

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2023/1

EXPLORANDO A LÓGICA DE PROGRAMAÇÃO ATRAVÉS DE UM JOGO COM AUXÍLIO DA REALIDADE AUMENTADA

Bruno Ricardo Junkes

Prof. Dalton Solano dos Reis – Orientador

1 INTRODUÇÃO

É notável que nos últimos anos, uma das áreas que mais tem crescido é a da computação, e com isso, exigindo cada vez mais profissionais qualificados para atuar em suas diversas subáreas. Contudo, sabe-se que esta não é uma área fácil, trazendo uma alta complexidade consigo e fazendo com que muitos tenham dificuldades em absorver conteúdos relacionados.

Devido a estas dificuldades, acaba-se culminando na desistência por grande parte dos estudantes de computação, muitas vezes resultante de metodologias defasadas de ensino utilizados por seus professores, complexidade da lógica de programação e falta de conhecimento de conceitos básicos (SOUZA; BATISTA; BARBOSA, 2016). Moraes, Mendes Neto e Osório (2020, p. 3) também apontam que “cada aluno possui suas dificuldades individuais, seu ritmo de aprendizagem, seus interesses e motivações, de forma que os professores precisam identificar as características de seus alunos e as dificuldades por eles enfrentadas para que consigam dar o suporte devido”.

Tendo estas dificuldades em mente, faz-se necessária a criação ou adaptação de métodos utilizados para lecionar as matérias que envolvam a área da computação, a fim de “proporcionar um ensino e aprendizagem de programação mais consistente e atraente para os alunos, inclusive com a inserção da programação na educação básica, na intenção que o aluno tenha acesso a conteúdos da computação” (SOUZA; FALCÃO; MELLO, 2021, p. 1265). Um conceito importante não só para a área, mas também para tarefas do cotidiano, que deve ser abordado é o da lógica de programação.

A lógica de programação pode ser entendida como um conjunto de regras e técnicas utilizadas para criar algoritmos, ou seja, sequências lógicas de passos que visam resolver determinados problemas ou realizar tarefas específicas. Segundo Batista, Fonseca e Horta (2020, p. 99) o ensino da lógica “beneficia o desenvolvimento do processo de aprendizagem, como o raciocínio lógico e a tomada de decisão, auxiliando na formação de ideias e resolução de problemas.”

Uma área que vem crescendo na computação, trazendo consigo um novo horizonte para o desenvolvimento de novas metodologias que visam auxiliar no ensino da lógica de programação, é a da Realidade Aumentada (RA). RA nada mais é, do que a combinação de elementos do mundo real e do virtual em tempo real, possibilitando uma melhora nos sentidos, principalmente da visão (ELVAS, 2018). Usar a tecnologia da RA neste contexto é favorável pois cria um ambiente descontraído, usando uma tecnologia relativamente nova, que chama a atenção e desperta o interesse das pessoas que interagem com ela. Não só isso, mas também sua facilidade de uso, pois, para executar uma simples aplicação é necessário apenas um aparelho que possua uma câmera, que nos dias de hoje se tornou algo comum com o uso de aparelhos como o celular.

Com base no exposto, este trabalho propõe-se a disponibilizar um jogo para dispositivos móveis que, utilizando da tecnologia de RA, possa ser usado como ponto de partida para o estudo da lógica nas fases iniciais de ensino e servir como um reforço para estudantes da área da computação. Para isso o jogador deverá, através do uso de blocos com comandos que representam ações, condições e outros conceitos da programação, reprogramar as chamadas entidades que são representadas por inimigos, armadilhas, acionáveis como alavancas e o personagem principal. Tendo como objetivo levar o personagem principal até o fim de cada nível proposto sem que ele seja atrapalhado por uma armadilha ou capturado por um inimigo.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar um jogo que seja capaz de introduzir e/ou melhorar o desenvolvimento da lógica de programação, tanto para crianças quanto para pessoas que já possuem algum conhecimento.

Os objetivos específicos são:

- a) disponibilizar um ambiente em RA que permita a interação com objetos 3D;
- b) dispensar o uso de marcadores para fixação dos cenários do jogo;
- c) avaliar a opinião dos usuários que interagem com o jogo sobre sua efetividade.

