Escola Técnica Estadual Presidente Vargas

MECA – FIGHT 2

Mogi das Cruzes, SP
2024

Isabela Sofia Vieira Leme
Lucas Takumi Nomura
Paulo Eduardo Mendes de Oliveira

Débora Liliane de Souza Renó

MECA – FIGHT 2

Relatório apresentado à 8ª FEMIC - Feira Mineira de Iniciação Científica.
Orientação da Prof.ª Dr.ª Débora Renó

Mogi das Cruzes, SP

2024

RESUMO

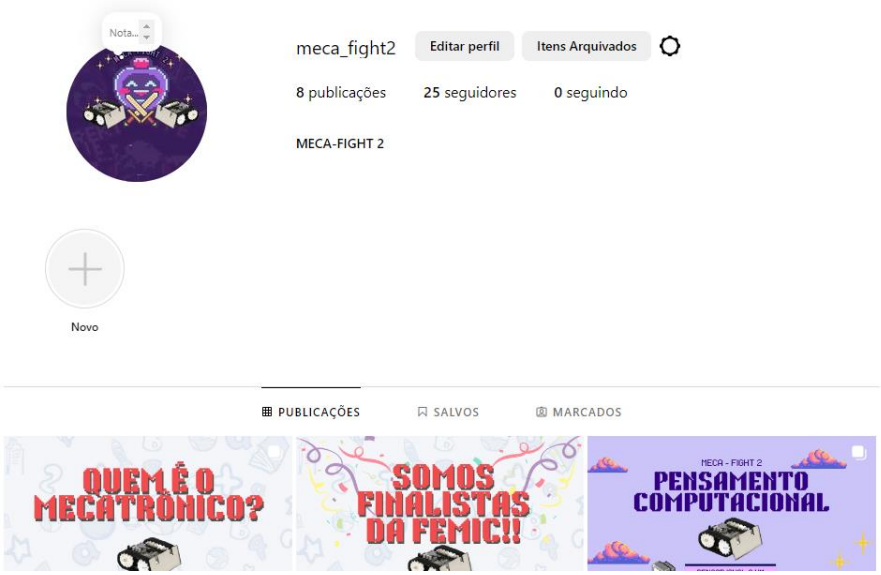
MECA - FIGHT 2 surgiu de um trabalho escolar proposto na componente curricular “Estudos Avançados em Ciências da Natureza e suas Tecnologias” em que equipes de estudantes eram desafiados a criar um jogo com a temática “máquinas” a partir da linguagem de programação em blocos em um aplicativo para dispositivos móveis denominado OctoStudio. Criar e programar um game consistia em uma das metas a ser atingida pelos estudantes do 3º ano do Ensino Médio da Escola Técnica Estadual Presidente Vargas nesta disciplina a fim de evidenciar de forma prática uma das áreas de atuação do profissional técnico formado em Mecatrônica. MECA - FIGHT 2 foi além deste objetivo geral, atingiu também os objetivos específicos que incluíam: aprofundar os estudos dos alunos em linguagem de programação e ampliar o interesse de outros estudantes pelo desenvolvimento de jogos e pelo curso técnico ofertado pela escola. Nosso game funciona na interface de dois celulares, um com o código para os jogadores controlarem seus personagens, e outro para contabilizar os pontos do jogo – todos os códigos criados por nossa equipe foram redigidos na linguagem de programação em blocos no aplicativo OctoStudio de disponibilização gratuita pelo Instituto de Tecnologia de Massachusetts. MECA - FIGHT 2 foi apresentado na Feira Tecnológica da Escola Técnica Estadual Presidente Vargas 2024 nos dias 17 e 18 de outubro, em Mogi das Cruzes. Nosso projeto contribuiu para o aperfeiçoamento do raciocínio lógico e da capacidade de solucionar problemas no cotidiano e na área técnica mecatrônica. Sugerindo, desta forma, que aplicativos de programação, como o utilizado em linguagem em blocos, é uma ferramenta educacional para o desenvolvimento de aprendizagem significativa mesmo em público de faixa etária mais jovem, pois não requer conhecimentos prévios em programação.

Palavras-chave: linguagem de programação em blocos, jogos, mecatrônica.

APÊNDICE 1 – Perfil do Instragram

A equipe criou um perfil na rede social, a fim de divulgar os resultados do projeto e mostrar assuntos pertinentes ao projeto, como pensamento computacional, como funciona a programação em blocos, e etc.

