



Universidade Federal de São Carlos - UFSCar
29 de julho a 02 de agosto de 2024

UMA AVENTURA PELA MATEMÁTICA NO MUNDO DO JOGO DO LABIRINTO

Sabrina Silva de Andrade - Graduanda do curso de matemática da Universidade Federal de Catalão - UFCAT

Wesley Silva de Andrade - Graduando da Universidade Federal de Catalão

Daniel da Silveira Guimarães - Professor da Universidade Federal de Catalão

Thiago Porto de Almeida Freitas - Professor da Universidade Federal de Catalão

Contatos: sabrina_andrade@discente.ufcat.edu.br; wesleyandrade@discente.ufcat.edu.br;
danielguimaraes@ufcat.edu.br; thiagoporto@ufcat.edu.br.



Uma Aventura pela Matemática no Mundo do Jogo do Labirinto

Sumário

- Introdução;
- Objetivos;
- Metodologias;
- Resultados;
- Conclusão.

Introdução

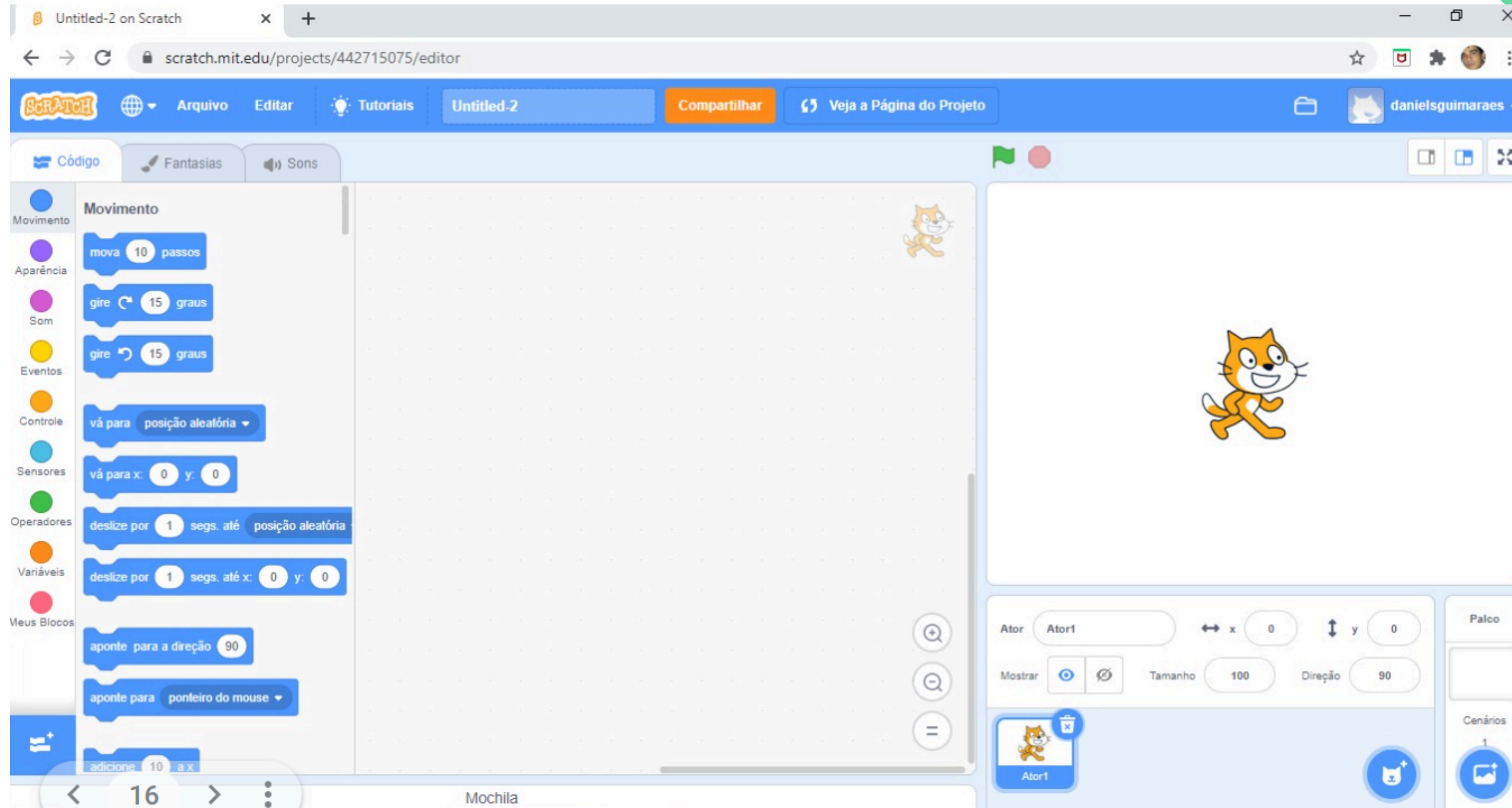
- Este trabalho apresenta um relato de experiência desenvolvido no âmbito do Programa Institucional de Bolsas de Iniciação à Docência (PIBID) da Universidade Federal de Catalão (UFCAT)
- Tal experiência teve como premissa abordar **temas matemáticos importantes** no 8º ano na confecção de um **jogo digital** utilizando a plataforma Gratuita do **Scratch**.

Mas o que é Scratch?

O Scratch é uma linguagem de programação criada em 2007 pelo Media Lab. do MIT, sendo disponível no formato online. De forma divertida e de fácil manuseio, já que a forma de linguagem de programação que utiliza são blocos com as palavras chaves para serem inseridas na programação. Podemos criar nele animações, jogos e outros programas interativos.

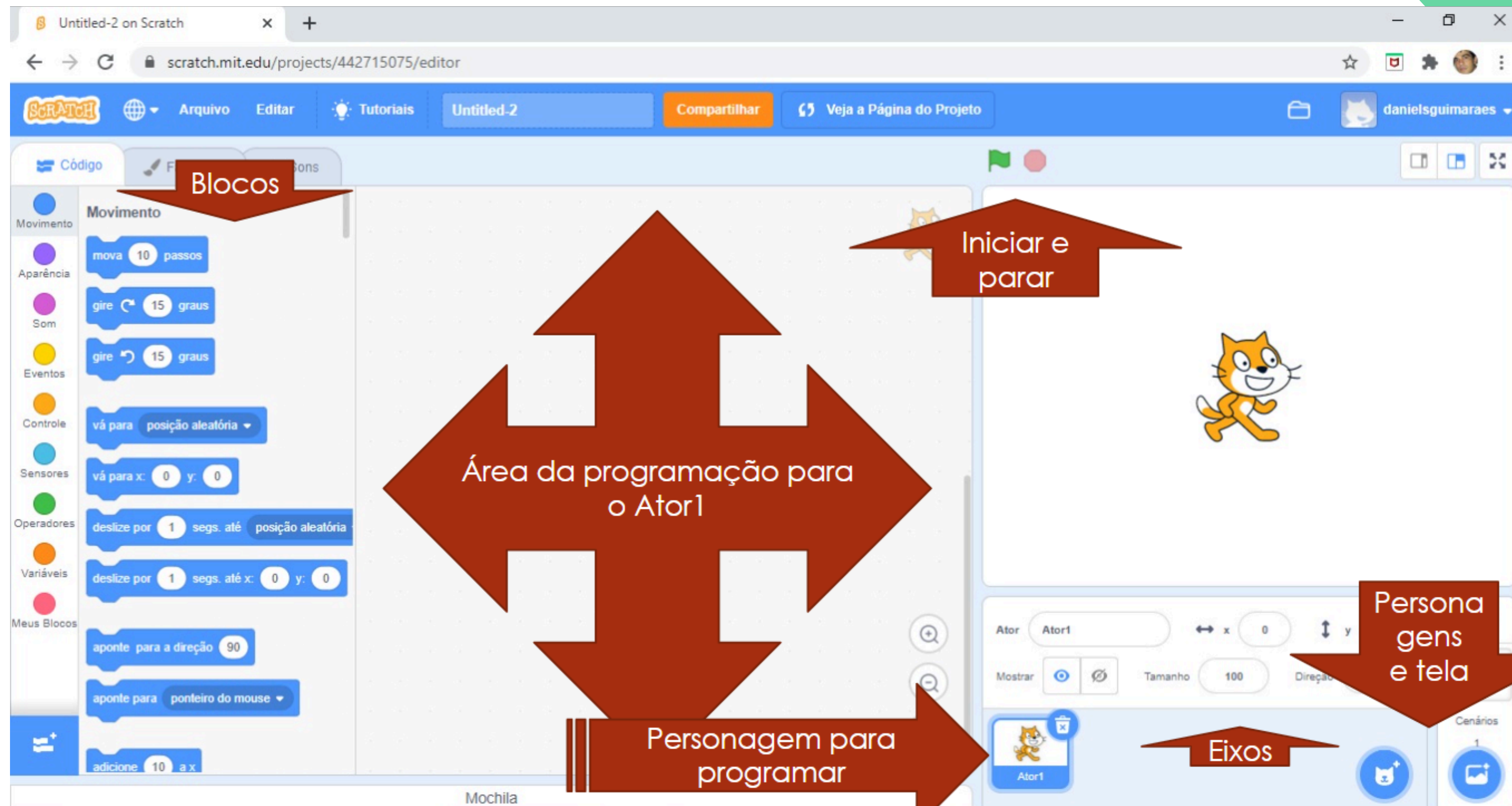
Atualmente, **o Scratch está disponível online** e pode ser acessado em: <https://scratch.mit.edu/>.

Figura 01: Página de programação do Scratch.



Fonte: Acervo do PIBID Matemática/UFCAT.