2 TRABALHOS CORRELATOS

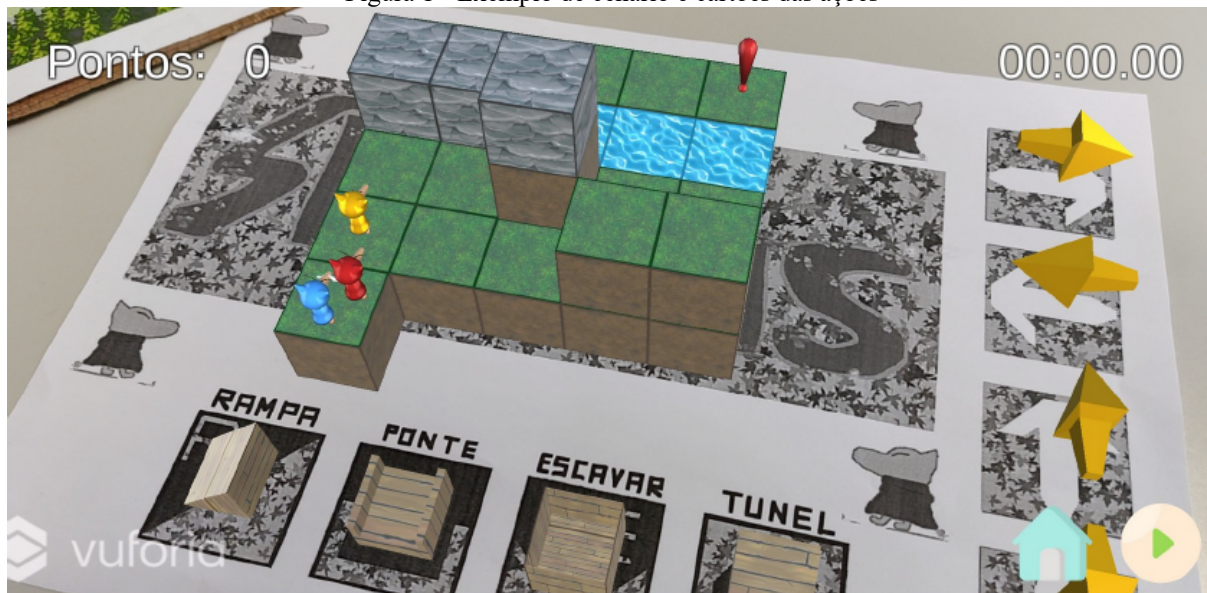
Nesta seção serão apresentados os trabalhos que se correlacionam com os objetivos deste projeto. Na subseção 2.1 é apresentado um jogo em RA que busca ajudar no treinamento do Pensamento Computacional (PC) (SARAIVA, 2022). A subseção 2.2 aborda um jogo em RA, que tem como objetivo, trazer de forma divertida os conceitos de lógica e programação (Mello e Antoniazzi, 2020). Por fim, na subseção 2.3 é apresentada a ideia de uma Interface de Usuário Tangível (IUT) com o intuito de trabalhar o conceito de PC utilizando como base o Furbot (KUNTZ, 2020).

2.1 BUILDING A GAME WITH AUGMENTED REALITY: FOR TRAINING COMPUTATIONAL THINKING

Saraiva (2022) traz um estudo sobre a RA e seu impacto na construção de jogos com o intuito de treinar o PC e com isso, desenvolver e testar um jogo com o objetivo de validar seus benefícios. O jogo em seu estado final, tem como métrica ajudar a desenvolver os quatro pilares do PC: reconhecimento de padrões, decomposição, algoritmos e abstração. Através do uso de cartões (Figura 1) que representam ações que um personagem denominado “Construtor” pode realizar, o jogador deve criar um caminho para levar todos os personagens até o final do mapa, sendo avaliados o tempo para chegar na solução e a quantidade de cartões utilizados.

Utilizando o motor de jogos Unity, foram construídas as cenas como a exemplificada na Figura 1 que trazem consigo os desafios que possuem diversas soluções e que devem ser desenvolvidas pelo jogador no intuito de resolvê-las. Com ela também foram desenvolvidos os *scripts*, conjunto de instruções a serem executados pelo programa que neste caso foram escritos inteiramente utilizando a linguagem C#. Estes *scripts* são responsáveis por comandar a movimentação e lógica das ações dos personagens, lógicas em geral e regras do jogo. A Unity possui um componente chamado *camera* que é responsável por criar a visualização do jogo em si, porém, o autor utilizou outra ferramenta baseada na plataforma Vuforia para trabalhar com o conceito da RA.

Figura 1 - Exemplo de cenário e cartões das ações



Fonte: Saraiva (2022).

A plataforma Vuforia realiza o processamento de imagens pré-cadastradas gerando marcadores virtuais, porque quanto mais ruídos a imagem possuir, melhor será a qualidade dos marcadores. De começo, foram utilizadas imagens muito específicas e detalhadas para testes, na qual mais tarde, foram substituídas por imagens criadas pelo autor para melhor representar as ações e cenários do jogo.

Segundo Saraiva (2022), o visual do jogo é uma parte importante, pois através dela é possível transmitir informações visuais ao jogador sem a necessidade de explicá-las em palavras. A partir de modelos e texturas prontos retirados da Asset Store, loja *online* da própria Unity, foi criada uma base para a modelagem dos objetos que compõem os personagens e cenário, todos editados a partir da ferramenta Blender, que permite criar animações, efeitos e modelos 3D. Outra ferramenta utilizada em cima das texturas dos blocos foi o Gimp, com o intuito de adicionar bordas pretas aos blocos para melhorar a visualização da divisão dos blocos. Para Saraiva (2022), o áudio possui a mesma importância do visual, portanto foram coletados diversos áudios para uso no menu, nas fases, na movimentação e em diversos outros pontos do jogo.