Figura 1 - perfil do MECA - FIGHT 2



Fonte: Instagram

Acesso ao perfil do instagram: https://www.instagram.com/meca_fight2/

APÊNDICE 2 – Tabela de links de publicações no Instagram

Tabela 1 – Links de publicações realizadas na página do Instagram do Projeto

MECA - FIGHT 2	
Publicação	Link
Programação do jogo (controle e personagens)	https://www.instagram.com/p/C_JmwcPSIWs/?igsh=MWoxYWMza2E1dWF1Nw==
Programação do jogo (pontuação e vencedor)	https://www.instagram.com/p/C_JsDIuSfXr/?igsh=MWplZGk0a3Rra28yeA==
Vídeo do funcionamento do jogo	https://www.instagram.com/p/C_ReJIHy-XZ/?igsh=MW5qYzFidng1Z2ZseQ==
Programação em blocos	https://www.instagram.com/p/C_W3-M2NkVy/?igsh=MXV4NGtmdnN2c3JmdQ==
Gamificação	https://www.instagram.com/p/C_b0Ji3tlh2/?igsh=c2V5cmQxd2wyZXN5
Pensamento Computacional	https://www.instagram.com/p/C_rJnSvyL3o/?igsh=MWludnlsdzUyNjM4eg==
Anúncio: Somos Finalistas da FEMIC!!	https://www.instagram.com/p/DAR0UyHymkD/?igsh=cWh3a21iZ2o5cDZq
Quem é o Mecatrônico?	https://www.instagram.com/p/DAyxh3aSC4J/?igsh=bXcweDc2aWkxbWJ5

Fonte: Autoria própria

APÊNDICE 3 – Apresentação do OctoStudio e proposta de trabalho feita em sala de aula. Através desta apresentação, tivemos o primeiro contato com o OctoStudio e, assim, foi apresentada a proposta de trabalho, que futuramente se tornou no projeto MECA – FIGHT 2

Figura 2 - Apresentação do OctoStudio e da proposta de trabalho

Fonte: Genially

Acesso disponível em: https://view.genially.com/661c6b7696e1540014e158cd/personal-branding-octostudio-3o-eacnt

APÊNDICE 4 – Integrantes da equipe e o jogo finalizado

Figura 3 - Foto com os integrantes do grupo e o jogo finalizado

Fonte: Autoria própria

APÊNDICE 5 – Equipe trabalhando no projeto

Figura 4 - Equipe trabalhando no jogo MECA –FIGHT 2

Fonte: Autoria própria

Tabela de figuras

Figura 1 - perfil do MECA - FIGHT 2....4
Figura 2 - Apresentação do OctoStudio e da proposta de trabalho....6
Figura 3 - Foto com os integrantes do grupo e o jogo finalizado....7
Figura 4 - Equipe trabalhando no jogo MECA –FIGHT 2 8
Figura 5 - Fluxograma das Etapas de realização do MECA – FIGHT 2. 11
Figura 6 - Variáveis de forma geométrica e seus tipos 12
Figura 7 - Blocos de controle e variáveis criadas 12
Figura 8 - Função delay 13
Figura 9 - Interface 1....14
Figura 10 - Interface 2....14
Figura 11 - Programação controle do robô1 15
Figura 12 - Programação controle do robô 2 15
Figura 13 - Programação do robô 1 16
Figura 14 - Programação do robô 2 16
Figura 15 - Programação do botão reset 17
Figura 16 - Programação do robô1 da interface 2....18
Figura 17 - Programação do robô2 da interface 2....19

SUMÁRIO

1 INTRODUÇÃO 5

2 JUSTIFICATIVA 7

3 OBJETIVO GERAL 9

4 METODOLOGIA 10

5 RESULTADOS OBTIDOS 14

6 CONCLUSÕES 20

REFERÊNCIAS 22

1 INTRODUÇÃO

Durante as últimas décadas, a revolução tecnológica nos trouxe vários benefícios através da criação de várias ferramentas, as quais facilitaram e simplificaram o cotidiano do ser humano, sendo possível se comunicar de diversas formas, como por exemplo pelo celular trocando mensagens, chamadas de voz ou pelas redes sociais, realizar a compra de produtos sem sair de casa, através dos sites de compra, pesquisas rápidas com a geração de um relatório bem consistente, como o ChatGPT entre outras coisas.

Com tudo isso, é possível perceber um ramo crescente nos últimos anos, sendo ele o de desenvolvimento de jogos. Uma área que ganhou credibilidade e muito interesse ao redor do mundo impulsionando várias empresas e instituições a tornar a criação de jogos mais acessível aos internautas, principalmente dos jovens estudantes, estabelecendo meios mais intuitivos para que se possa desenvolvê-los ao mesmo tempo em que haja o incentivo a imaginação e ao raciocínio lógico dos usuários.