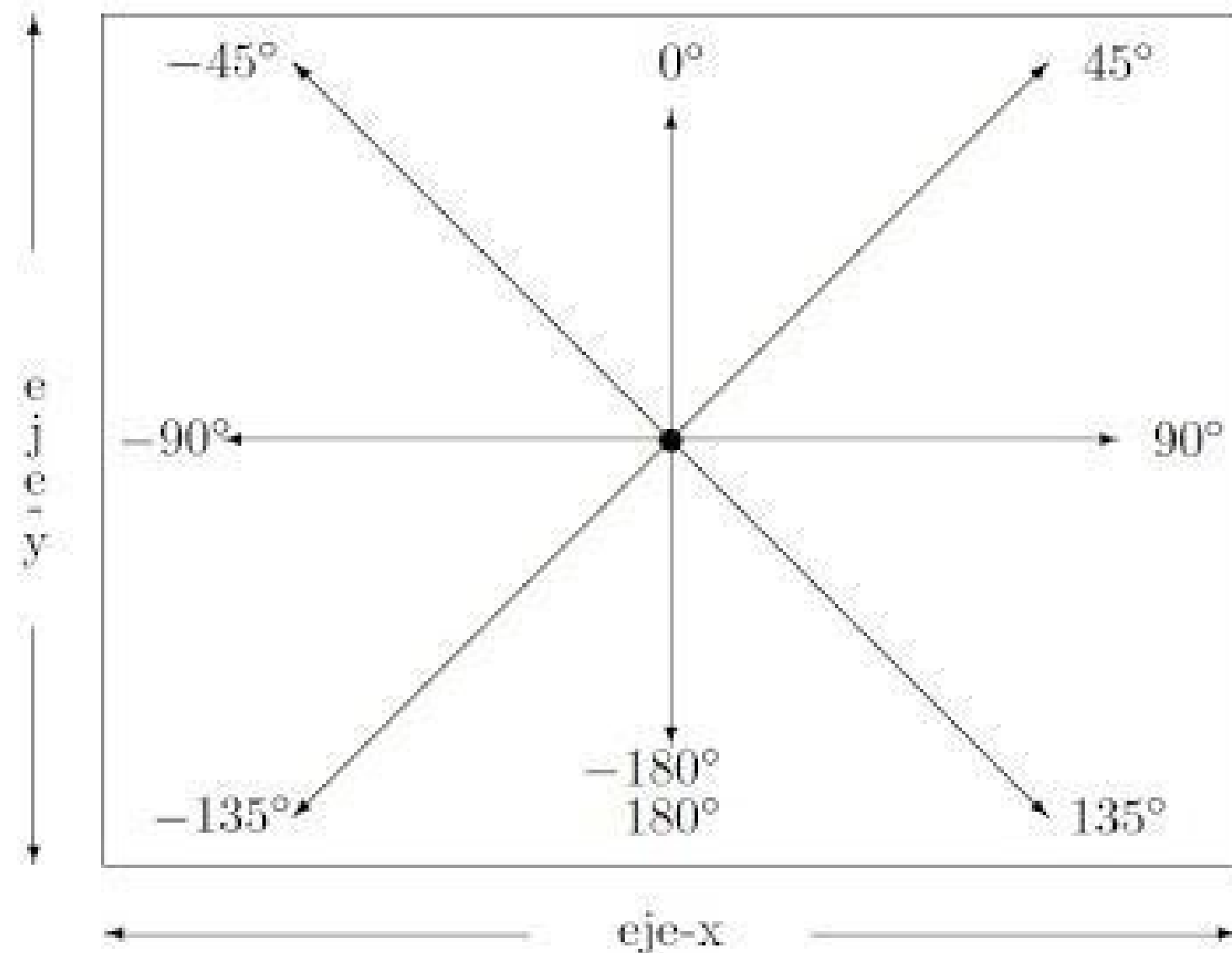
Figura 02: Página de programação do Scratch.



Fonte: Acervo do PIBID Matemática/UFCAT.

É possível trabalhar a Matemática no Scratch?

Figura 03: Ângulos na tela.



Fonte: Acervo do PIBID Matemática/UFCAT.

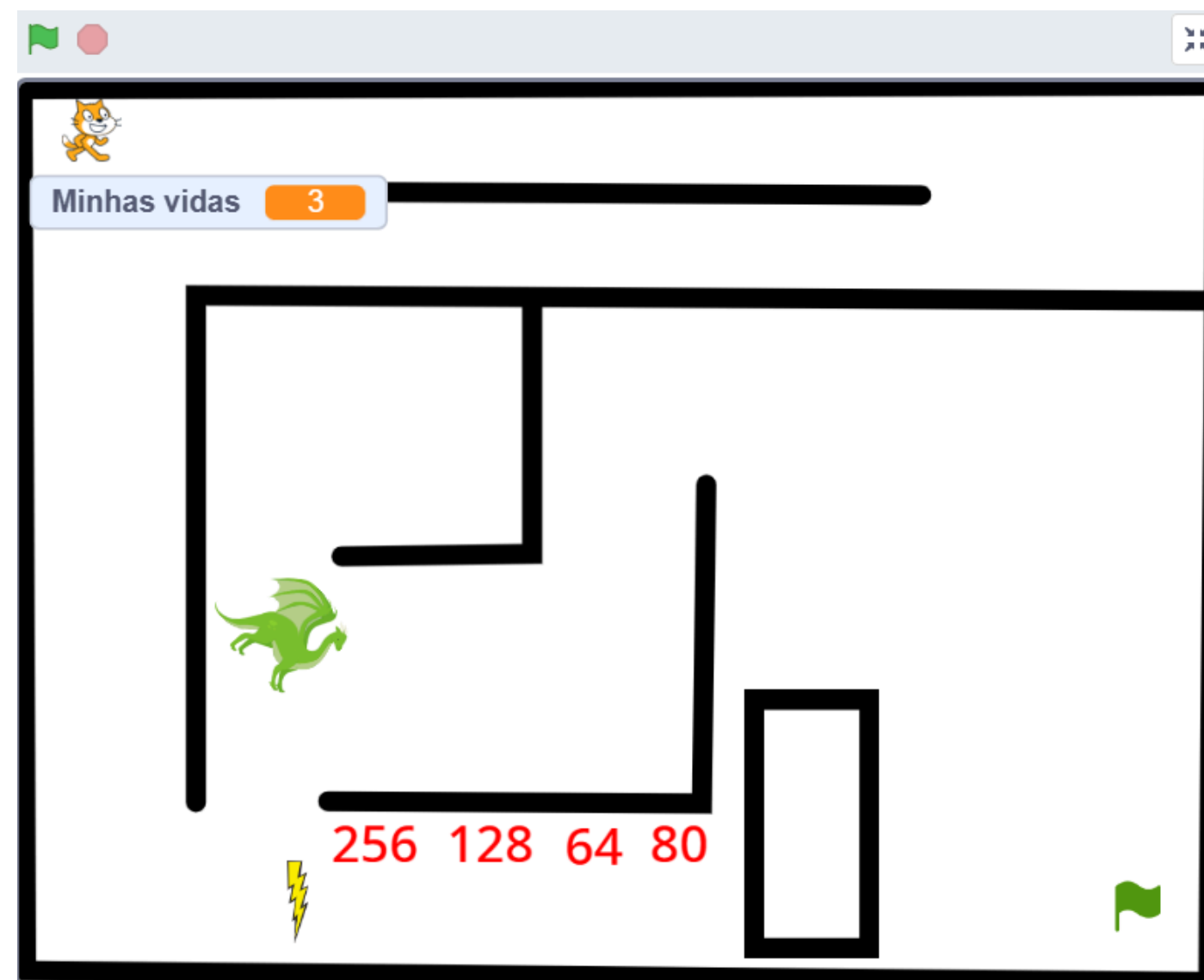
Figura 04: Plano cartesiano na tela.



Fonte: Acervo do PIBID Matemática/UFCAT.

- O jogo escolhido foi o do Labirinto, produzido a partir de uma proposta apresentada em uma disciplina da **Especialização EAD e GRATUITA em Robótica Educacional e suas Tecnologias no Ensino de Matemática da UFCAT** - <https://imtec.ufcat.edu.br/robotica-educacional>.

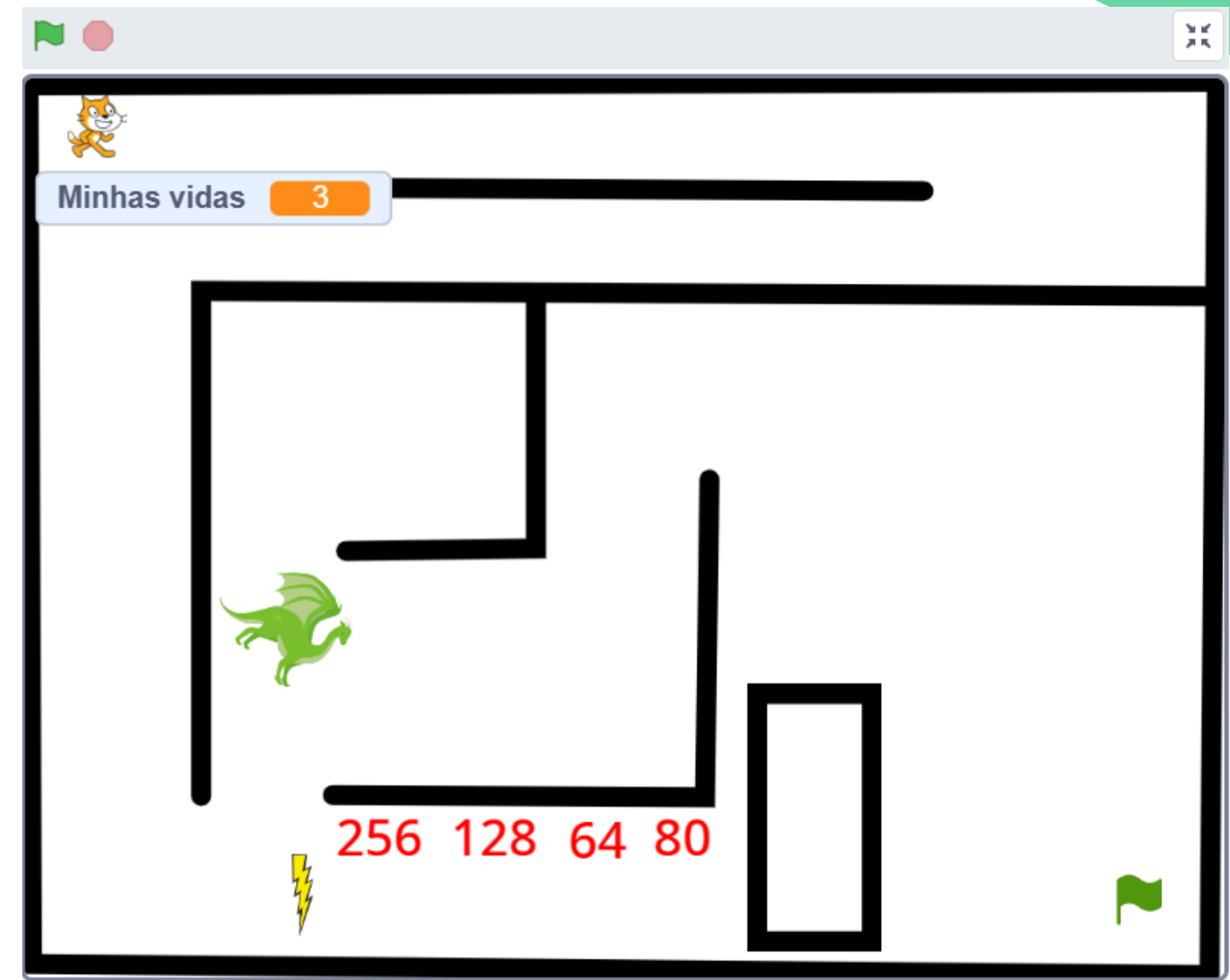
Figura 05: Jogo do labirinto.