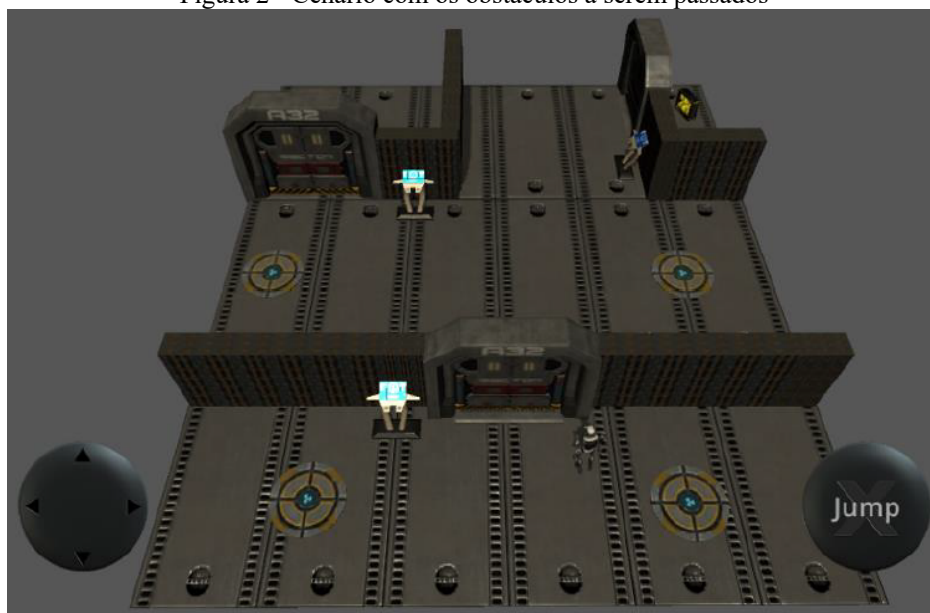
Por fim, Saraiva (2022) abre espaço para trabalhos futuros como uma melhoria geral dos modelos e texturas, adição de novos níveis com dificuldades elevadas, novos modos de jogo e geração de níveis aleatórios gerando maior valor para o jogo. Concluiu então que o objetivo de desenvolver um jogo em RA que colaborasse

com o estudo das quatro características do PC foi alcançado e que a partir dos estudos, pesquisas, apresentação e *feedbacks* coletados é realmente possível treinar habilidades de PC através de jogos.

2.2 JOGO COM UTILIZAÇÃO DE REALIDADE AUMENTADA VOLTADO PARA O DESENVOLVIMENTO LÓGICO APLICADO AO ENSINO FUNDAMENTAL E MÉDIO

Mello e Antoniazzi (2020) buscaram desenvolver um jogo para dispositivos móveis, utilizando os conceitos da tecnologia de RA, com o intuito de criar uma metodologia de ensino que proporcione um meio mais divertido e desafiador para estudantes sobre lógica e programação e aumentar o interesse nas áreas da informática, fornecendo um acesso facilitado ao estudo desses assuntos. O jogo propõe fases com diversos obstáculos, que devem ser resolvidas de forma lógica para levar o personagem (Robo Kyle) até seu destino (Figura 2), no qual a cada fase é apresentado um novo conceito relacionado a programação.

Figura 2 - Cenário com os obstáculos a serem passados



Fonte: Mello e Antoniazzi (2020).

O jogo foi desenvolvido com o uso da Unity 3D, seguindo duas etapas que envolveram a criação gráfica e a programação dos *scripts*, no qual devem trabalhar em conjunto para que o jogo seja executado como o esperado. Os objetos e o modelo do personagem foram extraídos da Asset Store e os *scripts* para cada objeto associado foram desenvolvidos pelos autores, utilizando a linguagem de programação C#.

Esses *scripts* contemplam as diversas funções do jogo, como a de movimentação do personagem que é tratada a partir da detecção de botões virtuais, movimentando-o vertical e horizontalmente, e suas interações com os objetos do ambiente a partir de gatilhos causados pela colisão com os objetos. Também foi desenvolvida uma interface que a cada fase é apresentada contendo uma nova informação sobre programação, a fim de ser aplicada na fase selecionada e, com a conclusão dela, será utilizado para responder uma pergunta que, se respondida corretamente, desbloqueará a próxima fase.

Para visualizar o jogo com uso da RA foi criado um marcador utilizando a plataforma Vuforia, que a partir do processamento de uma imagem, cria marcadores virtuais que podem ser importados na Unity para uso posterior. Assim, com o uso da câmera de um celular, é possível apontar para a imagem pré-processada e montar cada fase nela, permitindo ao jogador utilizar os botões criados para movimentar o personagem pelo cenário.

Atualmente, o jogo se encontra em estado parcial de desenvolvimento, no qual Mello e Antoniazzi (2020) afirmam que criarão mais fases, realizarão um aprimoramento das texturas e iluminação, e farão novos testes a fim de detectar possíveis erros. Com sua conclusão, planeja-se apresentar o jogo em escolas com o intuito de aplicar questionários, no qual poderá ser avaliada a eficácia dele na apresentação dos conceitos envolvidos na programação.