Um grande exemplo disso é a linguagem de programação Scratch, criada em 2007 pelo Media Lab do MIT (Instituto de Tecnologia de Massachusetts). Com inspiração na linguagem LOGO, densenvolvida por Papert, a plataforma visa tornar a programação mais acessível e divertida para os internautas independente da idade. A interface conta com uma abordagem mais visual, ou seja, os usuários criam seus programas através do encaixe de blocos, os quais representam uma ação específica no projeto, como alterar uma variável, mover objetos, emitir sons, e etc, não exigindo o conhecimento prévio básico em linguagem de programação (WIKIPEDIA, 2024).

Observamos, também, que vários projetos inclusivos foram e estão sendo desenvolvidos, na tentativa de formular um novo meio de aprendizado que seja dinâmico, divertido e efetivo nos alunos, sendo um deles justamente utilizando os jogos e o desenvolvimento desses, como já disse Mitchel Resnick (2017), criador do programa Scratch, “A sociedade precisa agir e pensar de maneira criativa e inovadora. Isso começa no jardim de infância”.

Além disso, podemos citar o projeto OctoStudio. Criado pelo grupo Lifelong Kindergarten, uma plataforma que utiliza da linguagem Scratch voltada para jovens e

iniciativas e defesa juvenil. Promovendo a criatividade e o empoderamento dos infanto-juvenis, e ajudando a cultivar a próxima geração de inovadores e líderes, disponibilizando a expressão criativa a todos, especialmente aqueles de comunidades marginalizadas (SWEENEY,2024).

2 JUSTIFICATIVA

Percebemos que desde o advento da internet no mundo, as tecnologias desenvolvidas estão cada vez mais envolvendo a facilidade do usuário em acessá-las e poder criá-las por conta própria, isso é, utilizar da tecnologia para criar seu próprio conteúdo como, jogos, sites, blogs entre outros trazendo boas oportunidades para que esse internauta use sua criatividade e ganhe conhecimento por conta própria. Mostrando que, atualmente, é essencial ter conhecimento sobre as várias tecnologias disponíveis e se atualizar sobre elas.

Um estudo realizado pela Universidade da Geórgia (UGA), pelos autores Abesadze & Nozadze (2020), mostra que há vários anos se discutia diferentes formas para melhorar os processos de aprendizagem, sendo observado que era necessário atualizar esse sistema. Uma das soluções encontradas foi introduzir aulas de programação (codificação) desde a primeira série como disciplina independente nas escolas, ou seja, trazendo tecnologias atuais e que demonstram ser essenciais para a sociedade, sendo possível concluir que, além dos estudantes desenvolverem habilidades pertinentes ao século XXI, foi demonstrado melhora no rendimento das aulas, tornando-as mais divertidas e interativas, inclusive, aperfeiçoando o raciocínio lógico deles.

No entanto, percebe que, na realidade contemporânea, a educação escolar não evoluiu de acordo com as tecnologias presentes no mercado, mantendo sua estrutura tradicional e, muitas das vezes, maçante. Bloqueando o pensamento criativo e cognitivo dos alunos, impedindo que as tecnologias sejam aproveitadas adequadamente por eles.

A criatividade e curiosidade são competências essenciais nos dias atuais, contudo, nem todos os estudantes experimentam a liberdade de expressão que a tecnologia oferece. Sendo possível reconhecer dois cenários, um com a aprendizagem formativa aliada a tecnologia, resultando em experiências com maior interesse e autoeficácia, e outro com a educação com base nos sistemas opressivos da sociedade, resultando em bloqueios na criatividade e interesse (Brown & Brown, 2020; Love, 2019).

Portanto, uma investigação das ferramentas disponíveis na rede, na área de linguagem de programação, nos possibilita que aprofundemos nossos conhecimentos no

objeto de estudo enquanto esse cenário se reverte, proporcionando a formação de conhecimentos relevantes e estimulando o pensamento criativo e cognitivo dos jovens.

3 OBJETIVOS

3.1 Objetivo geral

Explorar os recursos da linguagem de programação em blocos, a fim de aprofundar os conhecimentos na área, facilitando a aprendizagem da área do curso, encontrando meios intuitivos para o público – alvo (interessados em desenvolvimento de jogos, programação e mecatrônica) começar os estudos sem a necessidade de conhecimento técnico prévio, de forma criativa e divertida, além de evidenciar, de forma prática, uma das áreas do profissional técnico formado em mecatrônica.