Fonte: Acervo do PIBID Matemática/UFCAT.

- Os alunos tinham que programar o jogo do labirinto, com um personagem principal que se locomove na horizontal e na vertical, um monstro que se locomove na diagonal e um desafio matemático para liberar uma passagem. O objetivo é pegar a bandeira verde.
- A matemática não está apenas no desafio, a programação toda envolve conceitos matemáticos.

Figura 05: Jogo do labirinto.



Fonte: Acervo do PIBID Matemática/UFSCar.

Objetivos

- Muitos estudantes têm aversão a matemática, principalmente quando começa a exigir uma compreensão do abstrato.
- Uma alternativa é despertar o interesse dos estudantes na matemática com algo que já existe familiaridade.
- O método apresentado é aliar a matéria que está sendo ministrada com a tecnologia.
- Verificar se o emprego da tecnologia proporciona uma dinâmica de aprendizado recíproco no ambiente escolar.

Metodologia

- Para o desenvolvimento da atividade foram necessárias 04 aulas de 45 minutos.
- Iniciamos então a primeira aula apresentando o Scratch, o labirinto que seria desenvolvido e o manual que seria orientação para o jogo, que pode ser acessado em <https://drive.google.com/file/d/1634skw9KleUR17UDjA9137zuaheu1xie/view?usp=sharing>.

Nome: _____

Escola: _____

Série/Turma: _____ Data: ___/___/___

O Scratch é uma linguagem de programação criada em 2007 pelo Media Lab. do MIT, sendo disponível no formato online. De forma divertida e de fácil manuseio, já que a forma de linguagem de programação que utiliza são blocos com as palavras chaves para serem inseridas na programação, veja Figura 2. Podemos criar nele animações, jogos e outros programas interativos.

Conforme a Figura 1A, o Scratch ocorre em um plano cartesiano e o formato de ângulo nele é dado como na Figura 1B.

Figura 1A: O plano cartesiano no Scratch

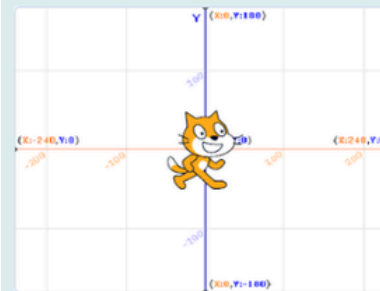
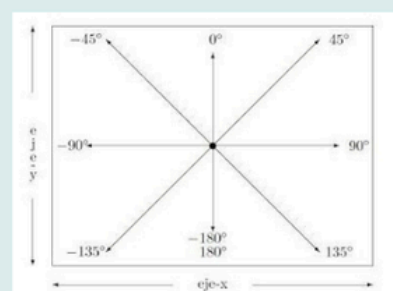


Figura 1B: O ângulos no Scratch



Fonte: Própria

A ideia desse MANUAL é desenvolver um jogo do labirinto, com uma abordagem matemática, baseado na ideia de Souza (2023). Veja a Figura 10.

Neste jogo, o gatinho precisa realizar deslocamentos pela horizontal e/ou vertical, dentro do labirinto, sendo vencedor o jogador que conseguir tocar a bandeira verde. Os desafios do jogo são: 1) não ultrapassar a linha preta; 2) caso toque no dragão, que se desloca em uma reta entre dois pontos e bloqueia parcialmente uma das passagens, perderá uma vida, das 3 vidas que são disponibilizadas no início do jogo; 3) caso deseje tocar no raio, um desafio matemático será apresentado e sua resposta correta, fará com que a porta em retângulo desloque na vertical permitindo a passagem, porém a sua resposta errada retirará uma vida do jogador, sendo somente uma oportunidade de realizar o desafio.

Desenvolvimento do jogo

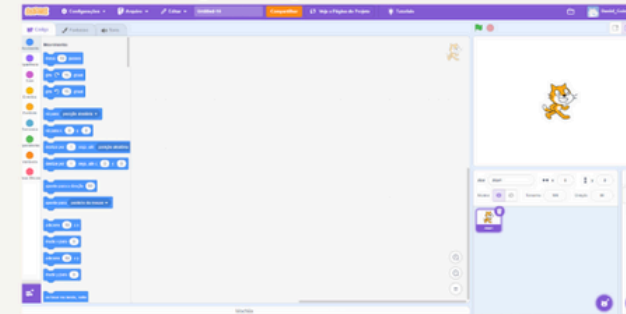
Aula 1

Figura 06: Manual parte 1

1º Passo: Acesse <https://scratch.mit.edu/>, realize o login e clique em "Criar" no menu superior.

2º Passo: No canto inferior direito, conforme a Figura 2, vá na opção "selecionar cenário" e escolha pintar, com isso faça um labirinto de sua escolha, ou siga o modelo como na Figura 10A. Sugestão, utilize a Linha e o Retângulo e um contorno de tamanho 16.

Figura 2: Pagina inicial do Scratch



Fonte: Própria

3º Passo: Você pode alterar o seu ator em "Selecionar um Ator" no link à direita inferior, ir na lupa e escolher um novo ator. Caso realize essa inserção, exclua o ator gatinho. Veja na parte de programação central que o gatinho aparece na parte superior, próximo as bandeirinhas verde e vermelha. Isso significa que esse ator esta escolhido para ser programado.

4º Passo: Selecionado o ator que deseja programar, precisa escolher os bloquinhos que estão no menu à esquerda do Scratch para realizar a programação desse ator, posicionando esse blocos na parte de programação no centro do programa. Lembre-se que para o ator ir para frente significa estar movimentando na horizontal, sendo necessário acrescentando em x um valor positivo, para trás na horizontal, adicionando um valor negativo para x, e em ambos os casos o y permanecerá constante. E os movimentos na vertical? Discuta com os seus colegas!

Na escolha dos blocos, comece com os bloquinhos:

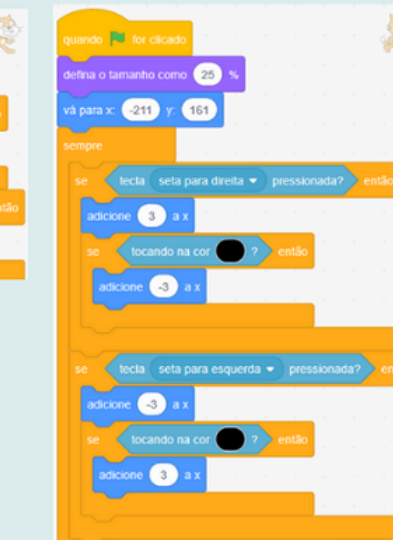
- "quando bandeira for clicada" - para iniciar o jogo ao clicar na bandeirinha verde;
- "sempre" - para que sempre os comandos dentro desse bloco sejam executados;
- "se <> então" - para ocorrer o movimento caso alguma tecla do teclado for tocada;
- "tecla <espaço> pressionada?" - para detectar a tecla sendo acionada;
- "adicione (10) a x" - para o deslocamento na horizontal. Veja na Figura 3A como ficou a nossa proposta para o deslocamento horizontal para a direita e para a esquerda. Repita o processo para os deslocamentos na vertical e clique na bandeira verde acima do personagem para o teste.

Importante verificar que os blocos com as mesmas características, são das mesmas cores.

Figura 3A: Programação movimento



Figura 3B: Programação movimento labirinto



Fonte: Própria

Aula 2

5º Passo: Sobre o tamanho do ator introduza o bloco "defina o tamanho como (100)%", antes do sempre e opte por valores entre 20% a 30%, tornando o proporcional ao tamanho do labirinto. Posicione o ator onde você deseje que ele reinicie o jogo caso perca, utilizando o bloco "vá para x: () y: ()" com as definições de x e y da posição selecionada, inserido antes do bloco "sempre", como na Figura 3B.

6º Passo: Para impedir o gatinho de atravessar todas as paredes, iremos inserir o bloco "se <> então" dentro do primeiro "se <seta para direita> então" e apos "adicionar (3) a x", e em <> adicionar o bloco "tocando na cor ()?", alterando a cor para preta. Neste caso, iremos inserir dentro da parte então deste bloco "se <tocando na cor ()? > então" um valor do movimento inversamente proporcional (caso for +3, por exemplo, o movimento para direita, ao tocar no preto pela direita, teremos o -3). Repita esse processo para todos os deslocamentos. Veja a Figura 3B para o movimento horizontal para a direita e para a esquerda.

7º Passo: Iremos inserir uma dificuldade para o nosso labirinto, sugerimos um monstro, em nosso caso o Dragão, que se desloca pela diagonal, veja Figura 10A, com o tamanho do Dragão de 20. Para a programação desse personagem, verifique se está selecionado e posicionado em um local que fechará a passagem, no nosso caso a posição utilizando os blocos "vá para x (-136) y (-37)". Inserindo os blocos como na Figura 4, podemos testar o Dragão ao clicar na bandeira verde. Discuta a função dos blocos.

Figura 07: Manual parte 2

Figura 4: Programação do monstro Dragão



Fonte: Própria

Aula 3

8º Passo: Para darmos continuidade é fundamental que os personagens interajam, a interação pensada é a de quando o personagem principal encosta no monstro, respectivamente aqui o gatinho e o Dragão, ele perca uma vida e volte para o início, e para além disso caso perca todas as vidas transmita uma mensagem de perdeu. Para isso, na programação do personagem principal é necessário criar o sistema de vida, e um ponto inicial, para determinar o ponto de início basta arrastar seu personagem principal até o local e anota sua coordenada x,y. Além disso, utilizaremos na programação a transmissão da mensagem "Perdeu", que foi digitada ao incluir o bloco "transmita (mensagem 1)".

