



Jogo da cobrinha - I

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Apoio Técnico

Equipe UFMS

2025

SUMÁRIO

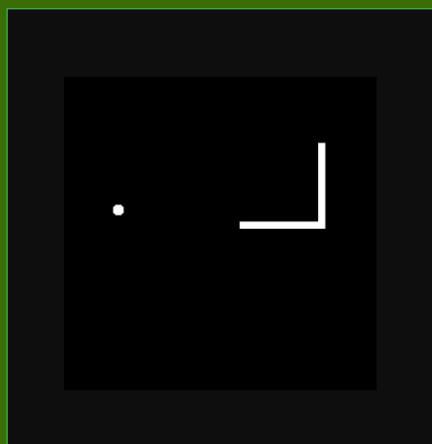
Introdução	2
Objetivos	3
Roteiro da aula	4
1. Contextualização	4
2. Programação	6
3. Feedback e finalização	13
Referências bibliográficas	13

Introdução

No mundo dos games, a categoria Arcade desperta emoções e lembranças aos mais variados jogadores, sendo uma referência para a história dos videogames.

Neste nosso último projeto do Módulo 4 de Robótica Primeiros Passos, vamos mergulhar nesse mundo para criar um dos jogos mais icônicos: o **jogo da cobrinha** (em inglês, *Snake*) com carinho da Robótica Paraná.

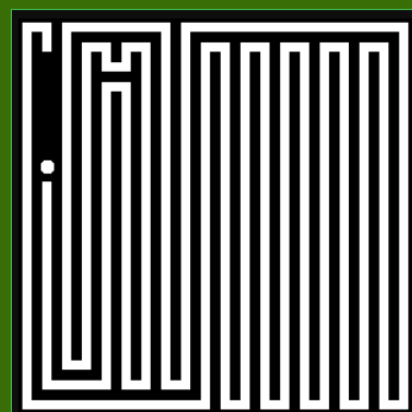
Figura 1 – Jogo da cobrinha



Fonte: Wikimedia Commons.



[Confira o gif animado de um jogo da cobrinha perfeito!!](#)



Simbolizando nosso percurso pela Robótica Primeiros Passos, no qual vamos incrementando experiência e conhecimento, propomos o jogo da cobrinha, um clássico arcade, para representar os incrementos em referência à história dos jogos.

Os jogos de arcade, ou fliperamas, não são apenas uma relíquia nostálgica: são o berço da indústria de videogames em um laboratório para a ciência da computação.

A história dos arcades começa com máquinas eletromecânicas nos anos 30. No entanto, a revolução digital começa, de fato, no início dos anos 70, com jogos como Computer Space e Pong (lembra que temos uma versão deste? Está nas **Aulas 4 e 5 - Criando jogos no mBlock - Pong [Partes I e II]**, do Módulo 2 de Robótica Primeiros Passos.)



Objetivos desta aula

- Conhecer o conceito de jogos arcade.
- Desenvolver a interface, pelo mBlock, do jogo da cobrinha.
- Programar o jogo da cobrinha versão Robótica Paraná.

Lista de materiais

- Notebook
- mBlock.

Roteiro da aula

1. Contextualização

Alguns elementos, no decorrer da história e da cultura, se tornam “clássicos”.

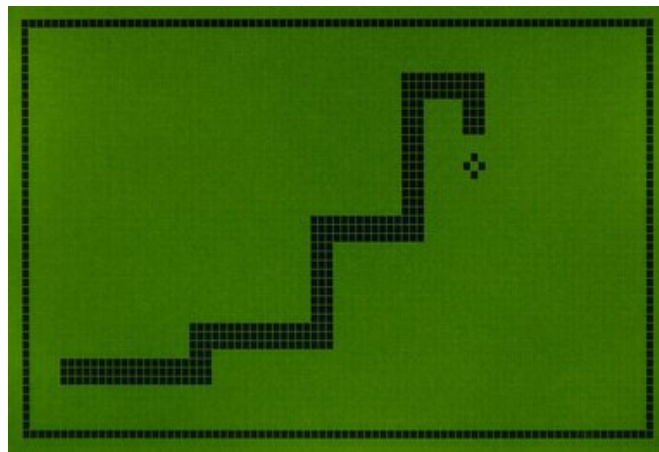
Os jogos eletrônicos nasceram muito antes dos celulares e computadores modernos. Na década de 1950, quando cientistas criaram pequenos experimentos em grandes computadores, apenas por curiosidade e diversão, surgem os primeiros clássicos: Tennis for Two (1958), jogado em um osciloscópio, seguido por Spacewar! (1962).