3.2 Objetivos específicos

Estabelecido o objetivo geral, os objetivos específicos definidos foram:

- Aplicar aprendizados adquiridos no desenvolvimento do projeto nas aulas de mecatrônica, seja diretamente ou indiretamente;
- Expor o projeto em feiras educacionais, como o da FEMIC e da Etec Presidente Vargas;
- Despertar o interesse de outros estudantes pelo desenvolvimento de jogos e pelo curso técnico em Mecatrônica ofertado pela escola.

4 METODOLOGIA

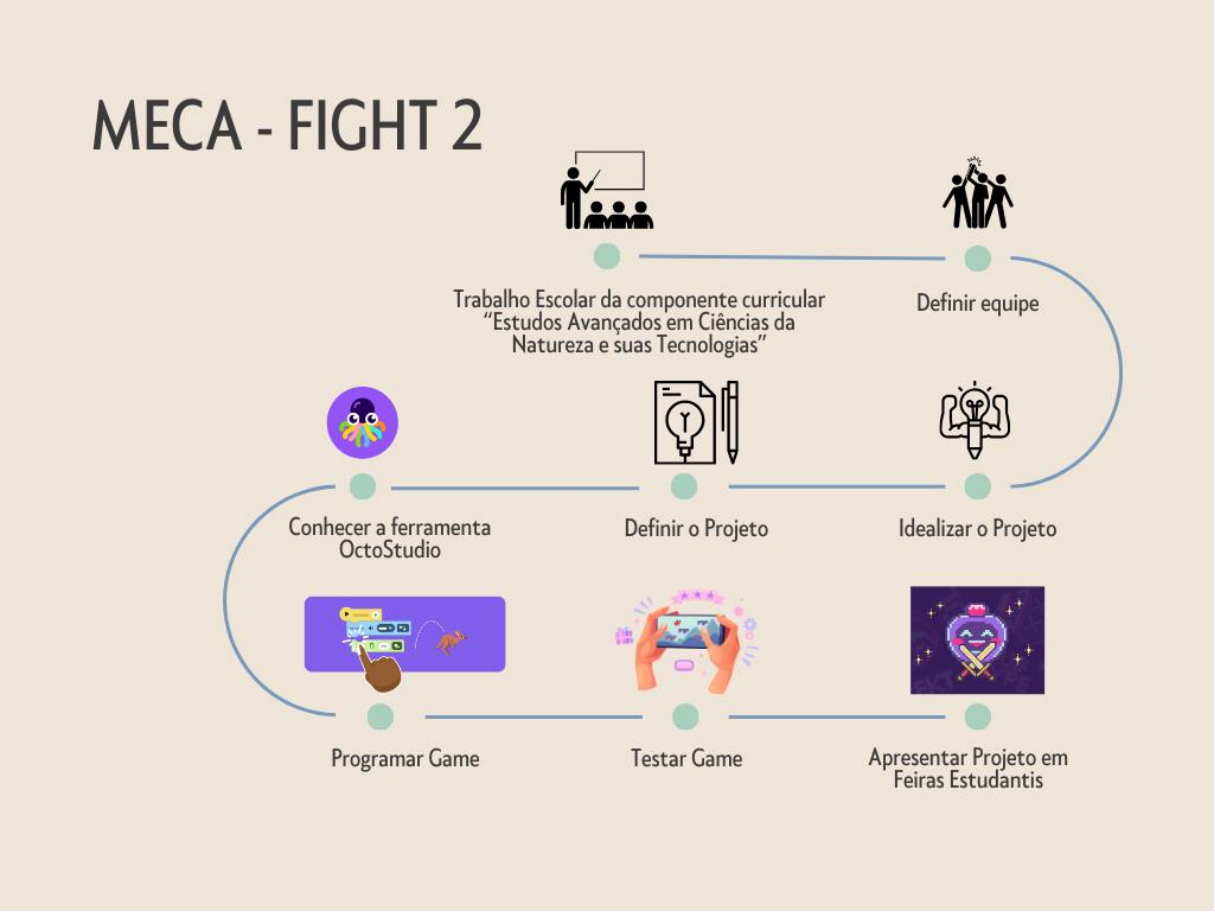
Nossa equipe buscou utilizar do procedimento experimental de pesquisa para a realização do projeto, realizando testes para analisar variáveis e suas respectivas combinações e verificando os resultados obtidos.

A proposta surgiu de um trabalho escolar no componente curricular “Estudos Avançados em Ciêncais da Natureza e suas Tecnologias” onde as equipes foram desafiadas a criar um jogo com a temática “máquinas” a partir da linguagem de programação em blocos no software OctoStudio, disponível para dispositivos móveis. A partir desse desafio, a equipe aderiu a ideia de desenvolver um jogo inspirado nos jogos da Nintendo® “Street – Fighter” dos consoles “retrô” populares das décadas de 80 e 90.

