

Desenvolvimento de um jogo para ensino de lógica de programação

Thiago de Carvalho Regis; José Alexandre Ducatti

e-mail:

thiago.regis@fatec.sp.gov.br; jose.ducatti@fatec.sp.gov.br

Resumo: Este trabalho apresenta o desenvolvimento de um jogo educativo dedicado ao ensino de lógica de programação, utilizando a Unity e a linguagem C#. O principal objetivo é proporcionar uma experiência interativa e envolvente para facilitar a compreensão de conceitos fundamentais de algoritmos. A metodologia adotada abrange a elaboração do Game Design Document (GDD), prototipagem, desenvolvimento na Unity, testes e compilação para desktop. Os resultados obtidos destacam a criação de desafios que exploram estruturas lógicas e algoritmos. Conforme o jogador avança, a dificuldade do jogo aumenta, incorporando conceitos de desafios anteriores e introduzindo novos. O projeto representa uma ferramenta eficaz para o ensino de lógica de programação, oferecendo uma abordagem inovadora e adaptativa.

Palavras-chave: Jogo educativo. Lógica de programação. Unity. Ensino interativo.

Abstract: This work presents the development of an educational game dedicated to teaching programming logic, using Unity and the C# language. The main objective is to provide an interactive and engaging experience to facilitate the understanding of fundamental concepts of algorithms. The adopted methodology includes the elaboration of the Game Design Document (GDD), prototyping, development in Unity, testing, and compilation for desktop. The results highlight the creation of challenges that explore logical structures and algorithms. As the player progresses, the game's difficulty increases, incorporating concepts from previous challenges and introducing new ones. The project represents an effective tool for teaching programming logic, offering an innovative and adaptive approach.

Keywords: Educational game. Programming logic. Unity. Interactive teaching.

1 Introdução

À medida que avançamos em um mundo cada vez mais tecnológico, torna-se indispensável adquirir habilidades em linguagens de programação para uma comunicação efetiva com os computadores. Nos dias de hoje, estamos constantemente rodeados por dispositivos digitais, desde smartphones compactos até computadores poderosos. No entanto, para interagir com essas máquinas e desfrutar plenamente de suas capacidades, é necessário utilizar software desenvolvido através de linguagens de programação.

Assim como utilizamos a linguagem falada para nos comunicarmos uns com os outros, as linguagens de programação nos permitem estabelecer uma comunicação clara e precisa com os computadores. Enquanto a linguagem humana emprega palavras, frases e gramática, a programação faz uso de uma sintaxe específica, estruturas lógicas e algoritmos

bem definidos, permitindo instruir essas máquinas a realizar determinadas ações e tarefas que teríamos dificuldade de realizar.

Aprender uma linguagem de programação é muito mais do que apenas adquirir conhecimentos técnicos. É adentrar em um mundo de possibilidades, onde somos capacitados a transformar ideias em realidade, desenvolver aplicativos inovadores e solucionar problemas complexos.

Devido a isso esse trabalho tem como objetivo desenvolver um jogo utilizando a Unity, uma plataforma de desenvolvimento de jogos, e a linguagem C#. O jogo será projetado para ensinar conceitos de algoritmos e lógica de programação de forma interativa e envolvente, incentivando-os a explorar e aprofundar seus conhecimentos nesse campo.

2 Justificativa

Conforme a pesquisa do Brasscon estima que as empresas de tecnologia demandem 797 mil talentos em tecnologia de 2021 a 2025 de desenvolvimento e segunda projeções da Brasscon e o número de formandos da área de TI, terá um déficit anual de 106 mil talentos, e 530 mil em cinco anos, sendo necessário cada vez mais, ter conhecimentos nessa área, devido que cada vez mais o mercado de trabalho busca a digitalização no ambiente do trabalho, com ecommerces, aplicativos moveis, jogos e outras formas. Para criação desses componentes é necessário utilizar lógica de programação, para achar a solução do problema e algoritmos para colocar essa solução em prática. Além que com essas habilidades permitem a pessoa criar histórias e ideias que ela tem.

Este trabalho aborda a importância da programação como uma habilidade essencial, não apenas pela sua relevância no mercado, mas também pela capacidade de transformar ideias em realidade no ambiente virtual.

3 Objetivo

A meta do trabalho é desenvolver um ambiente que proporcione uma experiência imersiva e estimulante aos usuários, projetando desafios e quebra-cabeças que explorem conceitos fundamentais de algoritmos e lógica de programação, de modo estimular a aprendizagem aos jogadores.