Nos anos 1970, os videogames começaram a chegar às casas das pessoas. O jogo Pong (1972), inspirado no tênis de mesa, foi um sucesso e mostrou que era possível transformar ideias simples em experiências divertidas.

Na década de 1980, os arcades e consoles domésticos se popularizaram com jogos clássicos como Pac-Man, Donkey Kong e Tetris. Todos tinham algo em comum: gráficos simples, mas uma lógica de programação muito bem pensada.

É nesse contexto que surge o Jogo da Cobrinha (Snake), criado originalmente em 1976, e que ficou mundialmente conhecido nos celulares Nokia nos anos 1990 e 2000. Apesar da simplicidade, o jogo utilizava conceitos importantes de coordenadas, variáveis e condições – os mesmos que exploraremos no mBlock.

Figura 2 - Interface para Nokia 6110



Fonte: Wikimedia Commons.

O que torna os jogos de arcade tão relevantes para além do entretenimento é sua eficiência. Os programadores da época tinham recursos de hardwares mais limitados e precisavam criar experiências envolventes usando pouquíssima memória e poder de processamento. Isso contribuiu para a ciência da computação devido ao desenvolvimento de lógicas de programação limpas e otimizadas.

Assim como os primeiros desenvolvedores, hoje vamos nos colocar no papel de criadores de jogos, entendendo que cada movimento da cobrinha, cada colisão e cada ponto é resultado de decisões lógicas que eles vão programar.



Durante a aula, iremos:

- Configurar o ambiente no mBlock e criar um novo projeto.
- Adicionar os sprites (cobra, comida e cenário).
- Criar variáveis para armazenar as posições da cabeça e do corpo da cobra.
- Programar os controles da direção usando as setas do teclado.
- Implementar o movimento contínuo e o crescimento da cobra ao comer.
- Adicionar a comida do jogo, programando sua posição aleatória após ser comida.
- Detectar colisões com as bordas e com o próprio corpo.
- Personalizar o fundo e as cores, ajustando brilho e matiz no menu Preencher.

A ideia é mostrar que, por trás da diversão, existe matemática, pensamento computacional e criatividade – pilares fundamentais da programação. Vamos lá?

2. Programação

Esta aula, além de ser muito divertida, é também relativamente simples. Não utilizaremos circuitos nem montagem de protótipos – todo o trabalho será feito diretamente no mBlock, tornando o processo mais rápido e fácil de testar.

Poderemos alterar cores, formatos e movimentos livremente, explorando a criatividade. Para esta atividade, utilizaremos apenas as abas **<Atores>** e **<Fundo>** do ambiente do mBlock.

O primeiro passo para iniciar a programação é escolher quem será sua cobrinha e sugerimos usar a mascote B1T para nosso jogo ter a carinha Robótica Paraná! Para isso, basta acessar a pasta compartilhada de imagens desta aula no [Drive Robótica Paraná](#) e baixar as figuras [Mascote.png](#) e [fundoCobrinha.svg](#). Baixe também a figura [pilha.png](#), que usaremos mais tarde.

No mBlock, vá na aba **"Atores"** e clique em **"Adicionar"** e **"Carregar"**. Na sua pasta de Downloads, selecione o arquivo [Mascote.png](#), a cabeça da cobrinha do nosso jogo, clicando em **"Abrir"**.