Com isso, o nome do projeto surgiu, “MECA – FIGHT 2”, significando luta (FIGHT) mecatrônica (MECA), e assim percebemos que além de ser uma boa oportunidade para se divertir, esse experimento serviria, também, para aprofundar os conhecimentos na área de mecatrônica, principalmente em linguagem de programação, e atrair pessoas interessadas em desenvolvimento de jogos, sendo uma das propostas divulgar o programa em feiras educacionais, como o da Escola Técnica Estadual Presidente Vargas.

Os testes foram sendo realizados com tentativas, erros e acertos, uma vez que nenhum dos integrantes da equipe detinha conhecimento do funcionamento do software, portanto, buscamos primeiramente organizar nossas ideias, tendo como foco planejar o funcionamento do jogo. O jogo teria a seguinte dinâmica: os jogadores teriam o controle do personagem para movimentá-lo para frente, pressionando o botão na tela do dispositivo, e a colisão dos personagens faria com que eles se repelissem, os afastando do centro da tela, assim o jogador que conseguisse fazer o outro ser empurrado para a borda da tela ganharia a partida. Assim, isso seria desenvolvido em duas interfaces: uma possibilitando a visualização dos personagens e o controle deles, e outra com a contagem de pontuações e indicação do vencedor da partida do jogo. Na Figura 1, temos a representação das etapas do projeto em forma de fluxograma, a fim de melhor entendimento e análise dos processos realizados.

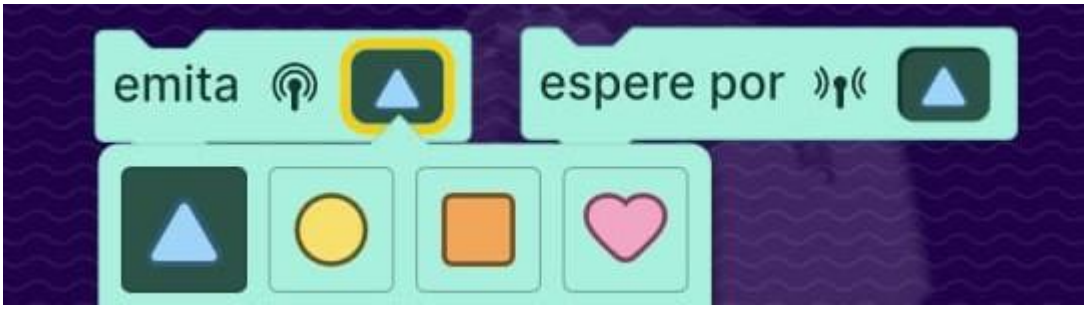
Figura 5 - Fluxograma das Etapas de realização do MECA – FIGHT 2.



Os testes foram realizados aos poucos, priorizando primeiramente o funcionamento da programação e lógica do projeto. No entanto, algumas dificuldades foram enfrentadas durante esse processo, sendo elas envolvendo principalmente a transmissão e recebimento de variáveis e conflito de informações internos e entre os dispositivos.

No primeiro momento, para que houvesse a comunicação entre os personagens e os controles do jogo era necessário que variáveis específicas para cada objeto fossem transmitidas, permitindo a interação do sistema. Para isso, foram utilizadas, inicialmente, os blocos "emita (forma geométrica)" e "espere por (forma geométrica)". Contudo, dois problemas foram encontrados: a programação necessitava de 6 variáveis distintas, quantidade além da disponibilizada pelo software (4 variáveis), e o tempo de resposta e conflito de variáveis era muito grande, comprometendo o funcionamento do programa.

Figura 6 - Variáveis de forma geométrica e seus tipos



Fonte: OctoStudio

A solução para isso foi substituir esses blocos pelos blocos de controle, os quais tinham funcionamento parecido, entretanto, funcionavam transmitindo e recebendo variáveis de texto, ou seja, poderiam ser criadas quantas variáveis fossem necessárias. Assim as 6 variáveis puderam ser criadas e o tempo de resposta foi encurtado.

Figura 7 - Blocos de controle e variáveis criadas

Fonte: OctoStudio

Concluindo-se a programação da primeira interface, dos personagens e dos controles, iniciamos o desenvolvimento da segunda interface, da pontuação e indicação do vencedor. Para que houvesse a contagem de pontos pelo programa, era necessário que, novamente, as variáveis comunicassem quando o personagem encostava na borda da tela do dispositivo, assim, um único obstáculo foi encontrado: os blocos de controle não serviriam para a comunicação entre dispositivos diferentes, ou seja, nessa ocasião, os blocos “emitir (forma

geométrica)” e “espere por (forma geométrica)” deveriam ser utilizados.

