além de problemas técnicos, também foi necessário lidar com as ideias se multiplicando e tornando o projeto de um jogo 2D simples de raciocínio lógico, em um projeto de um jogo AAA [4] (categoria de jogos desenvolvidos pelas indústrias de jogos já consolidadas no mercado, que normalmente possuem milhões de dólares de orçamento).

Quanto às limitações, com os recursos que estavam disponíveis, não foi possível desenvolver o jogo para ser compatível com dispositivos que possuem o sistema operacional iOS. Para isso, seria necessário um MacBook e um iPhone. Portanto, o jogo foi desenvolvido e está disponível apenas para dispositivos móveis Android.

## 5.2 TRABALHOS FUTUROS

Esta é uma versão bem breve do que realmente poderia ser mostrado, pois as possibilidades de produtos de software que podem ser desenvolvidos desde o primeiro ano de curso de Ciência da Computação, são ilimitadas.

Há a possibilidade de evoluir com o jogo desenvolvido neste trabalho, aprimorando os elementos audiovisuais e adicionando novas funcionalidades, tais como:

- Nível inicial de tutorial com instruções básicas para jogar o jogo;
- Sistema de Ranking baseado na quantidade de níveis que o jogador avançou no jogo;
- Sistema de conquistas e títulos, para os jogadores buscarem concluir marcas pré-estabelecidas de níveis, por exemplo;
- Adicionar possibilidade de customização dos elementos do jogo utilizando moedas;
- Perfil para cada jogador com nome e um ícone baseado nos títulos e conquistas;
- Novo modo de jogo no qual o tempo é limitado;
- Novo modo de jogo no qual o jogador deve descobrir 3 combinações diferentes, uma em seguida da outra, antes que um outro jogador o faça.

## 6. AGRADECIMENTOS

Sou muito grato primeiramente a Deus, por ter me dado o discernimento, coragem e resiliência para concluir a jornada da graduação. Sou grato também à minha mãe Luciana, meu pai de coração Fabiano e minha avó Maria de Fátima, pessoas que foram importantíssimas, sem elas, nada disso seria possível. Agradeço a confiança e apoio de Maria Helena, que me ajudou também na minha missão de ingressar na universidade. Deixo minha gratidão também para minha fiel companheira Ana Vitória e meu amigo Diego Alex que se mostraram sempre dispostos a me ajudar nos pequenos problemas e obstáculos que surgiram no caminho da minha formação. Agradeço ao professor Marcelo Alves que me orientou na escrita e no desenvolvimento deste trabalho. Também deixo meus agradecimentos aos demais que fizeram parte de alguma forma deste

processo. Serei eternamente grato a todos, muito obrigado !

## 7. REFERÊNCIAS

- [1] Lógica de Programação: O que é e como aprender ?  
<https://blog.betrybe.com/tecnologia/logica-de-programacao/>
- [2] Classes (OOP) | Brilliant Math & Science Wiki  
<https://brilliant.org/wiki/classes-oop/>
- [3] What is a Game Engine ?  
<https://fullscale.io/blog/what-is-game-engine/>
- [4] Unity Real-Time Development Platform | 3D, 2D & AR Engine  
<https://unity.com/>
- [5] Unity - Game Development Terms  
<https://unity.com/pt/how-to/beginner/game-development-terms>
- [6] Figma: the collaborative interface design tool.  
<https://www.figma.com/>
- [7] Freesound - Freesound  
<https://freesound.org/>
- [8] Audacity ® | Free, open source, cross-platform audio software for multi-track recording and editing.  
<https://www.audacityteam.org/>
- [9] GitHub  
<https://github.com/>
- [10] Git  
<https://git-scm.com/>
- [11] Feedback do jogo 'The code' - Google Forms  
<https://forms.gle/vRdTahXUSuBZoqLd6>
- [12] Feedback do jogo 'The code' (Responses) - Google Sheets  
<https://docs.google.com/spreadsheets/d/1UfGbUuF-CF-SRtvicLuJ68shFpgL00S5d0w-y92OHHhg/edit?usp=sharing>