4 Fundamentação Teórica

Foi realizada uma pesquisa abrangente sobre como utilizar o Game Design Document (GDD) no ambiente de trabalho, com o intuito de identificar modelos e documentação que detalhem sua elaboração.

Além disso, foi feito uma pesquisa minuciosa na documentação da Game Engine Unity, visando uma compreensão aprofundada dessa ferramenta também na linguagem C# para

uma compreensão mais sólida da linguagem de programação. Durante o decorrer do projeto, caso haja dúvida será consultado a documentação novamente.

Para o desenvolvimento dos scripts do jogo foi utilizado o ambiente virtual da Microsoft o Microsoft Visual Studio como IDE principal no trabalho.

Na Unity foi utilizado o “*Scene Manager*”, que sua principal função é fazer o gerenciamento das cenas do jogo, além da utilização dos elementos de UI (*User Interface*) da Unity, que nos permite criar interfaces de usuários para o jogador interagir com elas.

Para otimizar a utilização da Unity no desenvolvimento de jogos 2D, foram consultados recursos do livro "UNITY: Design e Desenvolvimento de Jogos" de William Pereira Alves, e aplicado conceitos desse livro na criação do game.

Um dos principais objetivos do jogo é despertar a curiosidade do jogador e ensinar conceitos básicos de algoritmos e lógica de programação. Para embasar esse aspecto, foi feita uma pesquisa aprofundada em diversas fontes, incluindo o livro 'Lógica de Programação e Algoritmos com JavaScript' do autor Edécio Fernando Iepsen, publicado pela editora Novatec Editora em 2022. Além disso, buscando uma compreensão abrangente dos temas de linguagem e lógica de programação, foi consultado o livro 'Entendendo Algoritmos: Um Guia Ilustrado Para Programadores e Outros Curiosos'.

Para a prototipagem de telas, foi feito uso de software *Canva* onde foi feito esboço do mapa e da tela da mecânica. A arte do jogo foi elaborada com recursos adquiridos no site *itch.io*, sendo as *pixel arts* de LimeZu e *finalbossblues*, também com o uso do *LibreSprite*, onde foram criadas outras artes.

5 Trabalhos Similares

Um dos trabalhos semelhantes é o de Pasqueline Dantas Scaico, que tem como objetivo a criação de um "serious game" (jogo sério) voltado para o ensino de programação a alunos do ensino médio, utilizando a plataforma Android. No jogo, o protagonista enfrenta o desafio de uma prova de programação no dia seguinte, porém, ele acaba adormecendo e não consegue estudar. Devido a esse sono, ele começa a sonhar que está dentro de um castelo. Dentro desse castelo, ele encontra diversos desafios relacionados à programação, e à medida que supera cada desafio, ele adquire conhecimento em programação. Além dos desafios de programação, o jogo também inclui elementos de exploração do castelo, onde o protagonista precisa mover objetos para solucionar os desafios. Também existem elementos de combate no jogo. Este jogo tem como objetivo tornar o aprendizado de programação uma experiência divertida e interativa para os alunos do ensino médio, incentivando o interesse deles por essa área e promovendo a aquisição de conhecimento de forma envolvente.

No trabalho intitulado "Utilizando Unity para Desenvolvimento de Jogos 2D" escrito por Gabriel Batistela Berto em FEMA no campus “José Santilli Sobrinho”, no ano de 2017, a plataforma Unity é utilizada para criar o jogo de luta "Falkisgate Fight", com o intuito de proporcionar entretenimento aos jogadores. O trabalho abrange a elaboração de um documento específico de jogos, o GDD (Game Design Document), que descreve detalhes e diretrizes

cruciais para o desenvolvimento do jogo. E o jogo trará as seguintes opções para o jogador que é enfrentar a inteligência artificial do jogo ou competir contra outro jogador.

Em outro trabalho com título "Serious game como ferramenta de ensino de lógica de programação para crianças" de Luís Filipe Severgnini publicado em Caxias do Sul no ano de 2016, o objetivo é desenvolver um serious game utilizando a plataforma Unity para o ensino de lógica de programação. Para alcançar esse objetivo, o autor investiga a relação entre os serious games e a aprendizagem, sob a perspectiva da teoria da Zona de Desenvolvimento Proximal de Vygotsky. Além disso, ele discute a importância da diversão nos jogos sérios. Assim esse trabalho busca a criação de um serious game na Unity visando tornar o aprendizado de lógica de programação mais atrativo.

6 Metodologia

Elaboração de protótipos e desenvolvimento de jogos por meio de uma pesquisa qualitativa e descritiva.