2.3 INTERFACE DE USUÁRIO TANGÍVEL PARA TRABALHAR COM O PENSAMENTO COMPUTACIONAL NO FURBOT

Kuntz (2020) inicia seu estudo apontando que nos últimos anos houve grandes mudanças no contexto educacional, marcados pelo uso predominante de novas tecnologias, na qual o ensino enfrenta o desafio de se adaptar a esses avanços. Também é dito por Kuntz (2020) que a introdução da tecnologia na escola pode trazer o benefício de desenvolver o PC, que traz consigo o empoderamento dos estudantes, permitindo que eles descrevam,

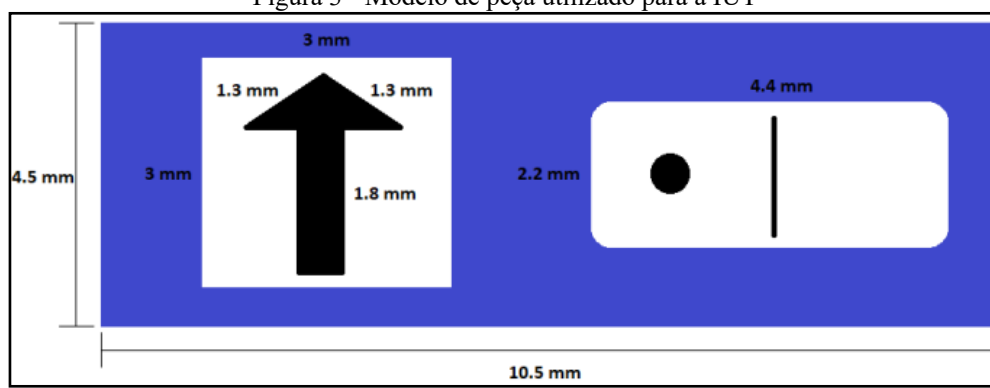
expliquem e resolvam problemas complexos por meio da análise lógica, representação abstrata e discussão de variáveis e estruturas condicionais.

A partir disso, teve-se como objetivo criar um módulo de IUT para o Furbot, jogo no qual se deve programar um robô com o intuito de chegar ao fim da fase tomando cuidado com obstáculos e com a energia do robô, proporcionando uma imersão na resolução de exercícios, com programação de movimentos, interface 2D, simulação e criação de peças de ações.

Para o desenvolvimento de cada fase, o jogador possui duas formas de programar a movimentação do robô, sendo elas através de um editor em que é inserida uma sequência de comandos ou utilizando detecção de contornos de uma imagem capturada. Esses comandos são compilados, passando por análise léxica e sintática que caso forem válidos, são convertidos e executados na linguagem da aplicação.

Para a utilização da IUT, Kuntz (2020) criou parâmetros para as peças que podem ser utilizadas para o desenvolvimento dos algoritmos, em que todas devem possuir um formato retangular com dimensões específicas, como o exemplificado na Figura 3. Para a peça principal que indica o início da verificação e execução dos comandos, deve existir um ícone de *play* no centro e ser da cor vermelha. Já as peças de movimentação e repetição foram divididas em dois lados, o lado direito utiliza um dominó para indicar quantas vezes deve-se executar o comando e do lado esquerdo a direção que o robô se movimentará ou a indicação de repetição.

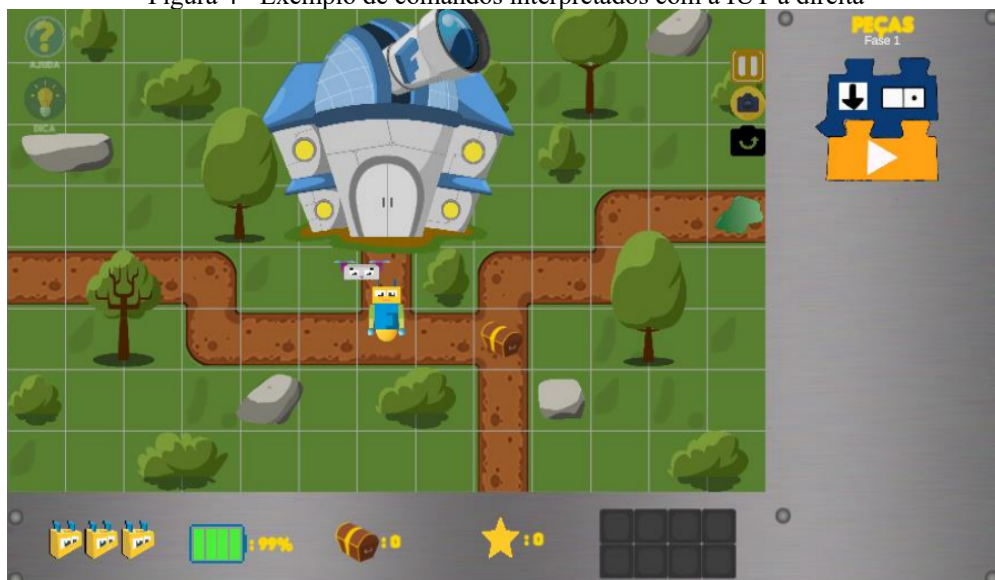
Figura 3 - Modelo de peça utilizado para a IUT



Fonte: Kuntz (2020).