Figura 3 - Mascote B1T

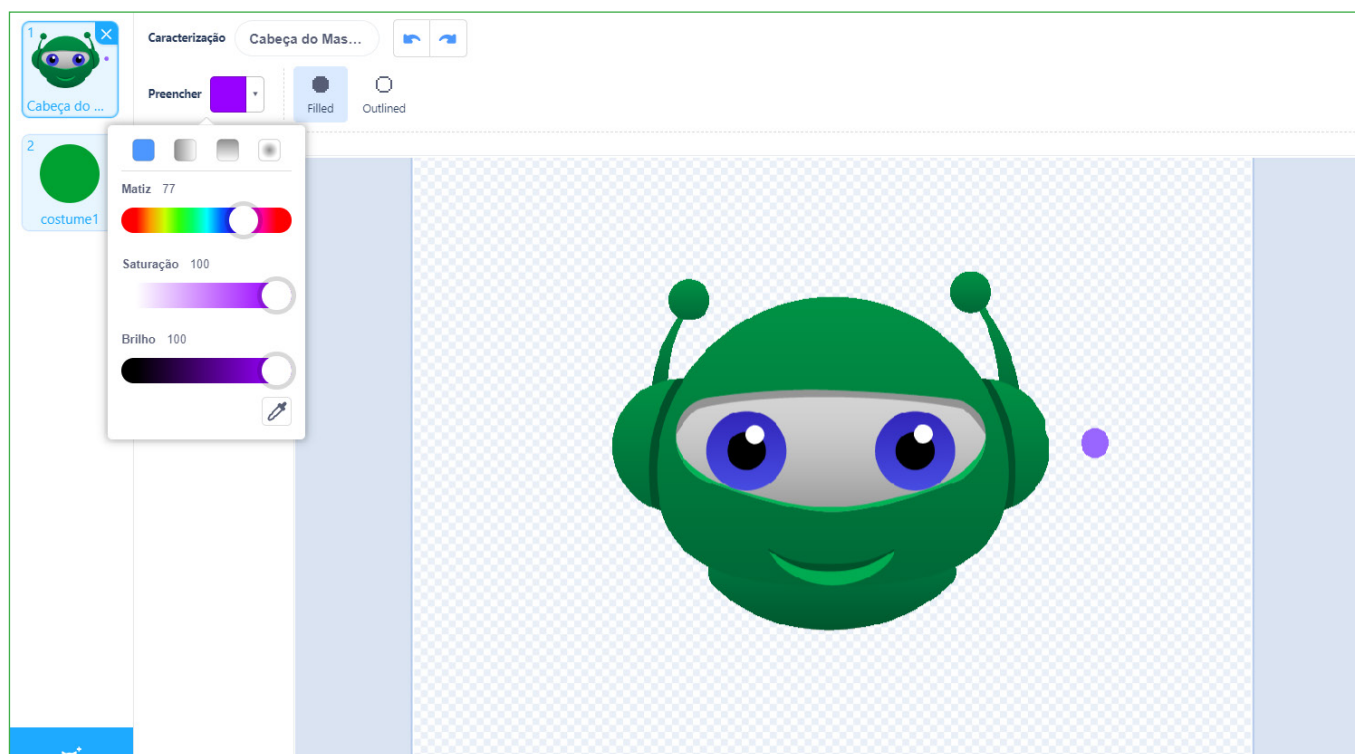


Fonte: Seed/DTI/CTE.

Será aberta a aba de **caracterização** e é nesse ponto que criaremos algo muito importante, a forma como a cobrinha detectará as bordas do jogo ou o próprio corpo durante sua movimentação.

Na opção **"Preencher"**, ajuste o brilho para **100** e na opção **"matiz"**, escolha uma cor (ela precisará ser diferente tanto do cenário de fundo, quanto do corpo da cobrinha, quanto das bordas do cenário), como **roxo** (mas pode ser rosa, vermelho, amarelo ou outra cor que se diferencie do que iremos fazer), e com o formato de circunferência desenhe uma bolinha do lado direito do B1T.

Figura 04 - Configuração da cabeça da cobrinha

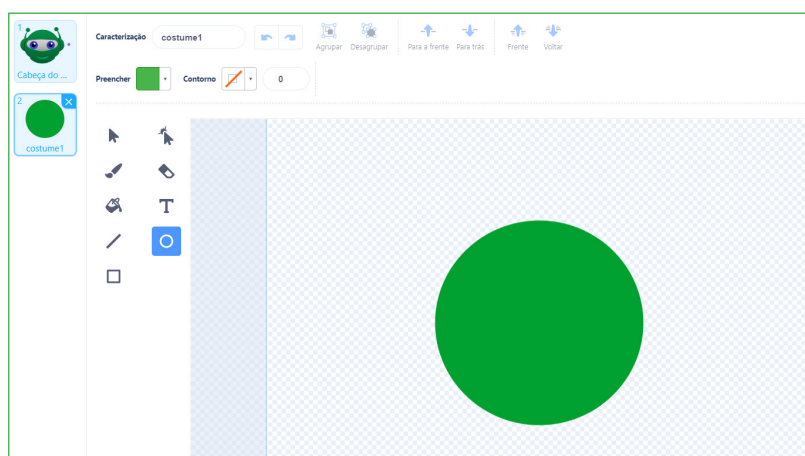


Fonte: mBlock.

Para fazer o corpinho da cobrinha, clique na opção "**Adicionar Fantasia**", disponível na parte inferior, e selecione qualquer uma das figuras. Apague o que quer que tenha na figura e desenhe um círculo.