7 Desenvolvimento

Para alcançar os objetivos do jogo, foi feito um processo bem definido, dividido em etapas cruciais. Inicialmente, foi o *Game Design Document* (GDD). Este documento servirá como o mapa mestre que orientará todo o desenvolvimento do jogo. Nele, delinearemos com precisão os objetivos do jogo, detalhando as mecânicas que o sustentarão, a trama, os personagens, o ambiente em que o jogo se desenrolará, seu estilo e a plataforma que pretendemos alcançar.

Em seguida, foi feita fase de prototipagem. Onde foram pegos os conceitos do GDD e transformados em protótipos, das telas do jogo. Isso incluirá a definição da mecânica de jogo, o mapeamento de cenários e a distribuição de itens e desafios. Essa etapa é fundamental para visualizar o jogo e otimizar seu design.

Com os protótipos como guia, iniciou-se a parte de desenvolvimento do jogo na Unity. Este é o núcleo do processo, em que se iniciou o desenvolvimento dos protótipos de telas. O objetivo é transformar os protótipos em realidade, criando telas que reflitam fielmente o conceito inicial do jogo.

Com a versão inicial do jogo em mãos, será realizado testes para a identificação de possíveis problemas, bugs e oportunidades de aprimoramento na mecânica do jogo. Simultaneamente, o código-fonte do jogo é revisado e otimizado, visando torná-lo mais eficiente. Isso engloba a eliminação de gargalos de desempenho, redução do uso de memória e a implementação de melhorias gerais de velocidade e a término da revisão passaremos para a fase final do desenvolvimento, que será a compilação do jogo para desktop.

Para auxiliar os jogadores nos desafios, foi criada uma apostila abordando os conceitos do jogo, incluindo explicações sobre lógica de programação e exercícios para reforçar o aprendizado.

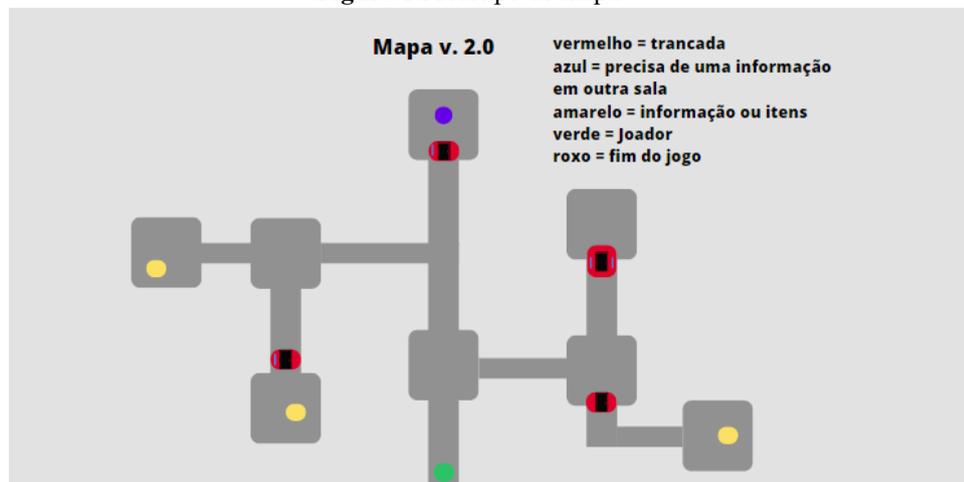
7.1 GDD

Foi elaborado a GDD onde foi documentado a história do jogo, jogos similares, o gênero do jogo, um resumo da *gameplay* do jogo, criação do personagem e ações que eles devem ter, gênero do jogo, plataforma do jogo, mapa e controles do jogo.

7.2 Prototipagem

Primeiramente, foram criados os protótipos de tela, delineando o mapa da fase e estabelecendo a distribuição dos desafios para o jogador, ver figura 1. Além disso, foi definido o caminho que o jogador deveria percorrer.