Para que o jogador possa utilizar a IUT é preciso antes realizar algumas configurações na opção “Menu” disponibilizada na interface principal do jogo, no qual o jogador deve realizar a identificação da peça principal na câmera que se pretende usar como *display* do jogo. Durante o jogo, a câmera selecionada é ativada e realiza a captura das peças posicionadas, realizando a identificação dos comandos e listando na parte direita da tela a sequência de execução deles (Figura 4).

Figura 4 - Exemplo de comandos interpretados com a IUT a direita



Fonte: Kuntz (2020).

Diversas tecnologias foram utilizadas para a criação e interpretação da IUT, a começar pelo motor gráfico Unity, utilizando *scripts* em C# juntamente ao pacote OpenCV For Unity que oferece métodos para conversão de

matrizes para Texture2D, processamento e reconhecimento de imagens. Para acessar as câmeras disponíveis e realizar a captura das imagens foram utilizados componentes da própria Unity chamados WebCamTexture e WebCamDevice, na qual Kuntz (2020) realizou um processamento das imagens para assim identificar suas respectivas ações e para a criação das peças foi utilizado o software de edição de imagens Photoshop.

Segundo Kuntz (2020) o trabalho atingiu seus objetivos, apresentando uma aplicação Android que reconhece e executa comandos com base nas peças criadas pelos usuários, na qual a jogabilidade é aprimorada pela interface tangível, tornando-a mais atrativa e compreensível. Embora ainda existam limitações, como influências da luz ambiente, sombras, posição e ângulo das peças, que podem resultar em comandos errados. Também são apontadas possíveis melhorias como, aprimorar a interação para determinar quando os comandos devem ser executados, minimizar as limitações ambientais, reduzir o tempo de reconhecimento das peças, melhorar o reconhecimento em diferentes ângulos e permitir que os usuários cadastrem novas peças.

3 PROPOSTA DE JOGO

Esta seção abordará a proposta do jogo, justificando sua criação, definindo os requisitos funcionais e não funcionais a serem aplicados, e a metodologia e cronograma a serem seguidos.

3.1 JUSTIFICATIVA

No Quadro 1 – Comparativo dos trabalhos correlatos é possível visualizar um comparativo entre os trabalhos que se correlacionam com o presente trabalho, onde as linhas representam as características e as colunas os correlatos.

Quadro 1 – Comparativo dos trabalhos correlatos

Trabalhos Correlatos Características	Saraiva (2022)	Mello e Antoniazzi (2020)	Kuntz (2020)
Ensina por meio de um jogo	Sim	Sim	Sim
Público-alvo	Ensino Fundamental	Ensino Fundamental e Médio	Ensino Fundamental
Abordagem principal	Pensamento Computacional	Lógica e Algoritmos	Pensamento Computacional
Método de desenvolvimento da lógica	Usando cartões com ações	Usando botões virtuais	Usando cartões com ações
Utiliza Realidade Aumentada	Sim	Sim	Não
Dispensa o uso de marcadores	Não	Não	X
Ferramenta utilizada para trabalhar a RA	Vuforia	Vuforia	X
Plataformas disponíveis	Android	Android	Android

Fonte: elaborado pelo autor.

Conforme pode ser analisado no Quadro 1, todos os trabalhos correlacionados utilizam a abordagem de *gameificação*, trazendo jogos que possuem o intuito de promover a diversão, engajamento e motivação dos alunos do ensino fundamental e/ou médio, a aprenderem conceitos por volta da lógica de programação.

Os trabalhos de Saraiva (2022) e Kuntz (2020) trazem o conceito de PC, que busca trabalhar com quatro pilares: Decomposição, dividindo problemas complexos em partes gerenciáveis e compreendendo suas interações; Reconhecimento de Padrões, encontrando soluções eficientes em problemas similares; Abstração, buscando simplificar problemas, focando em criar modelos de soluções mais eficientes; Algoritmo, desenvolvendo sequências lógicas de passos para resolver problemas complexos. Em ambos são utilizados cartões representando ações e conceitos da programação para o desenvolvimento da lógica, porém, Saraiva (2020) utiliza da tecnologia de RA para apresentar as funções através de objetos criados computacionalmente e renderizados utilizando os cartões como marcadores. Enquanto Kuntz (2020) realiza a captura de imagens para posteriormente realizar um processamento sobre elas e definir quais funções foram inseridas pelo jogador na intenção de montar uma resolução para a fase proposta.

Já Mello e Antoniazzi (2020) focam mais no desenvolvimento da lógica e algoritmos através da resolução de cada fase proposta, por meio de botões virtuais que o jogador pode interagir para movimentar o personagem principal por portas que devem ser abertas por meio da interação dele com os painéis de ativação respectivos a cada porta. Além disso, apresentam pequenos textos no início de cada uma das fases no intuito de ensinar conceitos da área da computação, na qual são cobrados no formato de um questionário ao fim de cada uma, para mostrar que o jogador realmente adquiriu o conhecimento e permitindo que ele passe para a próxima fase. Mello e Antoniazzi (2020) também utilizam da RA em seu jogo, renderizando as fases sobre um marcador com o objetivo de tornar o jogo mais atrativo e interativo.