Como último ajuste, foi necessário a determinação de uma função “delay” (atraso de tempo) antes da transmissão do bloco de controle, já que, sem ela, havia conflito de comandos dentro do sistema, comprometendo o sistema, fazendo com que personagens movimentassem mesmo sem os comandos, trocassem a ordem de movimentação, e etc. Portanto, foi estabelecido um “delay” de 0.2 segundos, tempo mínimo suficiente para evitar o conflito de dados e não comprometer o funcionamento do jogo.

Figura 8 - Função delay

Fonte: OctoStudio

5 RESULTADOS OBTIDOS

O projeto finalizado consiste em um jogo de luta com a temática “máquinas” com duas interfaces:

- Controle e visualização dos personagens;

Figura 9 - Interface 1

Fonte: OctoStudio

- Contagem de pontuação do jogo e indicação de vencedor;

Figura 10 - Interface 2

Fonte: OctoStudio

Cada objeto do jogo possui sua própria programação:

- Controles: espera-se que o jogador aperte o controle, se esse for executado, o comando espera um “delay” de 0.2 segundos, evitando o conflito de dados, e assim a variável “mensagem 1” ou “mensagem 2” é enviada para os demais componentes.

Figura 11 - Programação controle do robô1

Fonte: OctoStudio

Figura 12 - Programação controle do robô 2

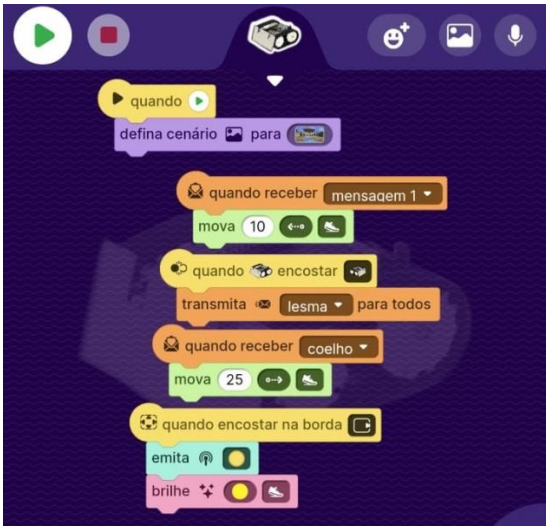
Fonte: OctoStudio

- Personagens (robôs): os robôs recebem a variável de seus respectivos controles, no caso, “mensagem 1” ou “mensagem 2”, e em seguida movem-se 10 unidades de locomoção para frente. Em caso de colisão entre os robôs, uma variável indicando essa ação é transmitida. No robô1 temos a variável “coelho” sendo transmitida, já no robô2, a variável “lesma” é transmitida. Assim, quando os robôs receberem as variáveis, esses recuarão 25 unidades para trás, de acordo com a variável recebida. Por fim, assim que um dos robôs encostar na borda da tela, esse emitirá uma variável em forma geométrica (triângulo ou círculo) para o segundo dispositivo e brilhará para indicar qual robô teve a sua derrota.