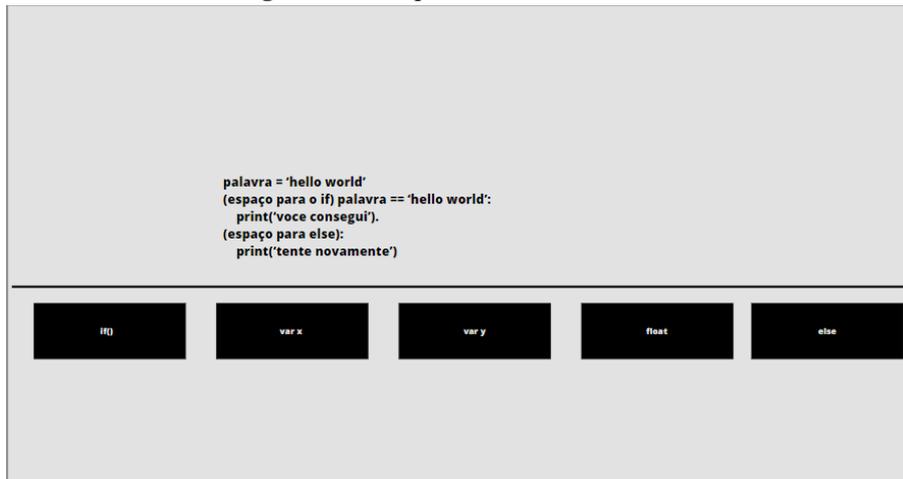
Figura 1 Protótipo do mapa



Fonte: Elaborado pelo autor, no aplicativo Canvas

Também foi desenvolvido um protótipo da mecânica dos quebra-cabeças que o jogador encontraria durante o jogo, ver figura 2. A mecânica foi criada em forma de completam blocos vazios com componentes necessários para completar os algoritmos.

Figura 2 Protótipo da tela do desafio



Fonte: Elaborado pelo autor, no aplicativo Canvas

7.3 Desenvolvimento do jogo

Primeiramente foi feito o menu inicial do jogo que tem duas funções de iniciar o jogo e fechar ele, ver figura 3. Para carregar a próxima cena foi utilizado o *Scene Management*, para carregar a fase 1, e para fechar o jogo, após a compilação foi utilizado uma função da Unity para fechar a aplicação.

Figura 3 Menu Inicial do jogo



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Após a criação do menu foi elaborado a primeira fase, utilizando as assets dos criadores finalbossesblues e LimeZu, ver figura 4, e nessa parte foram projetadas as *layers* de jogador e objetos sólidos, sendo elas para a separação dos *Game Objects*, assim podendo utilizar scripts para fazer os *Game Objects* poder ser interagir entre si. Com a *layers* criadas

foram feitos os, scripts de movimento e colisão com objetos sólidos e anexados as layers de jogador e objetos sólidos.

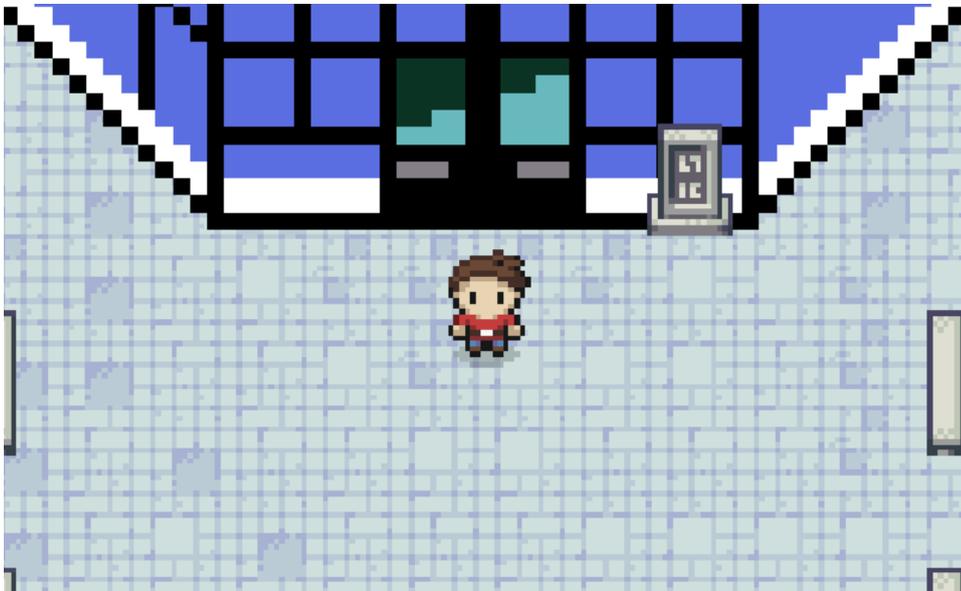
Figura 4 Tutorial do jogo



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Após a elaboração da fase de tutorial, foi feito o primeiro puzzle, que jogador terá que realizar para entrar na empresa. Ele será acessado, assim que o jogador interagir com a porta, utilizando a tecla de interação, ver figura 5.