Como já mencionado, Saraiva (2022) e Mello e Antoniazzi (2020) utilizaram a tecnologia da RA em seus jogos e para isso trabalharam com a ferramenta Vuforia, que permite realizar o processamento de imagens pré-cadastradas para usá-las como marcadores, permitindo a sobreposição de objetos virtuais em um ambiente físico por meio de dispositivos móveis como celulares e tablets, que no caso destes trabalhos e de Kuntz (2020), tiveram como plataforma foco o Android, por ser mais acessível.

Estes trabalhos possuem algumas desvantagens, como já mencionado todos os jogos apresentados necessitam de elementos externos ao jogo, sejam marcadores para a utilização da RA ou peças físicas, o que acaba limitando o espaço em que pode ser jogado e possuem a limitação de plataforma, embora o Android seja mais acessível ao público, um grande percentual dos alunos não possuiriam acesso aos jogos. Após a comparação dos trabalhos, assim tendo em mente seus objetivos e desvantagens, este trabalho se propõe a disponibilizar um jogo para trabalhar a lógica de programação, permitindo que o jogador programe e re programe as entidades (personagem principal, inimigos e armadilhas), visualize as sequências criadas sendo executadas em tempo real e em caso de falhas, possa entender por meio de *logs* de erro qual foi a ação que fez com que o algoritmo não conseguisse completar a fase proposta. Com o intuito de aumentar o interesse e permitindo maior interação com o jogo, ele utilizará a tecnologia de RA, porém se apropriando do AR Foudation, um *framework* relativamente novo e não muito explorado que permitirá que o jogo seja renderizado em qualquer superfície plana e controlada, ou seja, que não possua muitas texturas, reflexões e variações na iluminação, não mais existindo a necessidade de utilizar elementos externos ao jogo dando-lhe maior liberdade de renderização e visando permitir que o maior número de pessoas o utilizem, será disponibilizado nas plataformas Android e iOS.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O jogo a ser desenvolvido deverá contemplar os seguintes Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

- a) permitir a navegação entre os menus principal, configurações e seleção de níveis (RF);
- b) permitir a locomoção em volta do cenário que compõe os níveis com o intuito de aumentar a interatividade com o jogo (RF);
- c) permitir a interação com as entidades dos níveis a partir do dispositivo móvel (RF);
- d) permitir que seja possível montar algoritmos utilizando os blocos com comandos com o intuito de desenvolver a lógica de programação (RF);
- e) permitir que seja possível visualizar os blocos que compõem o código de cada entidade (RF);
- f) permitir que seja possível visualizar os *logs* de erro que impediram o sucesso do nível (RF);
- g) permitir que os blocos do código das entidades sejam reprogramados (RF);
- h) permitir executar os códigos programados com os blocos, para visualizar em tempo real a lógica desenvolvida (RF);
- i) ser desenvolvido no motor de jogos Unity com uso do *framework* AR Foudation (RNF);
- j) ser desenvolvido para as plataformas móveis Android e iOS (RNF);
- k) renderizar os níveis em qualquer superfície plana controlada (sem reflexos, texturas ou muitas irregularidades) (RNF);
- l) salvar o progresso do jogador através da classe `PlayerPrefs` (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: pesquisar e estudar sobre lógica de programação e algoritmos, realidade aumentada, AR Foudation e trabalhos correlatos;
- b) criação do roteiro para os níveis: definir o que fará parte de cada nível, e qual a lógica que se espera ser aplicada para cada entidade presente, para que se possa chegar ao fim do nível;
- c) montagem dos níveis: utilizando o motor de jogos Unity, realizar a montagem dos níveis seguindo o proposto no roteiro, utilizando-se de *assets* retirados da própria loja da Unity e outros sites como Kenney;
- d) elicitação de requisitos: baseado no levantamento bibliográfico e nos objetivos do trabalho, reavaliar e/ou agregar novos requisitos;
- e) implementação da visualização dos níveis em RA: utilizando o AR Foudation, apresentar os níveis em quaisquer superfícies planas, permitindo a visualização dele em qualquer direção que o jogador esteja se movimentando;
- f) criação dos blocos de comando: desenvolver cada bloco com um determinado comando, que serão utilizados para montagem da lógica a ser aplicada em cada entidade do nível para resolução dele;
- g) criação da interface para a programação: criar uma tela que apresente os blocos de comandos disponíveis para programar a entidade selecionada, e os comandos já utilizados que serão executados posteriormente;
- h) detecção da entidade selecionada: ao apontar o celular/tablet em direção a uma entidade, destacá-la

no nível e, ao pressionar a tela do aparelho, a interface de programação será aberta para a entidade selecionada;