Figura 5: Programação do personagem principal vidas



Fonte: Própria

9º passo: Inserimos um palco para ser apresentado caso a mensagem "perdeu" seja transmitida, mostrando para o usuário que o jogo acabou ao fim.

Além disso, o nosso personagem principal ficará do tamanho de 100% e todos os outros atores irão se esconder.

Figura 6A: Cenário para mensagem "Perdeu".



Figura 6B: Alterando cenário com a mensagem "Perdeu" e bandeira clicada



Figura 6C: Gatinho centro e tamanho 100 %

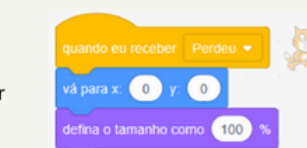


Figura 6D: Dragão esconde ao receber "Perdeu"



Figura 6E: Retângulo esconde e mostra



Fonte: Própria

Aula 4

10º passo: Vamos colocar um desafio matemático? Vai em "Selecionar um ator" e lá adicione o raio. Além disso, em "Selecionar um ator", vá em "Pintar" e construa um retângulo e depois toda a sequência de números conforme a Figura 10, colocando em um tamanho ideal. Para a sequência de números utilize a opção "Texto", em "Pintar". Queremos que quando o gatinho tocar o raio apareça um desafio matemático, esse passo é análogo com o passo anterior, precisando adicionar mais um bloco, "diga () por () segundos", com a seguinte pergunta "Vá até o 8º termo da sequência (1, 2, 4, 8, 16, ...) para que a porta retangular se abra!". Ocorrendo o acerto da resposta, será transmitida a mensagem "Sair Retângulo", com o erro da resposta será transmitida a mensagem "Errou", veja Figura 7A. Quando receber a mensagem "Errou", o nosso ator perderá vida e pode até perder o jogo (Figura 7B). Quando acertar, a porta retângulo se deslocará na vertical, veja Figura 7C.

Figura 7A: Ator tocando nos personagens

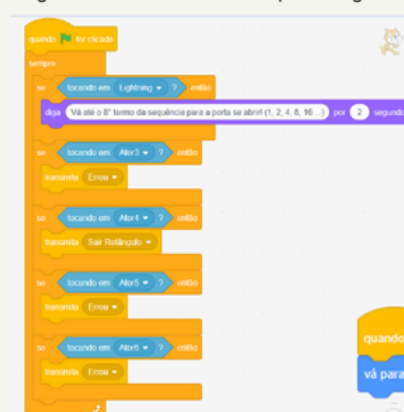
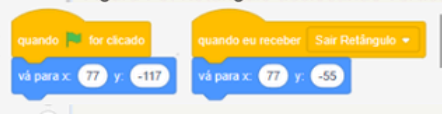


Figura 7B: Recebendo a mensagem Errou



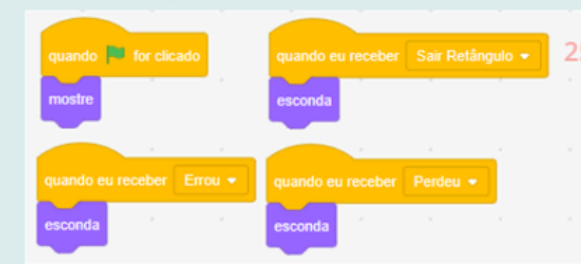
Figura 7C: Retângulo deslocando vertical



Fonte: Própria

Lembrando que queremos ter somente uma tentativa para a participação do desafio matemático. Repita a Figura 8 para todos os números e para o personagem raio.

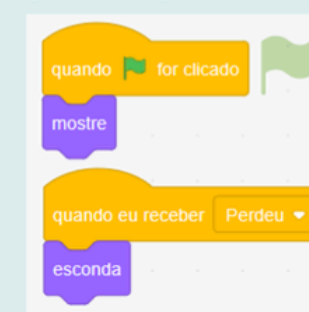
Figura 8: Programação dos números



Fonte: Própria

11º passo: Estamos prontos para inserir o último personagem do nosso jogo, que será uma bandeirinha de chegada. Após a inserção desse personagem, solicite que mostre quando o jogo for iniciado, que esconda quando receber a mensagem "Perdeu" (Figura 9A), e após ser tocada pelo gatinho, na programação do gatinho, inseriremos que o jogador receberá a mensagem de vencedor (Figura 9B).

Figura 9A: Configuração bandeirinha Figura 9B: Programação ganhou do gatinho



Fonte: Própria



Figura 10A: Proposta final de jogo

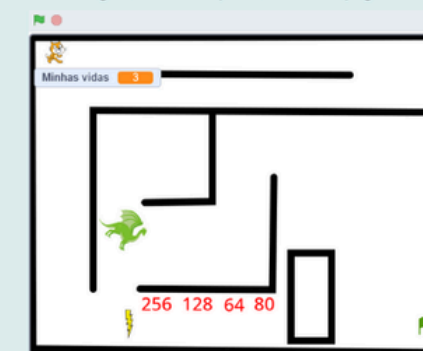
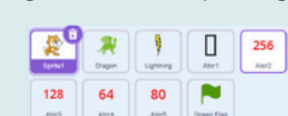


Figura 10B: Todos os personagens



Fonte: Própria

REFERÊNCIAS:

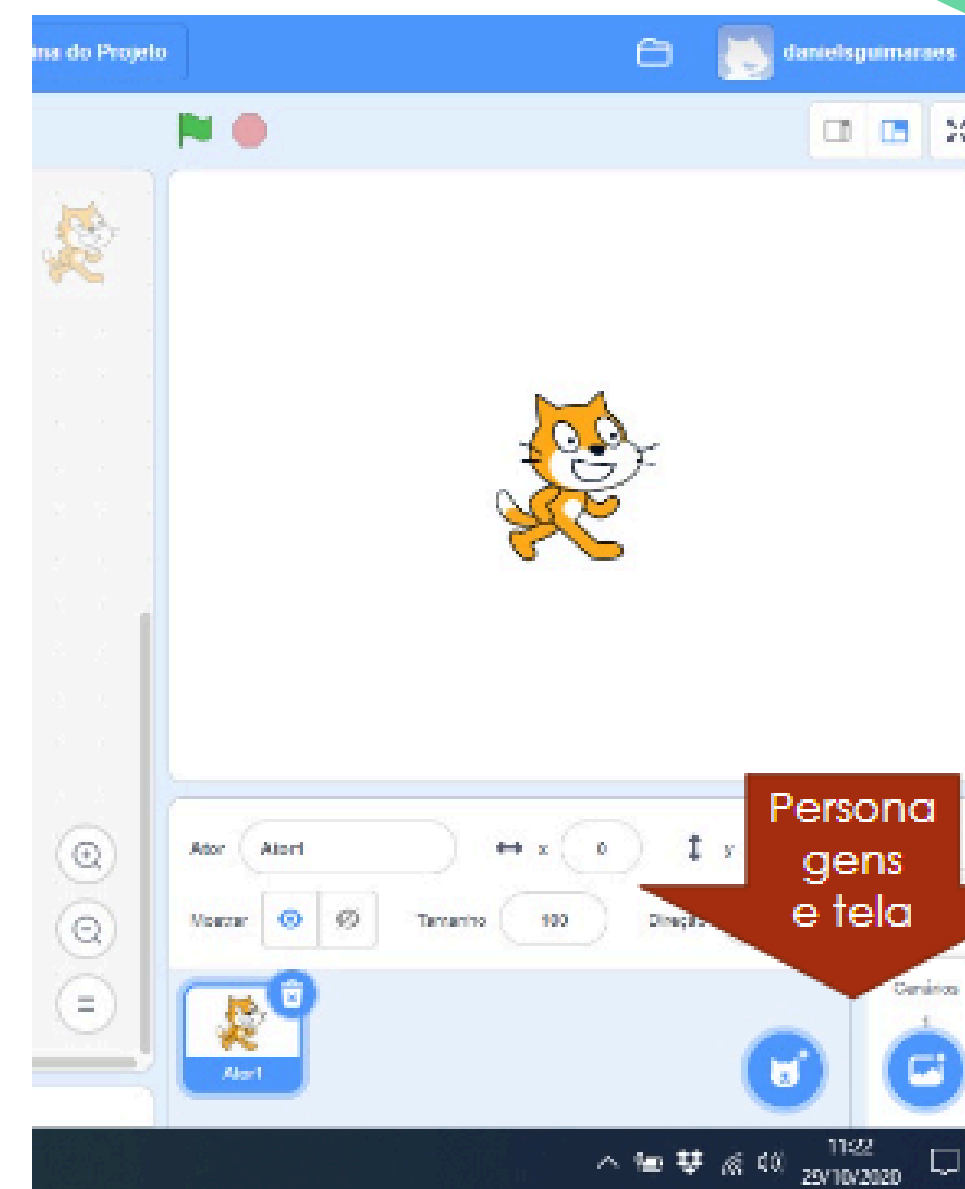
Sobre o Scratch. Disponível em <https://scratch.mit.edu/about>. Acesso em 5 mai. 2024.

SOUZA, Brenner. Aula 05 - 06 - Labirinto. 2023. Disponível em <https://scratch.mit.edu/projects/854331110>. Acesso em 11 mai. 2024.

Ainda na primeira aula, utilizando os dois itens personagens e tela, conforme a Figura 08, os alunos fizeram:

- O desenho do Labirinto, podendo variar conforme a escolha, porém podendo seguir o nosso exemplo.
- A troca do autor principal, que o nosso caso é o gatinho.