Figura 13 - Programação do robô 1

Fonte: OctoStudio

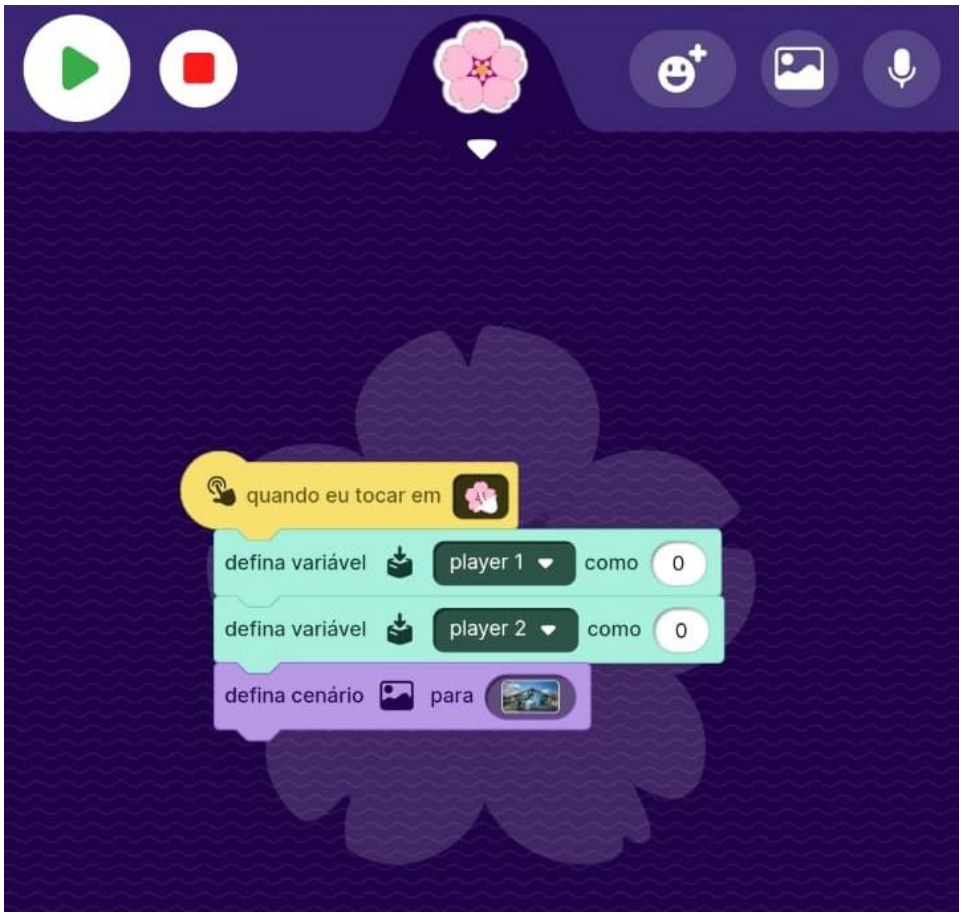
Figura 14 - Programação do robô 2



Fonte: OctoStudio

- Botão reset (flor): quando esse botão é pressionado, o código define as variáveis “player 1” e “player 2” com o valor inicial de 0, essas variáveis serão responsáveis para a contagem de pontos no jogo, além de ajustar o cenário para a cena inicial, preparando o jogo para o seu início.

Figura 15 - Programação do botão reset

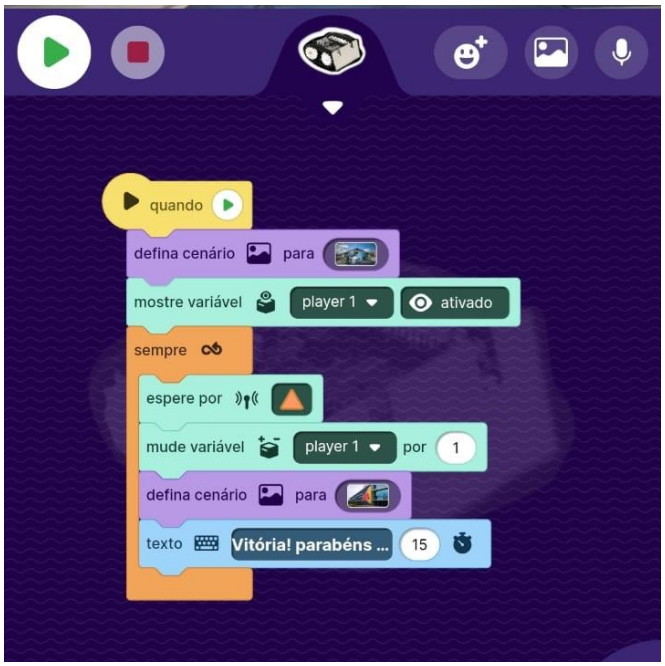


Fonte: OctoStudio

- Robô 1: Define o cenário inicial quando o programa se inicia e ativa a variável de contagem de pontos (player 1), com isso, o programa entra em um “*loop* infinito” (o programa sempre fará essas funções a não ser que o programa seja interrompido). Dentro do “*loop* infinito” a variável de forma geométrica (triângulo) é esperada a ser transmitida pelo dispositivo 1, quando o dispositivo 2 reconhecer essa variável é atribuída à variável “player 1” a soma de 1 unidade, contabilizando os pontos feitos na partida, e além disso o

cenário muda com o texto “Vitória! Parabéns player1!” por 15 segundos. Isso se repete até que os dispositivos sejam encerrados ou o botão de reset seja acionado.

Figura 16 - Programação do robô1 da interface 2



Fonte: OctoStudio

- Robô 2: possui programação semelhante ao do robô 1, a única diferença são as variáveis nele presentes, sendo a variável de forma geométrica triângulo sendo substituída pra a variável de forma geométrica círculo, contabilização feita pela variável “player2”, e cenário e texto distintos, todas adaptadas para indicar a vitória do player2.