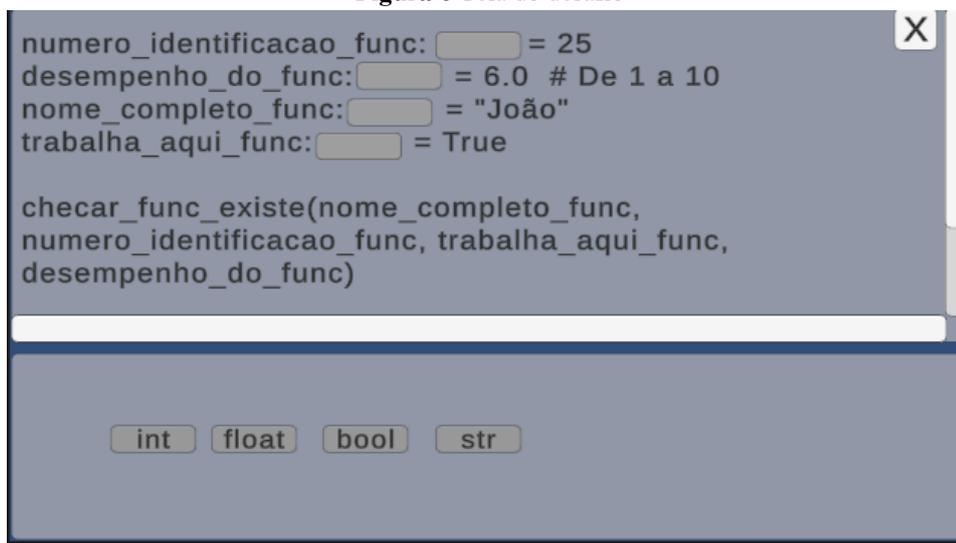
Figura 5 Entrada da empresa



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Com a cena do quebra-cabeça acessada, o jogador poderá arrastar os componentes para completar os algoritmos, ver figura 6.

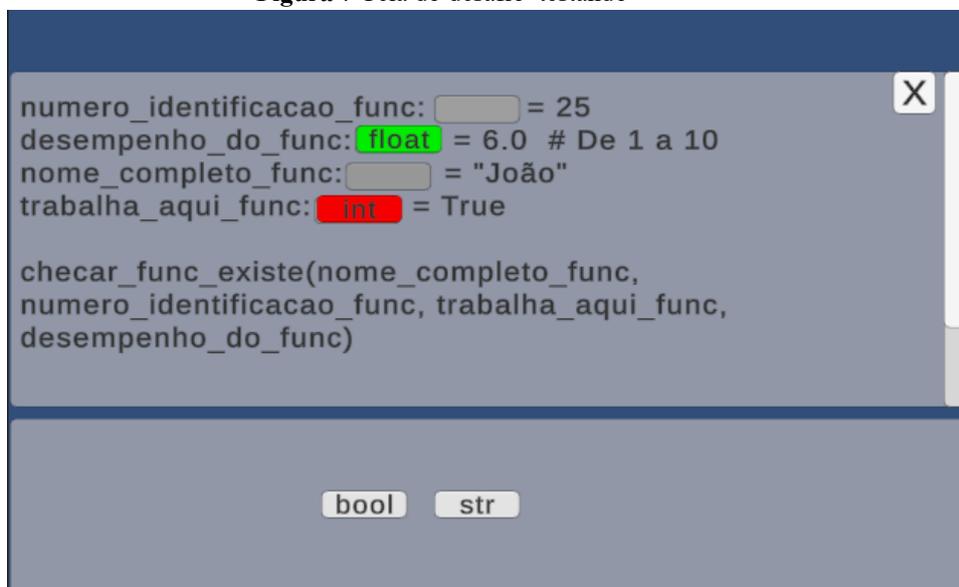
Figura 6 Tela do desafio



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Dentro do desafio, quando o jogador arrasta os componentes corretos para os locais certos, esses componentes tornam-se verdes. Se ele posiciona a estrutura no local errado, ela torna-se vermelha, ver figura 7. Quando o jogador completa todos os espaços com os componentes corretos, avança para a próxima fase.

Figura 7 Tela do desafio testando



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Essa dinâmica, apresentado na figura 7 anteriormente, foi implementada utilizando componentes de UI, como botões móveis. Quando esses botões se aproximam do local certo, mudam de cor, e a variável *isCorrect* é ajustada para *true*. Para cada botão, foi associado um script, e foi criada a função "checar" para verificar se o desafio está certo. Quando o desafio,

tiver correto, será exibido uma explicação do que foi feito, após isso, ver figura 8, o jogo mudara para a próxima fase.