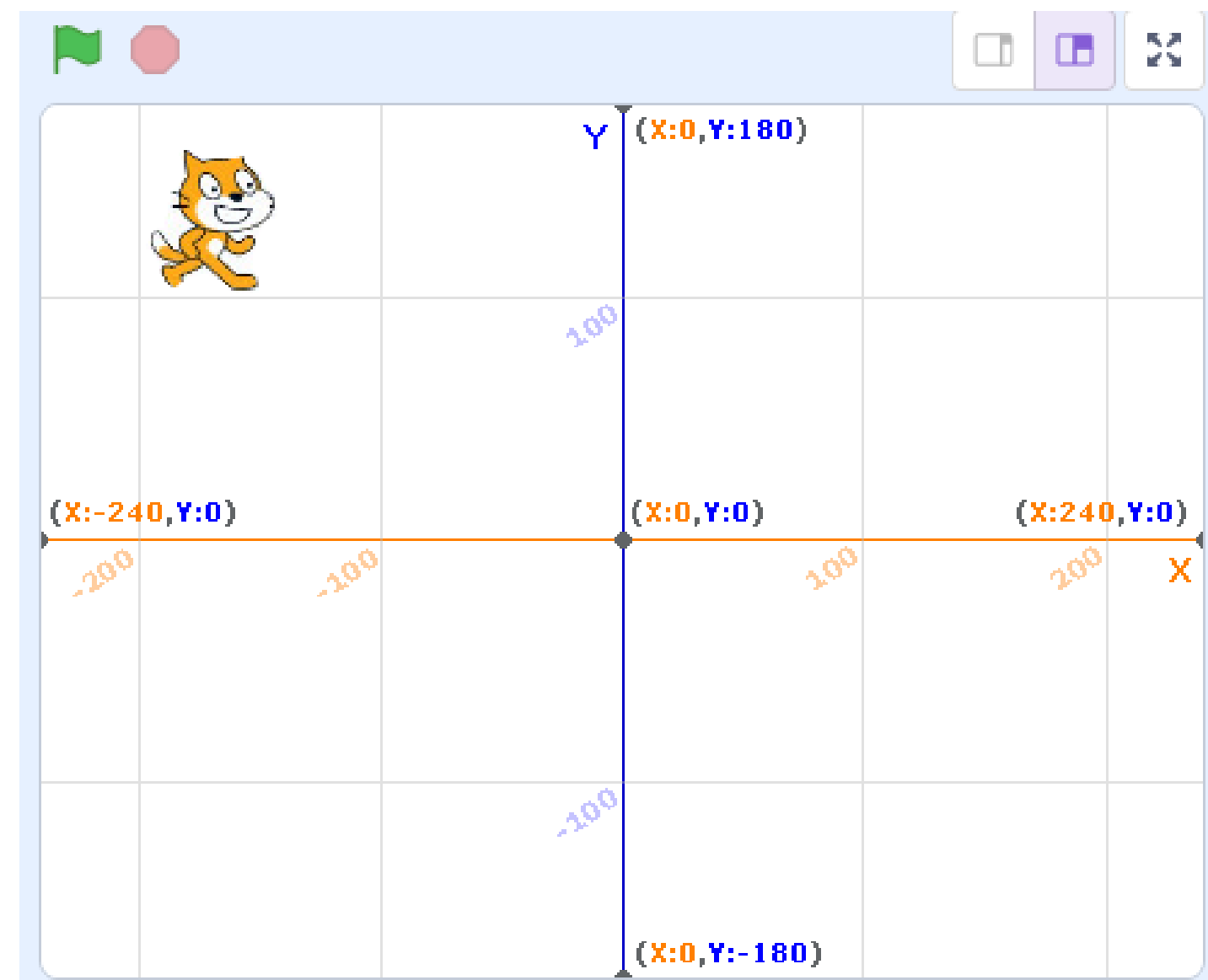
Figura 08: Itens personagens e tela.



Fonte: Acervo do PIBID Matemática/UFSCar.

- Começamos a programação com o gatinho, para deslocar na horizontal. Após a explicação dos blocos, fizemos o seguinte debate Matemático: Como locomover o gatinho para frente e para trás (horizontal).

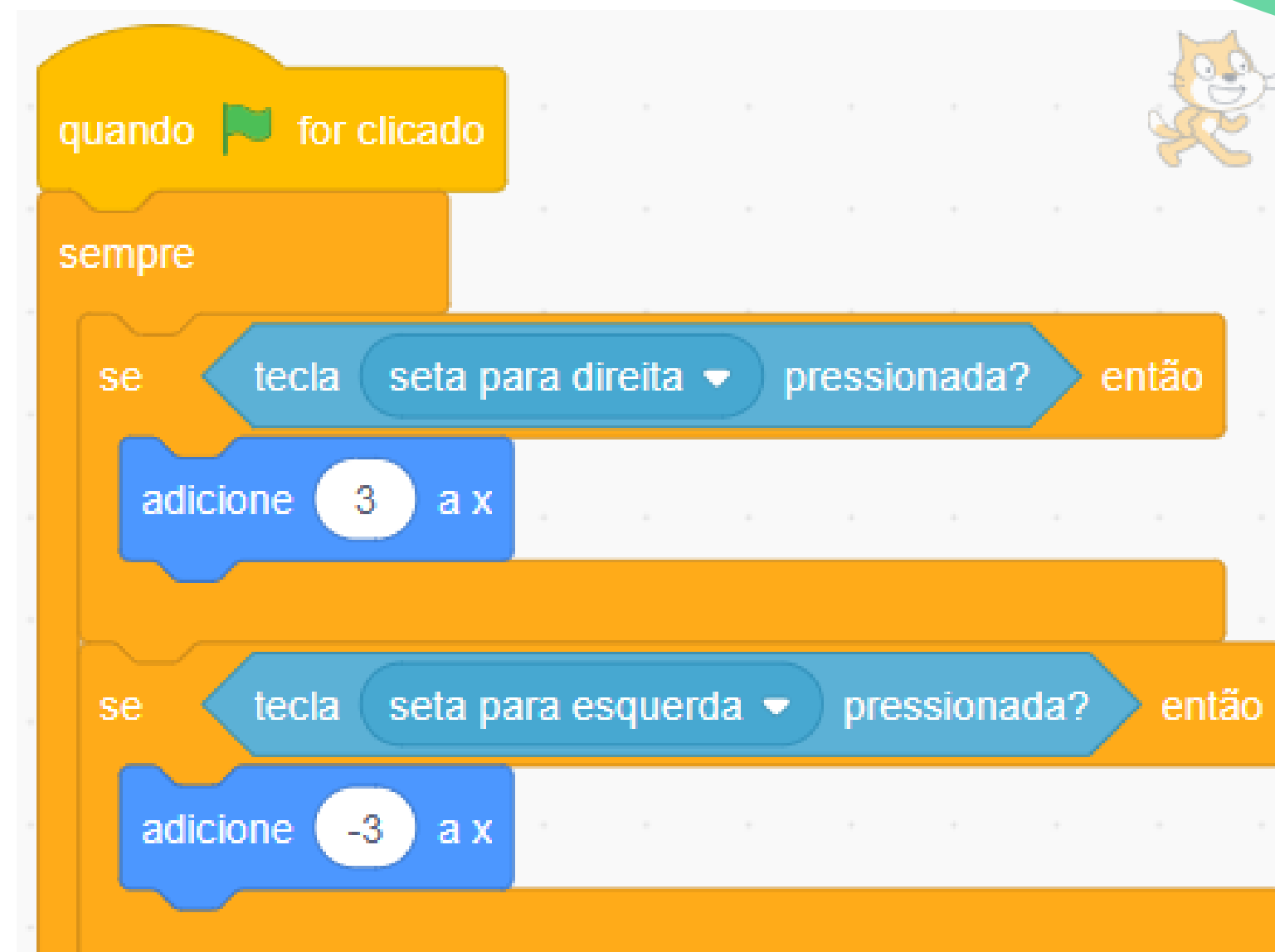
Figura 09: Plano cartesiano na tela.



Fonte: Acervo do PIBID Matemática/UFCAT.

- Para o gatinho se locomover para frente, é preciso que ele se move para direita, então o valor de x é positivo, e para se mover para trás é o análogo.
- Então para o gatinho se locomover na horizontal, é preciso escolher o bloco "adicione () a x ", o valor de x determina também a velocidade.

Figura 10: Programação para locomover na horizontal.



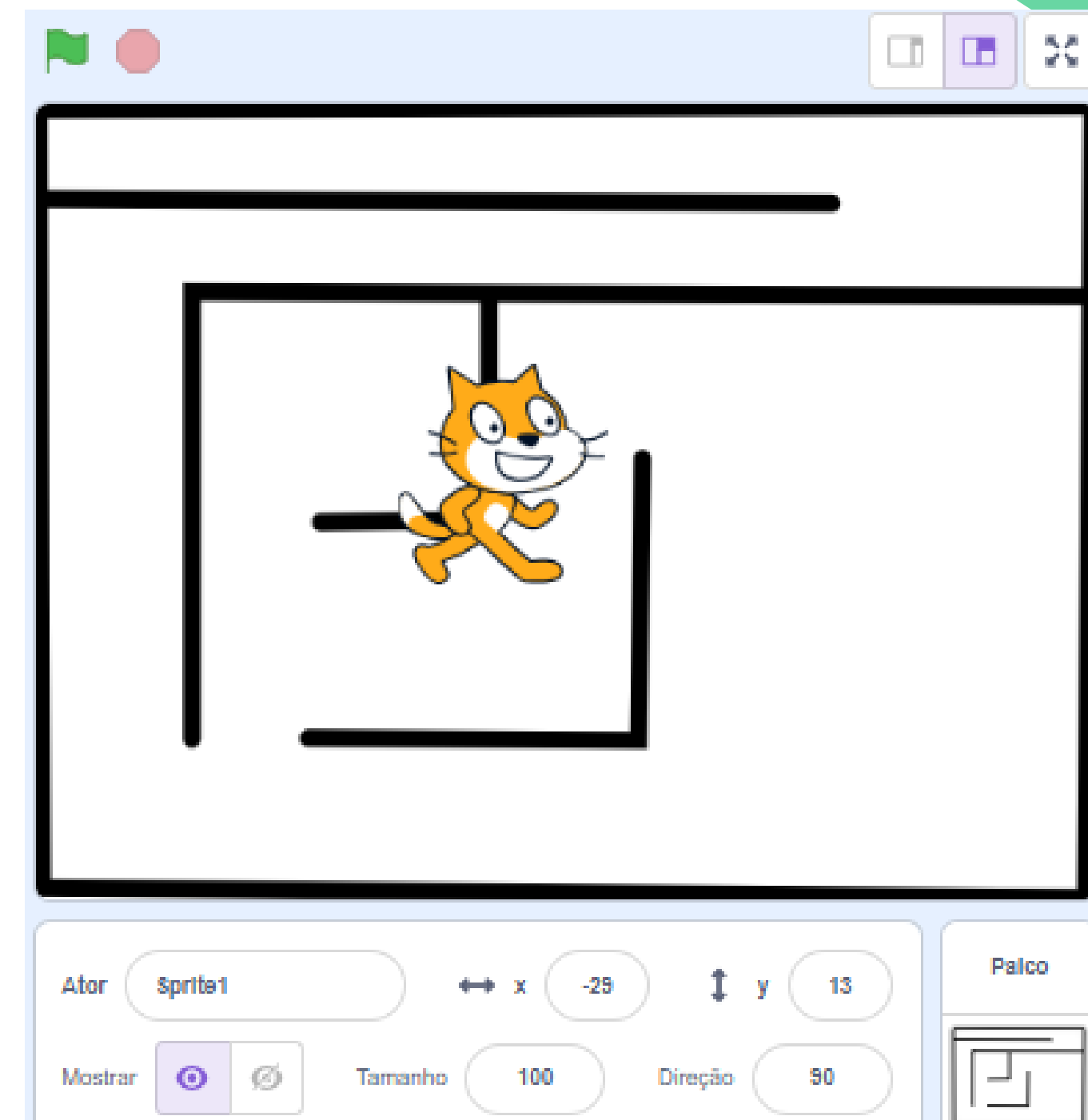
Fonte: Acervo do PIBID Matemática/UFCAT.