- i) implementação da leitura e interpretação dos comandos: cada bloco usado pelo jogador para montagem de seu programa, será convertido em linhas de comando que serão executadas pela entidade correspondente;
- j) execução e validação da lógica criada: a partir da interpretação dos comandos, o nível será executado e será avaliado se o jogador chegou no destino ou não, passando de nível ou tendo que encontrar uma lógica correta;
- k) implementação da interface de *log* de erro: criar uma tela que permitirá ao jogador, ver qual comando falhou em sua lógica e o porquê dessa falha ter acontecido;
- l) implementação das interfaces de navegação: implementar o menu do jogo, o menu de níveis, a interface dentro dos níveis que contemplarão diversos botões que darão acesso as interfaces criadas nas etapas anteriores;
- m) testes com usuários: realizar testes com usuários que possuam nenhum ou algum conhecimento de lógica de programação e analisar se o jogo foi divertido e conseguiu ensinar ou despertar o interesse pelo tema.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma de atividades a serem realizadas

etapas / quinzenas	2023									
	jul.		ago.		set.		out.		nov.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
criação do roteiro para os níveis										
montagem dos níveis										
elicitación de requisitos										
implementação da visualização dos níveis em RA										
criação dos blocos de comando										
criação da interface para a programação										
detecção da entidade selecionada										
implementação da leitura e interpretação dos comandos										
execução e validação da lógica criada										
implementação da interface de log de erro										
implementação das interfaces de navegação										
testes com usuários										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve brevemente sobre os assuntos que fundamentarão o estudo a ser realizado neste trabalho, no caso Lógica de Programação e Algoritmos, Realidade Aumentada (RA) e AR Foudation.

4.1 LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

Segundo Santos (2015) a lógica é um campo amplo de estudo que abrange o raciocínio e as leis do pensamento válido. Ela pode ser aplicada em atividades diversas e nas áreas das ciências exatas e matemática. É essencial para profissionais que buscam solucionar problemas e atingir metas eficientemente, utilizando recursos computacionais. O conhecimento em lógica é crucial para lidar com problemas administrativos, de controle, planejamento e estratégia, exigindo habilidades de raciocínio sólido. Já a lógica de programação conforme Xavier (2014, p. 21) “significa apenas contextualizar a lógica na programação de computadores, buscando a melhor sequência de ações para solucionar determinado problema.”

“Muitos recursos são usados pelos profissionais para representar o seu raciocínio lógico como os fluxogramas, os diagramas de bloco e até mesmo a linguagem natural e variações da língua nativa do profissional” (SANTOS, 2015, p. 26). Ainda segundo Santos (2015), a lógica de programação pode ser codificada em linguagens de programação seguindo padrões. Sendo a mais conhecida e utilizada, a programação estruturada é uma técnica importante que visa agilizar o desenvolvimento, facilitar a depuração, verificar erros e facilitar a manutenção dos programas. Ela envolve criar sequências de comandos, combinar blocos de instruções, compartilhar módulos e realizar revisões em equipe.

A lógica de programação está fortemente relacionada ao conceito de algoritmos, uma vez que ambos tem como premissa gerar sequências de passos lógicos e instruções que representam a solução para um problema específico. Segundo Santos (2015) um algoritmo pode ser representado de várias formas, mas o importante é que

seja compreensível e de fácil interpretação, na qual a clareza na descrição dele facilite sua depuração e correção de erros.

4.2 REALIDADE AUMENTADA

Segundo Tori, Hounsell e Kirner (2020) a ideia de aumentar a realidade utilizando recursos como espelhos, lentes e iluminações devidamente posicionados tem sido trabalhado desde o século XVII, porém, foi a partir de 1968 que o primeiro protótipo utilizando o conceito de RA foi criado por Ivan Sutherland, na qual permitia juntar imagens 3D geradas em computador sobre imagens reais. Embora diversas pesquisas abordando a RA já estavam sendo feitas, o termo somente foi desvinculado do conceito de Realidade Virtual (RV), que trata da completa imersão do usuário em um mundo virtual, em 1992 com o artigo de Tom Caudell e Tom Mizell, sendo em 2017 atribuído a Tom Caudell a criação do termo RA.

A RA é uma das subáreas da computação que se beneficiou da evolução da Tecnologia da Informação (TI), se tornando cada vez mais popular e acessível. Ela busca renderizar objetos virtuais 2D e/ou 3D no mundo real, permitindo com que as pessoas interajam com estes ao mesmo tempo que interagem com objetos do mundo real. Segundo Kirner e Siscoutto (2007), a realidade aumentada pode ser definida como um enriquecimento do mundo real, utilizando-se de elementos virtuais como textos, imagens dinâmicas e modelos 3D, sendo gerados por dispositivos tecnológicos em tempo real, com isso, permitindo que o usuário se sinta no mundo real, podendo interagir com elementos do mundo virtual.

Dependendo do contexto que se pretende utilizar a RA, podemos ter dois tipos de entradas: baseada em visão e baseada em sensores. A RA baseada em visão é mais robusta, flexível e fácil de utilizar, porém, possui problemas com iluminação e oclusão de informações. É nela que encontramos o recurso mais conhecido dentro da área, os marcadores, elementos utilizados para fornecer pontos de referência para a sobreposição dos objetos virtuais no mundo real. Já a RA baseada em sensores é mais precisa, utilizando sensores presentes nos dispositivos para fornecer informações e interações virtuais no mundo real, tendo inclusive recursos como acelerômetros, giroscópios, GPS, entre outros (TORI; HOUNSELL; KIRNER, 2020).

4.3 AR FOUNDATION

Podemos integrar a RA em dispositivos móveis usando um *framework* desenvolvido especificamente para esta tecnologia e seu Kit de Desenvolvimento de Software (SDK), um conjunto de ferramentas, bibliotecas e documentações, na qual possuem suas próprias especificações em que sua escolha é feita de acordo com critérios específicos. O AR Foundation é um exemplo de *framework* desenvolvido com este propósito.