// Código de movimentação de botões

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Mover_button : MonoBehaviour
{
    public void move()
    {
        this.transform.position = new Vector3(Input.mousePosition.x,Input.mousePosition.y,0
    );
    }
}
```

// Código de verificar botões

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class BOOL: MonoBehaviour
{
    public string cenaDestino;
    public Text explicacao;
    public Text puzzleName;
    private bool dialogoAtivo = false;
    public Image DialoguePanel;

    //Antigo
    public Image feedbackImage1;
    public Transform correctPosition1;
    public Transform wrongPosition1;
    public static bool isCorrectBool = false;

    private void Start()
    {
        feedbackImage1.color = new Color(0xEC / 255f, 0xEC / 255f, 0xEC / 255f);
    }

    private void Update()
    {
        Checar(INT.isCorrectInt, FLOAT.isCorrectFloat, isCorrectBool, STR.isCorrectStr);
    }
}
```

```
public void OnButtonDragged()
{
    if (Vector3.Distance(transform.position, correctPosition1.position) < 20f)
    {
        isCorrectBool = true;
        feedbackImage1.color = Color.green;
    }
    else if (Vector3.Distance(transform.position, wrongPosition1.position) < 20f)
    {
        isCorrectBool = false;
        feedbackImage1.color = Color.red;
    }
    else
    {
        isCorrectBool = false;
        feedbackImage1.color = new Color(0x9E / 255f, 0x9E / 255f, 0x9E / 255f);
    }
}

public bool Checar(bool isCorrect1, bool isCorrect2, bool isCorrect3, bool isCorrect4)
{
    if (isCorrect1 && isCorrect2 && isCorrect3 && isCorrect4)
    {
        StartCoroutine(IniciarDialogo());
    }
    return false;
}

private System.Collections.IEnumerator IniciarDialogo()
{
    // Exibir mensagem de diálogo na tela
    explicacao.text = "Nesse desafio voce, descobriu qual são os tipos de cada variavel, int
= para numeros, sem casa decimal, float = para numeros, com casa decimal, bool = para
valores True ou False e String = que são os caracteres";
    puzzleNane.text = "Tutorial";
    DialoguePanel.color = new UnityEngine.Color(255, 255, 158);

    explicacao.enabled = true;
    puzzleNane.enabled = true;

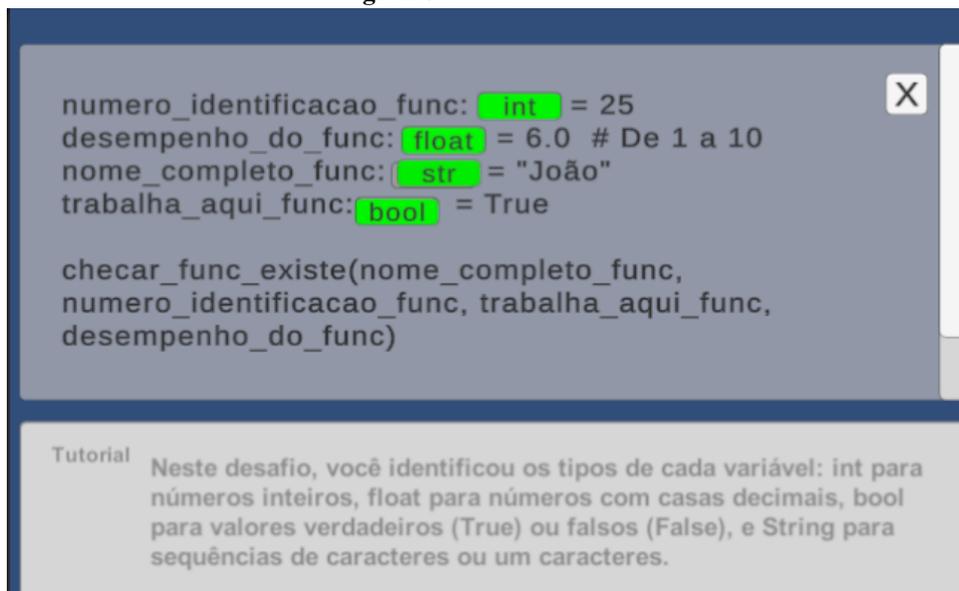
    dialogoAtivo = true;

    // Aguarde um curto período de tempo (pode ser ajustado)
    yield return new WaitForSeconds(10f);

    // Reseta o diálogo
```

```
dialogoAtivo = false;  
explicacao.enabled = false;  
  
// Carregue a cena de destino  
SceneManager.LoadScene(cenaDestino);  
  
}  
}
```

Figura 8 Desafio Resolvido



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

E na tela de mecânica tem um botão de sair, representado por um “X”, podendo ser visto na figura 8 anteriormente, quando pressionado carrega a cena anterior, com o personagem, perto da porta.