Na segunda aula, iniciamos perguntando quais problemas estavam ainda ocorrendo.

As respostas foram:

- o personagem principal está grande;
- o personagem principal está deslocado sobre a linha preta.

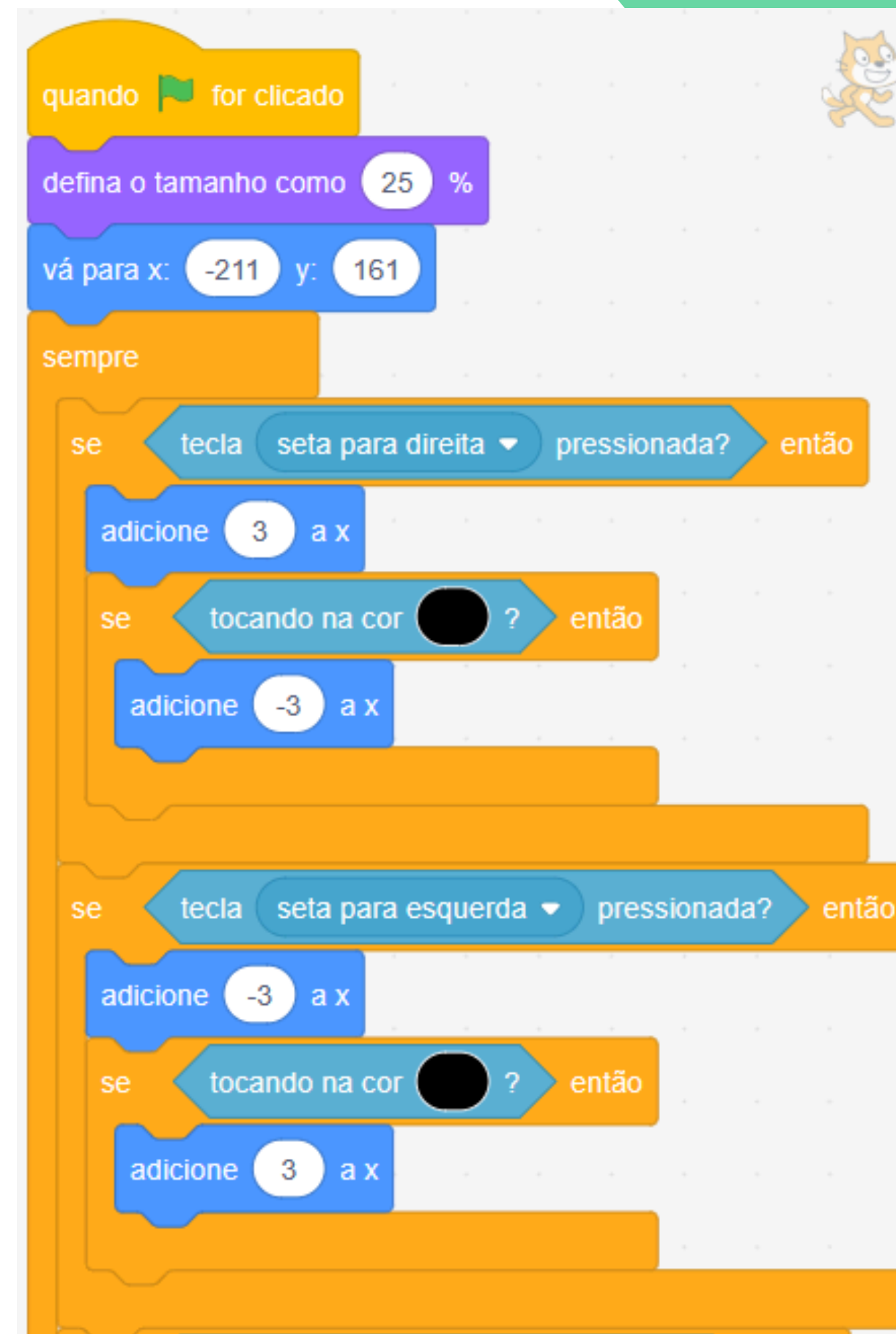
Figura 11: Problemas encontrados.



Fonte: Acervo do PIBID Matemática/UFCAT.

- Para impedir o gatinho de atravessar todas as paredes, iremos inserir os dois blocos “se <tocando na cor ()?> então”, colocamos um valor do deslocamento inversamente proporcional (caso for +3, por exemplo, o movimento para direita, ao tocar no preto pela direita, teremos o -3).
- **Após a finalização para o deslocamento na horizontal, deixamos os estudantes debaterem sobre o caso da vertical.**

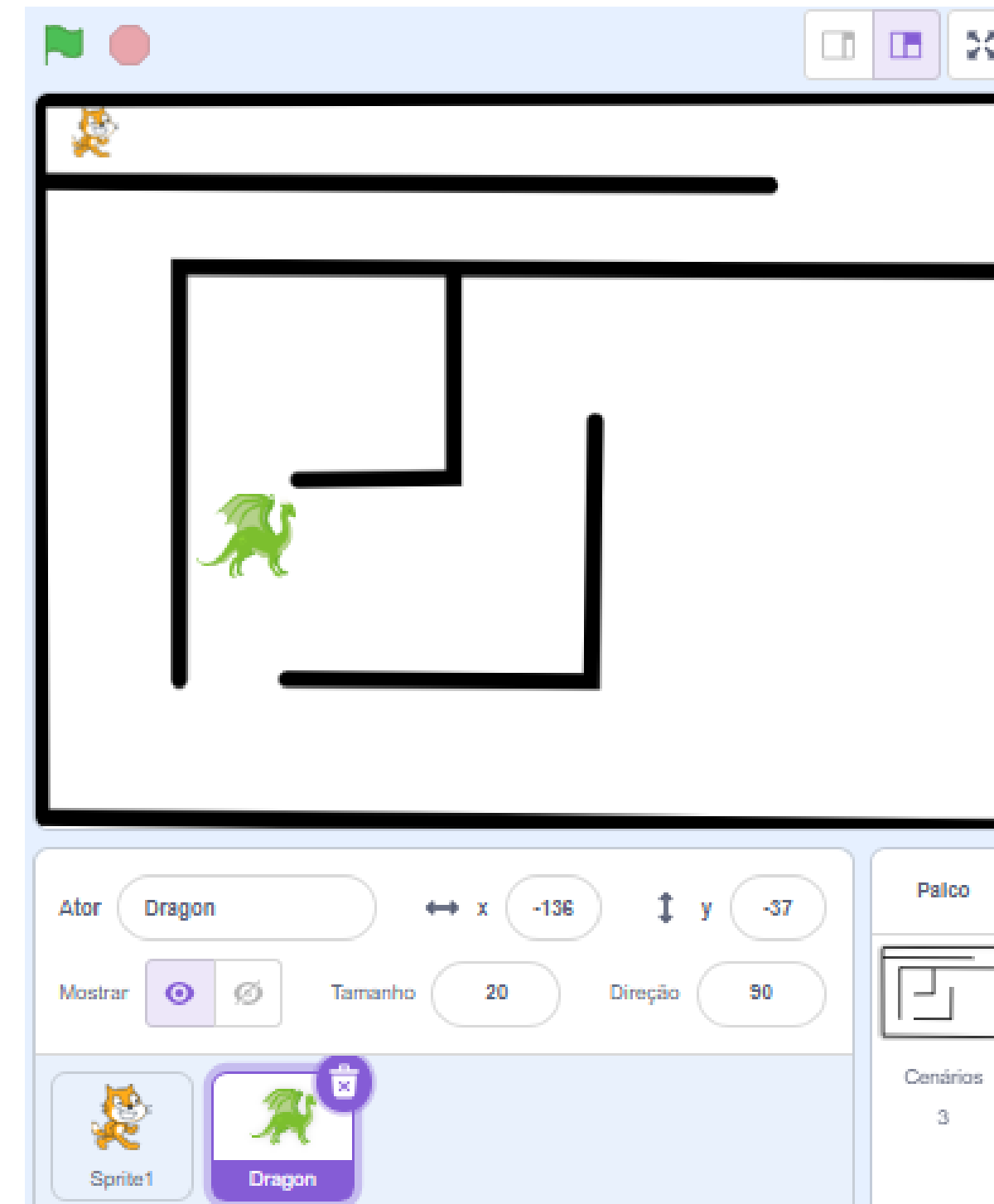
Figura 12: Programação para impedir ultrapassar a linha preta.



Fonte: Acervo do PIBID Matemática/UFSCar.