Através do AR Foundation é possível criar experiências de RA para as plataformas Android e iOS a partir do motor de jogos Unity. Com esta tecnologia é fornecida uma Interface de Programação de Aplicativos (API) que une os principais recursos de RA como ARKit, ARCore, Magic Leap e HoloLens, possibilitando o rastreamento de imagens e objetos 3D, a detecção de planos e o reconhecimento facial (UNITY, 2023).

REFERÊNCIAS

- BATISTA, Riann Martinelli; FONSECA, Alexandre Ramos; HORTA, Euler Guimarães. Programação em Blocos: impacto de um projeto de extensão executado em Escolas públicas de Diamantina/MG. **Informática na educação: teoria & prática**, Porto Alegre, v.23, n.2, maio/ago. 2020. Disponível em: <https://seer.ufrgs.br/InfEducTeoriaPratica/article/view/96741>. Acesso em: 18 jun. 2023.
- ELVAS, Filipe Barbosa da Cunha Mendes. **Realidade Aumentada Aplicada a Panoramas Táticos**. 2018. Dissertação (Mestrado em Ciências Militares Navais) – Escola Naval, Alfeite, 2018. Disponível em: <https://comum.rcaap.pt/handle/10400.26/25097>. Acesso em: 23 mar. 2023.
- KIRNER, Claudio; SISCOOTTO, Robson. Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações. In: **Symposium on Virtual and Augmented Reality**, 2007, Petrópolis. Porto Alegre: SBC, 2007. p. 2-292. Disponível em: http://de.ufpb.br/~labteve/publi/2007_svrps.pdf. Acesso em: 7 abr. 2023.
- KUNTZ, Jonathan Michels. **Interface de usuário tangível para trabalhar com o pensamento computacional no FURBOT**. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Universidade Regional de Blumenau, Blumenau, 2020. Disponível em: https://www.furb.br/dsc/arquivos/tccs/monografias/2020_1_jonathan-michels-kuntz_monografia.pdf. Acesso em: 16 jun. 2023.
- MELLO, Luiz Filipe Durgion de; ANTONIAZZI, Rodrigo Luiz. Jogo com utilização de realidade aumentada voltado para o desenvolvimento lógico aplicado ao ensino fundamental e médio. **RevInt: Revista Interdisciplinar de Ensino, Pesquisa e Extensão**, Cruz Alta, v.8. 2020. Disponível em: <https://revistaelectronica.unicruz.edu.br/index.php/revint/article/view/386>. Acesso em: 13 mar. 2023.
- MORAIS, Ceres Germanna Braga; MENDES NETO, Francisco Milton; OSÓRIO, António José. Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática de literatura. **Research, Society and Development**, [s. l.], v.9, n.10, 2020. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/9287>. Acesso em: 23 mar. 2023.

SANTOS, Fabiano. **Lógica de Programação**. 1. Ed. Rio de Janeiro: Editora SESES, 2015. Disponível em: <http://hood.com.br/new/filegator/repository/Introdu%C3%A7%C3%A3o%20%C3%A0%20Programa%C3%A7%C3%A3o/10%20Livro%202.pdf>. Acesso em: 20 jun. 2023.

SARAIVA, Francisco Manuel Vital. 2022. Dissertação (Mestrado em Engenharia de Informática) – Escola de Engenharia, Universidade do Minho, Minho, 2022. **Building a game with Augmented Reality: for training computational thinking**. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/83352>. Acesso em: 20 de mar. 2023.

SOUZA, Draylson Micael; BATISTA, Marisa Helena da Silva; BARBOSA, Ellen Francine. Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. **Revista Brasileira de Informática na Educação**, São Paulo, v.24, n.1, 2016. Disponível em: <http://ojs.sector3.com.br/index.php/rbie/article/view/3317>. Acesso em: 23 mar. 2023.

SOUZA, Franciely Alves de; FALCÃO, Taciana Pontual; MELLO, Rafael Ferreira. O Ensino de Programação na Educação Básica: Uma Revisão da Literatura. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 32., 2021, Recife. **Anais [...]**. Recife: UFRPE, 2021. p. 1265-1275. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/18148>. Acesso em: 23 mar. 2023.

TORI, Romero; HOUNSELL, Marcelo da Silva; KIRNER, Claudio. Realidade Aumentada. *In*: **Introdução a Realidade Virtual e Aumentada**. 3. Ed. Porto Alegre: Editora SBC, 2020. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/book/66>. Acesso em: 20 jun. 2023.

UNITY. **AR Foundation**. 2023. Disponível em: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>. Acesso em: 17 abr. 2023.

XAVIER, Gley Fabiano Cardoso. **Lógica de Programação**. 13. Ed. São Paulo: Editora Senac, 2014. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=lxW8EAAQBAJ&oi=fnd&pg=PP5&dq=l%C3%B3gica+de+programa%C3%A7%C3%A3o&ots=47IqhldzV9&sig=4Kf4Fk5Wq3_eqdBDKlvUpNsvD0A#v=onepage&q&f=false. Acesso em: 20 jun. 2023.