Figura 17 - Programação do robô2 da interface 2

Fonte: OctoStudio

6 CONCLUSÕES

A proposta do trabalho aparentava ser bem simples, no entanto, esse conseguiu demonstrar um grande potencial para se transformar em um projeto de desenlvolvimento de um jogo. Inicialmente o objetivo era apenas realizar o trabalho da componente curricular, contudo, percebemos que era possível aproveitar esssa oportunidade para aprofundar os conhecimentos na área de linguagem de programação, análisá-la, e criar um programa que atraísse a atenção do público-alvo, formando aprendizados de forma criativa e dinâmica e expondo a área do nosso curso técnico, Mecatrônica.

Os conhecimentos aprendidos durante o processo de construção do programa puderam ser aplicados positivamente nas aulas do curso técnico em Mecatrônica, demonstrando um rendimento melhor para raciocinar logicamente diante de problemas e resolvê-las. Uma vez que, a programação em blocos, com seu jeito intuitivo, prático e visual de se programar, facilitou a visualização de como montar e planejar os processos dentro de uma lógica construtiva, assim, nós, alunos de Mecatrônica, pudemos imaginar as etapas de cada problemática a serem resolvidas com muito mais facilidade, as organizando por setores, assemelhando a cada um dos objetos no programa do jogo, e pensando no processo em que cada um desses setores seria responsável, ou seja, a programação de cada um desses.

Aproveitamos as oportunidades para apresentar o MECA – FIGHT 2 na Feira Tecnológica da Escola Técnica Estadual Presidente Vargas e na 8ª FEMIC.

Concluímos que o software OctoStudio mostra ser uma ótima sugestão para introduzir os estudos em linguagem de programação, já que, apesar de ser indicado para um público infantil, esse software apresenta uma vasta gama de jogos que podem ser desenvolvidos sem o conhecimento prévio na área.

Além disso, é possível perceber que os blocos de controle e os blocos “emitir” e “receber” formas geométricas, apesar de terem funções de declaração de variáveis semelhantes, possuem diferenças. Os blocos de controle são ideiais para a troca e definição de variáveis internamente, ou seja, em um mesmo dispositivo, uma vez que é possível criar inúmeras condições e seu tempo de resposta aparenta ser mais curto, no entanto, esse bloco não se comunica quando se trata de interligar dois dispositivos ou mais, sendo uma

desvantagem. Nesse caso, os blocos de “emitir” e “receber” formas geométricas são bastante úteis, já que estabelecem conexão entre dispositivos e utilizam o sinal *bluetooth* para se comunicarem, ou seja, essa função não se mostra muito interessante para comunicação interna, já que tem tempo de resposta muito lento utilizando do sinal *bluetooth* do dispositivo, mas bem eficiente para a conexão entre vários aparelhos móveis.

Portanto, o aplicativo OctoStudio demonstra ser ótimo para introduzir e melhorar os conhecimentos em linguagem de programação, tanto que pudemos aplicar os conhecimentos adquiridos durante os experimentos nas aulas do curso, pois esse, possui uma interface intuitiva, para que todos possam entender seu funcionamento mesmo sem conhecimento prévio, e visual, fazendo com que as funções sejam auto-explicativas, facilitando para o usuário criar seu próprio programa. Várias funções estão disponíveis no *software*, mesmo que algumas sejam parecidas, cada uma tem sua especifidade e melhor campo de atuação, possibilitando que os programadores usem sua criatividade para criar, praticamente, qualquer jogo possível.

REFERÊNCIAS

ABESADZE, S; NOZADZE, D. Make 21st Century Education: The Importance of Teaching Programming in Schools. International Journal of Learning and Teaching, Geórgia, 3 set. 2020. Vol. 6, No 3, p. 158 – 162.

Brown, K. D., & Brown, A. L. (2020). Anti-Blackness and the School Curriculum. In C. A.

Criador do programa Scratch conversa sobre linguagem de programação. Secretaria Municipal de Educação. São Paulo, 16 novembro, 2017. Disponível em: https://educacao.sme.prefeitura.sp.gov.br/noticias/criador-do-programa-scratch-conversa-sobre-linguagem-de-programacao/#:~:text=O%20membro%20integrante%20do%20MIT,inf%C3%A2ncia%20para%20o%20processo%20criativo. Acesso em: 12 outubro, 2024.

Scratch. Wikipedia. 29 agosto, 2024. Disponível em: https://pt.wikipedia.org/wiki/Scratch. Acesso em: 12 outubro, 2024.

SWEENEY, D. Report to the President Year ended June 20, 2024 MIT Media Lab, p -5, 2024.

8ª Feira Mineira de Iniciação Científica