- Iremos inserir uma dificuldade para o labirinto, um monstro, que se desloca pela diagonal.
- Posicione o dragão de maneira que fecha a passagem. No nosso caso utilizamos o bloco "vá para x () y ()" com os valores x -136 e y -37

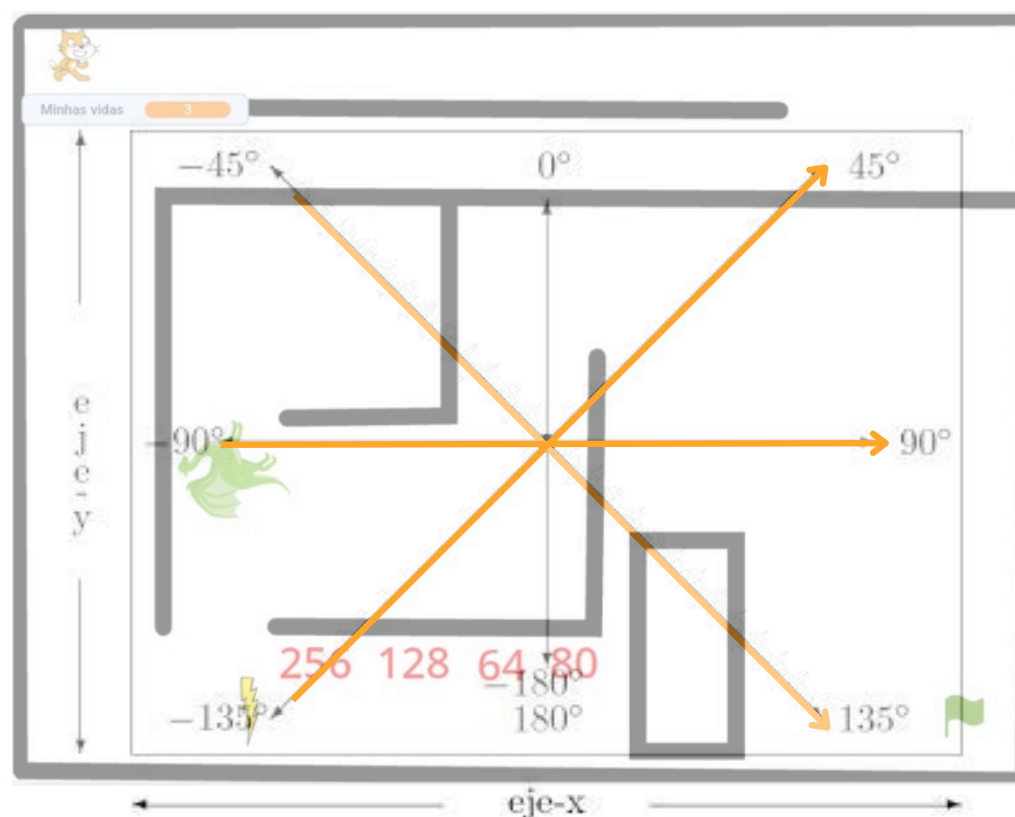
Figura 13: Monstro que se desloca pela diagonal.



Fonte: Acervo do PIBID Matemática/UFCAT.

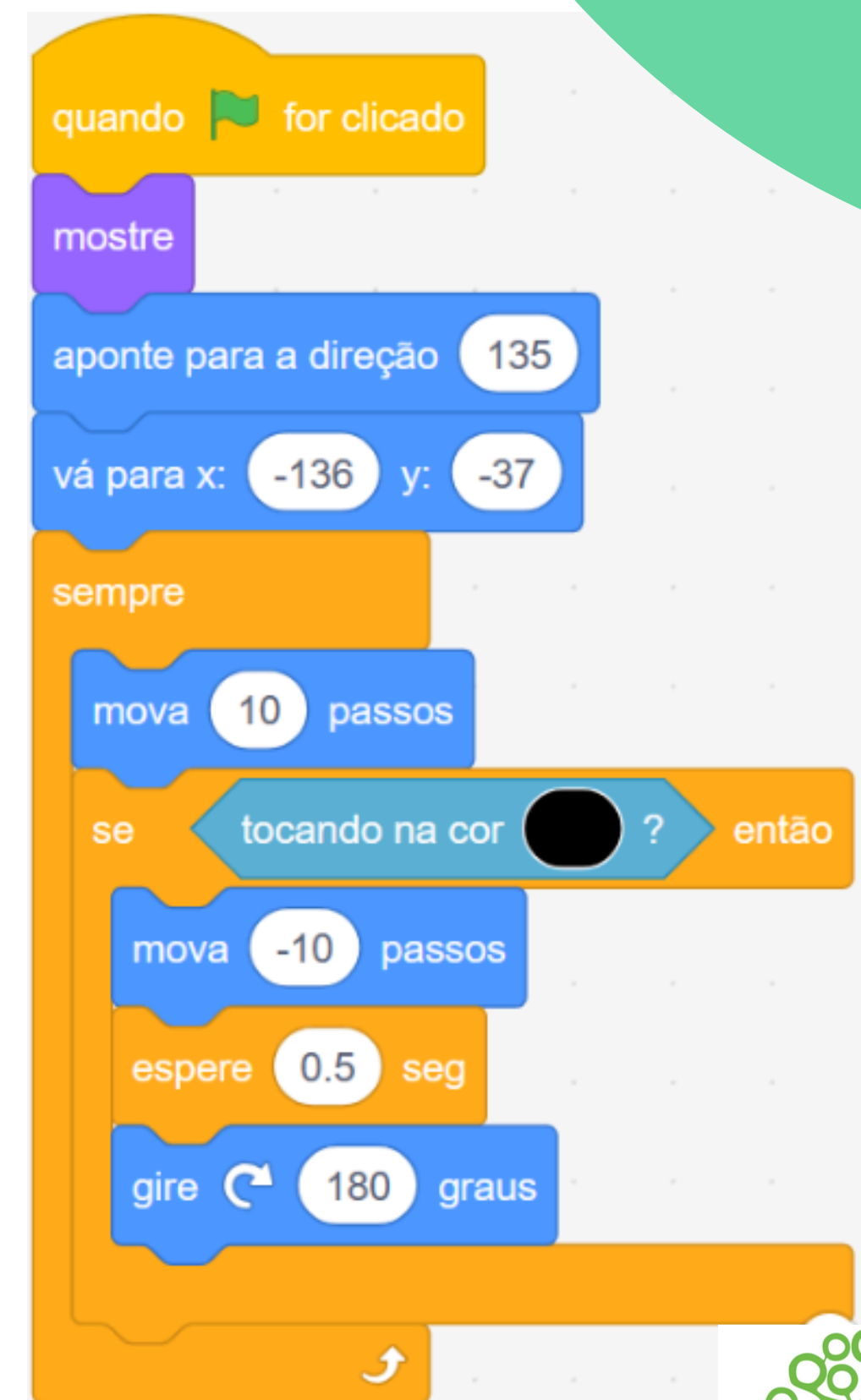
- Colocamos o bloco "aponte para a direção (135)"
- Dentro do sempre, queremos que o Dragão desloque 10 passos, porém caso toque a linha preta pare o movimento.
- Após esse toque na linha preta, gira 180 e volta no sempre, realizando novamente os deslocamentos .

Figura 14: Monstro apontando para a direção 135°.



Fonte: Acervo do PIBID Matemática/UFCAT.

Figura 15: Programação da locomoção do monstro.



Fonte: Acervo do PIBID Matemática/UFCAT.

- Na terceira aula, dermos prosseguimento a programação, como:

1. Fazer com que os personagens se interagem. Então quando o personagem encosta no monstro, respectivamente aqui o gatinho e o dragão, ele perca uma vida e volte no início.
2. Para isso, na programação do personagem principal é necessário criar o sistema de vida, e um ponto inicial, para determinar o ponto de início basta arrastar seu personagem principal até o local e anota sua coordenada x,y.
3. Além disso, utilizaremos na programação a transmissão da mensagem “Perdeu”, que foi digitada ao incluir o bloco “transmita (mensagem 1)”.

Figura 16: Programação das vidas do gatinho.

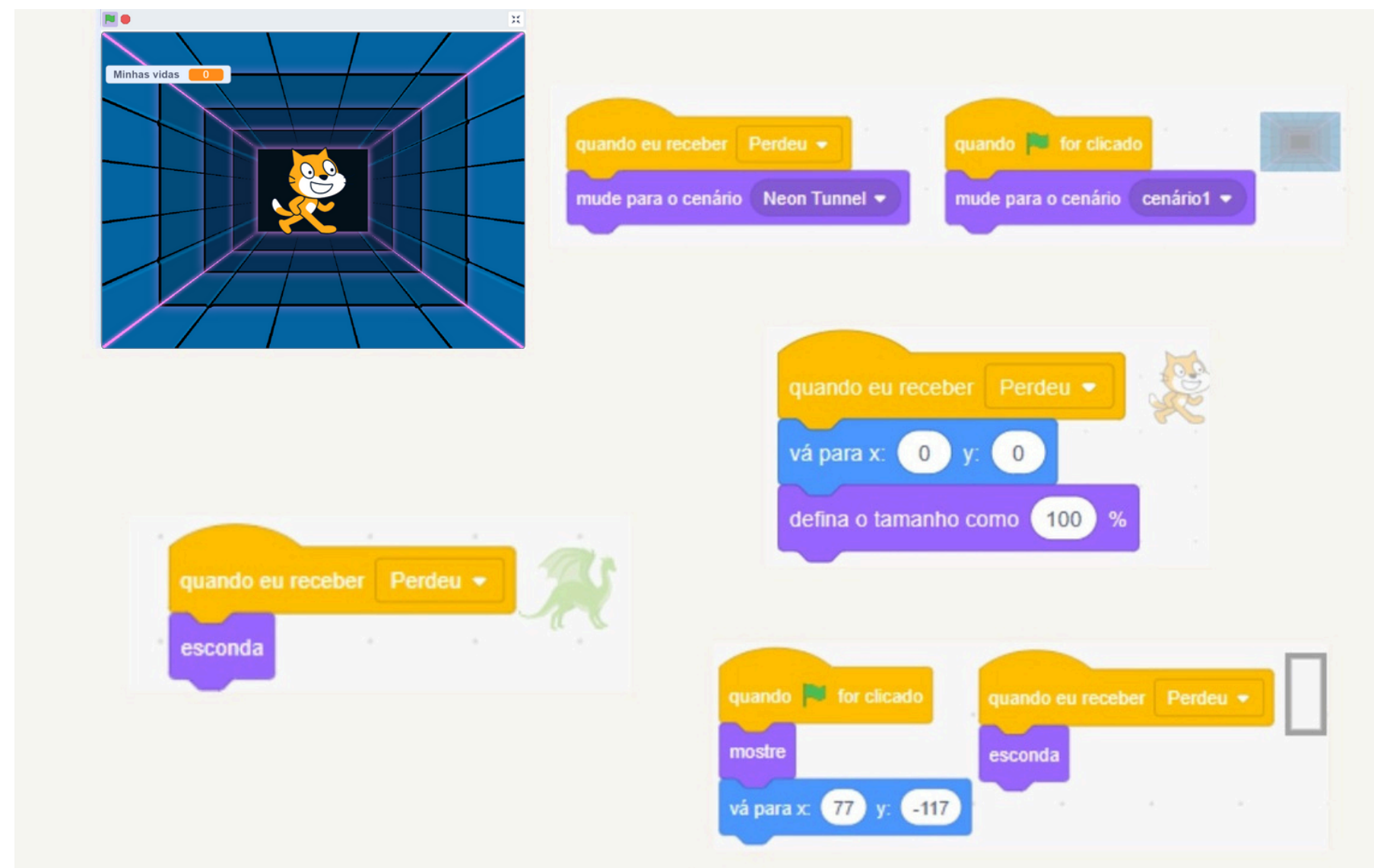


Fonte: Acervo do PIBID Matemática/UFCAT.