Após criar a tela de menu, tutorial e mecânica foi desenvolvido a fase do mapa da, com base no protótipo da figura 1 apresentado anteriormente, e nessa parte foi desenvolvido mais desafios relacionados a estruturas de condição e repetição, para cada porta, ver figura 9.

Figura 9 Dentro da empresa



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

7.4 Apostila

Para a criação da apostila, foram utilizados como referência o livro 'Lógica de Programação e Algoritmos com JavaScript' do autor Edécio Fernando Iepsen e os conhecimentos do autor para sua elaboração.

Na apostila foi criado tópicos começando de assuntos mais simples como, o que é lógica de programação, até estruturas de condição e repetição, e ao fim de cada capítulo é proposto alguns exerci-os para que o leitor possa fixar o conteúdo, e futuramente aplicá-los durante o jogo.

Figura 10 Apostila



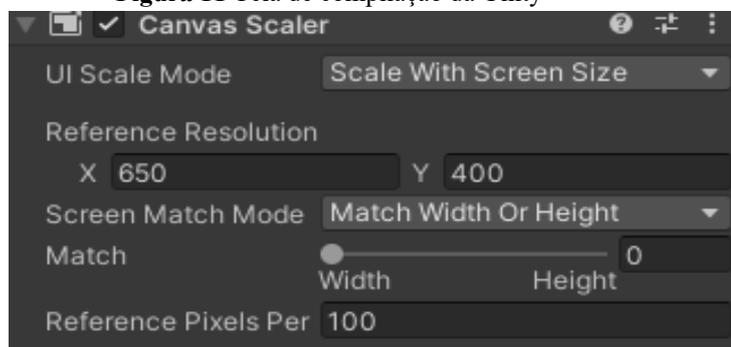
Fonte: Elaborado pelo autor, print screen da capa da apostila

7.5 Revisão e compilação do jogo

Após o término do desenvolvimento do jogo foi feita uma revisão nos códigos do jogo para localizar falhas ou melhorias, como no código da mecânica, ao contrário de atribuir a função checar para cada botão, foi atribuído apenas a um.

Além disso foi necessário usar o componente *Canvas Scaler* para poder definir uma resolução fixa para as telas, ver figura 11, pois como grande parte dos elementos são UI, esses elementos são flexíveis com a resolução, assim conforme a resolução aumenta o componente UI aumenta, e outros elementos presente nos componentes como textos e botões mudam de tamanho e ficam desproporcionais, devido a isso, foi travado a resolução.

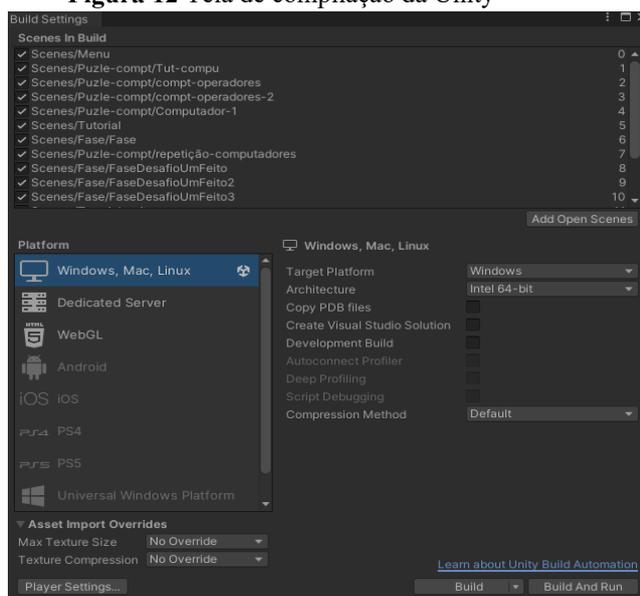
Figura 11 Tela de compilação da Unity



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

Com término da revisão do código foi feita a compilação do jogo para a sua versão desktop, a compilação foi feita utilizando a Unity, ver figura 12.

Figura 12 Tela de compilação da Unity



Fonte: Elaborado pelo autor, print screen da aplicação no sistema operacional Windows 11

8 Resultados e Discussões

Um dos principais resultados do projeto foi a criação do jogo, que proporciona a aprendizagem de algoritmos e lógica de programação por meio de desafios que exploram os conceitos dessas áreas e necessitam da lógica de programação do jogador para a resolução dos desafios.

Com a criação da apostila, a experiência do jogo melhorou significativamente, pois agora a resolução dos quebra-cabeças tornou-se mais eficiente. Através dela, os jogadores podem consultar os tópicos relacionados a cada assunto, visualizar exemplos de como utilizar cada componente e realizar exercícios para fixar o conteúdo. Isso proporciona uma abordagem mais prática para a aprendizagem durante o jogo.

Durante o desenvolvimento do jogo, ocorreram alterações significativas, como, no sistema de dificuldade, em vez de permitir que o jogador escolha a dificuldade no início, ela aumenta à medida que cada desafio é resolvido, incorporando conceitos de desafios anteriores com novos conteúdos.

Além disso, a mecânica do jogo passou por modificações. Inicialmente, a ideia era criar um sistema de desenvolvimento com um interpretador de texto semelhante a uma IDE. Contudo, ao longo do desenvolvimento, essa ideia foi simplificada para um sistema de blocos. Essa abordagem visa liberar o jogador da criação direta de código, permitindo que ele se concentre principalmente na lógica de programação.

Semelhante ao trabalho de Gabriel Batista Berto em 2017, a elaboração de um GDD e foi fundamental para a documentação do jogo. Esse processo contribuiu significativamente durante o desenvolvimento, pois vários aspectos do jogo já haviam sido definidos no documento.

9 Conclusões

Ao longo do trabalho foi desenvolvido um jogo, que pretende ensinar algoritmos e lógica de programação, no ambiente da Unity. A Unity demonstrou ser uma *game engine* excelente e intrigante para a criação desse game, destacando-se por suas funcionalidades robusta, facilidade de uso e ampla documentação da ferramenta. A escolha de empregar puzzles como ferramenta para ensinar lógica de programação, aliada a um material de suporte, trouxe uma experiência educativa completa e interativa.

Com a criação da apostila, tornou-se mais fácil explicar cada tema, pois além de abordar cada assunto de forma detalhada, ela também fixa os conteúdos por meio de exercícios, proporcionando uma experiência mais agradável para o jogador durante o jogo, logo que ele pode consultar os tópicos sempre que surgirem dúvidas durante as resoluções dos desafios.

Referências

ALVES, William Pereira. UNITY: Design e Desenvolvimento de Jogos. Alta Books, 2019.

BHARGAVA, Aditya Y. Entendendo Algoritmos: Um guia ilustrado para programadores e outros curiosos. Novatec Editora, 2018.

Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025.

Disponível em: <<https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>>.

Cloud City Tileset by finalbossblues. Disponível em: <https://finalbossblues.itch.io/cloud-city-tileset>. Acesso em: 4 dez. 2023.

GABRIEL BATISTELA BERTO. UTILIZANDO UNITY PARA DESENVOLVIMENTO DE JOGOS 2D. [s.l: s.n.]. Disponível em:

<<https://cepein.femanet.com.br/BDigital/arqTccs/1511320488.pdf>>. Acesso em: 14 out. 2023.

ITCH.IO. Download the latest indie games. Disponível em: <<https://itch.io/>>. Acesso em: 10 out. 2023.

IEPSEN, Edécio Fernando. Lógica de Programação e Algoritmos com JavaScript-2ª Edição: Uma introdução à programação de computadores com exemplos e exercícios para iniciantes. Novatec Editora, 2022.

Modern Office – Revamped – RPG Tileset [16x16] by LimeZu. Disponível em: <https://limezu.itch.io/modernoffice>.

SCAICO, Pasqueline Dantas et al. Implementação de um Jogo Sério para o Ensino de Programação para Alunos do Ensino Médio Baseado em m-learning. In: XX Workshop sobre Educação em Computação (WEI). 2012. p. 3-43.

SEVERGNINI, Luís Filipe. Serious game como ferramenta de ensino de lógica de programação para crianças. 2016.

TECHNOLOGIES, U. Plataforma de desenvolvimento em tempo real do Unity | 3D, 2D VR e Engine AR. Disponível em: <<https://unity.com/pt>>. Acessado em: 26 de maio de 2024

THE GIMP TEAM. GIMP. Disponível em: <<https://www.gimp.org/>>. Acesso em: 15 out. 2023.

WIKIPEDIA CONTRIBUTORS. Serious game. Disponível em:

<https://en.wikipedia.org/wiki/Serious_game>. Acesso em: 15 out. 2023.