4. Inserimos um palco para ser apresentado caso a mensagem “perdeu” seja transmitida, mostrando para o usuário que o jogo chegou ao fim.

5. Além disso, o nosso personagem principal ficará do tamanho de 100% e todos os outros atores irão se esconder.

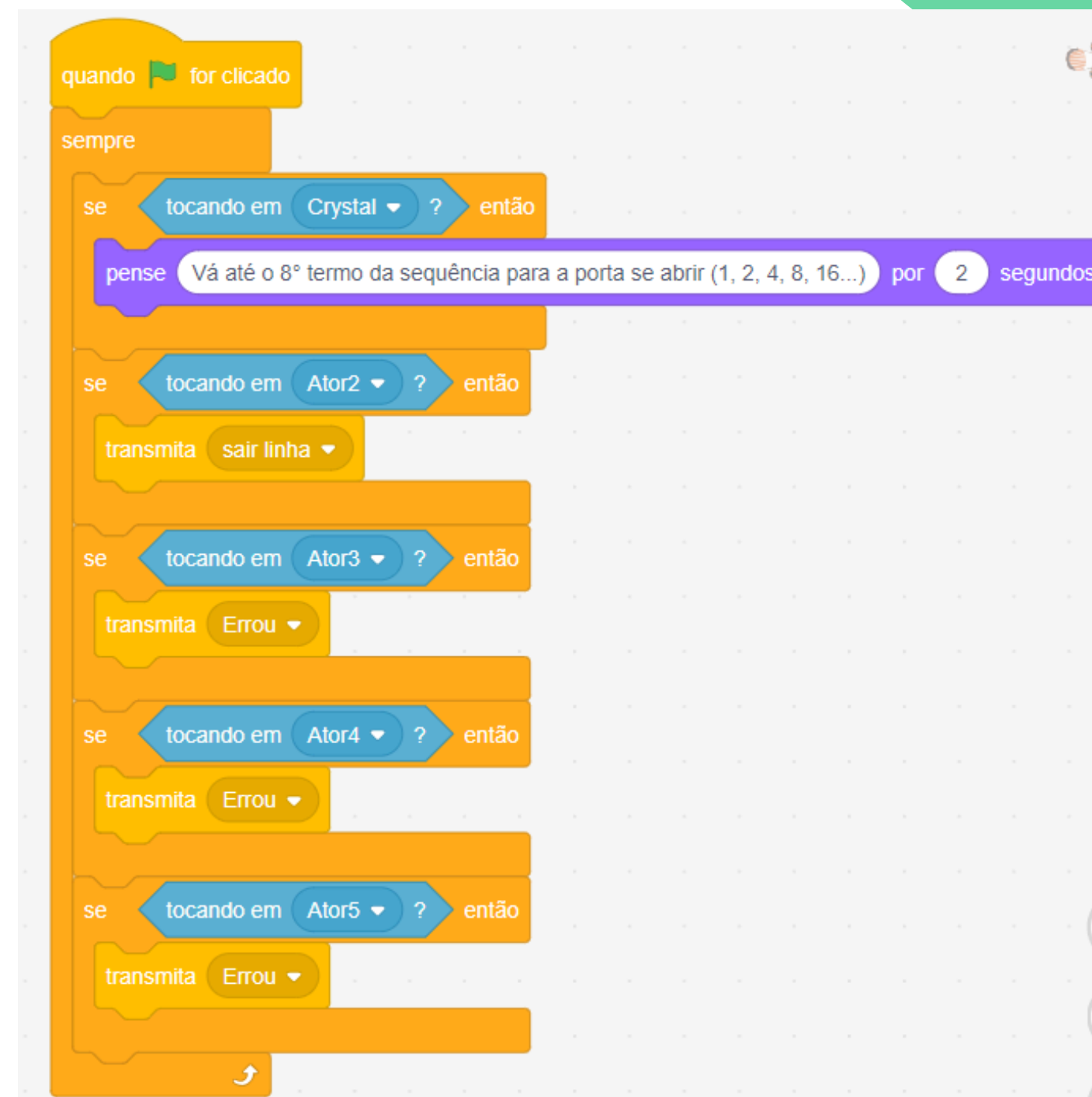
Figura 17: Palco e programações para quando perder.



Fonte: Acervo do PIBID Matemática/UFCAT.

- Agora já na quarta aula, solicitamos aos estudantes pensassem em um desafio matemático. Que se caso o jogador acerte, abrirá um caminho, caso o jogador erre, ele perderá uma vida.
- Para finalizar a programação falta apenas colocar um bandeira no final do labirinto.
- O objetivo do jogo é deixar o monstro fechando toda a passagem, forçando o jogador responder o desafio, para chegar o fim do labirinto e pegar a bandeira.

Figura 18: Programação com o desafio matemático.

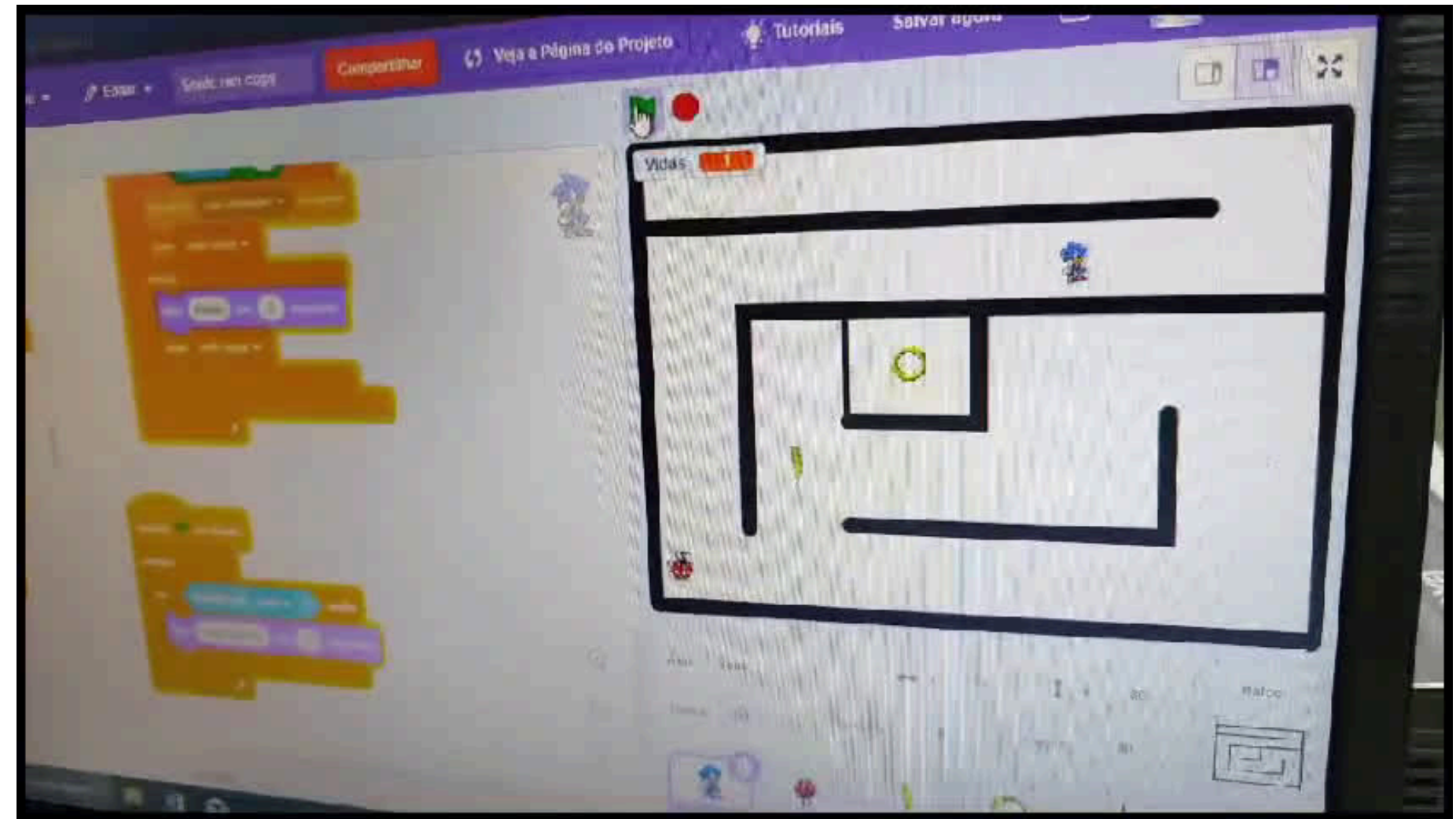


Fonte: Acervo do PIBID Matemática/UFCAT.

Resultados

- Essa abordagem se mostrou eficaz, pois ficou nítida a felicidade dos estudantes em suas descobertas dentro do Scratch, além de perceber que a todo o momento eles queriam testar as suas ideias e inserir suas criatividade.

Vídeo 01: Criatividade do estudante.



Fonte: Acervo do PIBID Matemática/UFSCar.

Conclusão

- A decisão de integrar jogos digitais com a Matemática teve um impacto positivo em todos participantes.
- Notamos um alto nível de envolvimento por parte dos alunos da turmas envolvidas na criação dos jogos.
- Assim como na resolução das questões matemáticas essenciais para o desenvolvimento do jogo.
- Além disso, essa abordagem proporcionou uma experiência enriquecedora para os alunos do PIBID.

Referências

- Equipe Pibid. Manual do jogo do Labirinto, 2023. Disponível em <https://drive.google.com/file/d/1634skw9KleUR17UDjA9137zuaheu1xie/view?usp=sharing>. Acesso em 13 mai. 2024.
- Johnson, A., & Smith, B. (2021). Enhancing Mathematical Learning through Game-based Approaches. *Journal of Educational Technology*, 15(3), 78-95.
- Rodriguez, C., & Hernandez, D. (2022). Exploring Programming in Mathematics Education: A Case Study on Student Engagement and Learning Outcomes. *International Journal of STEM Education*, 8(4), 15-30.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.