Figura 05 - Desenhando o corpo da cobrinha

Na opção "**Preencher**", escolha uma cor para o corpo da sua cobrinha, tomando o cuidado de ser diferente da cor que você escolheu anteriormente na configuração, vá no desenho de circunferência e então desenhe um círculo de forma similar ao mostrado a seguir.



Fonte: mBlock.

Figura 06 - Criação de variáveis

Agora, vamos à lógica do jogo! Saia da área de caracterização, clicando em **"X Caracterização"**, para voltar à seção de programação em blocos dos atores.

É importante notar que nesse projeto iremos iniciar várias coisas ao mesmo tempo, por isso, para facilitar, vamos criar vários blocos de funções para ficar mais simples a compreensão.

Na aba **"Variáveis"**, crie 5 variáveis: **<BAIXO>**, **<CIMA>**, **<ESQUERDA>**, **<DIREITA>** e **<tamanho>**.

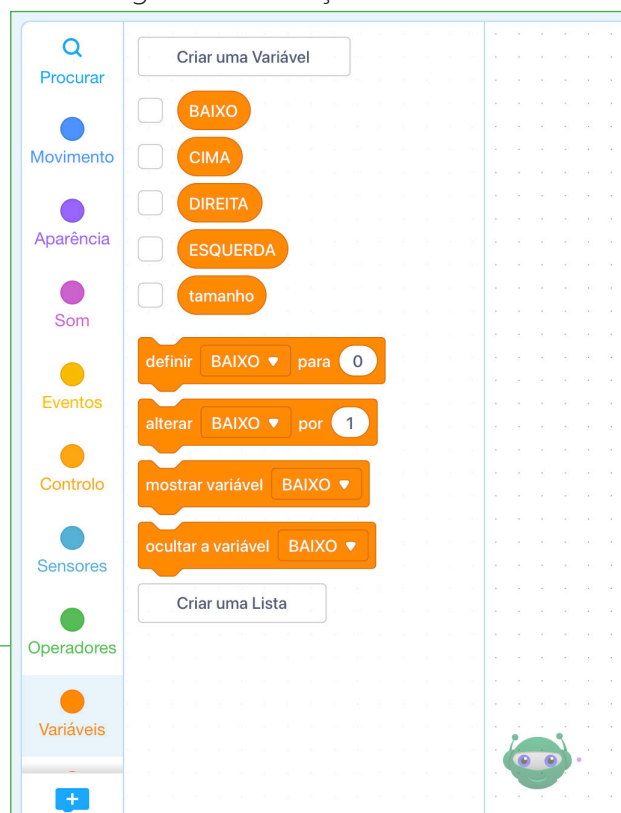
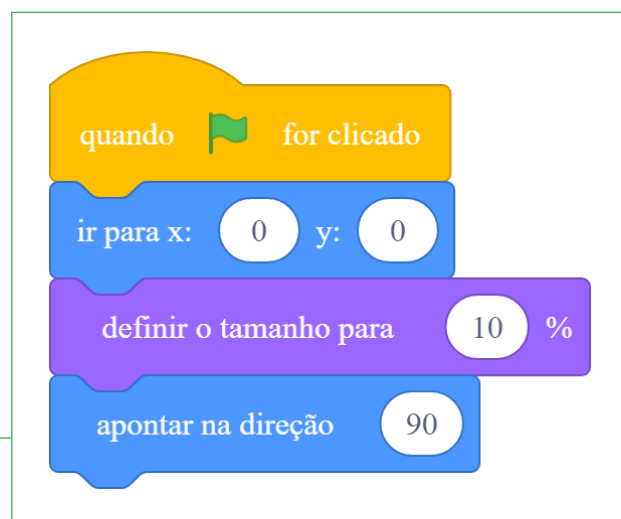


Figura 7 - Bloco para centralização da cobrinha ao iniciar o jogo

No primeiro bloco, após **< quando bandeira for clicado >**, crie um bloco de código para, sempre que o jogo iniciar, centralizar o ator da cobrinha no cenário e seu rosto na posição correta. Defina também o tamanho para 10% para que o jogo tenha maior usabilidade.



Fonte: mBlock.

Figura 8 - Definindo variáveis e programando a movimentação da cobrinha

Para programar os comandos, <**quando bandeira for clicado**> defina as variáveis CIMA, BAIXO, ESQUERDA E DIREITA para **0**, e na parte de <**repetir para sempre**> faça as seguintes verificações:

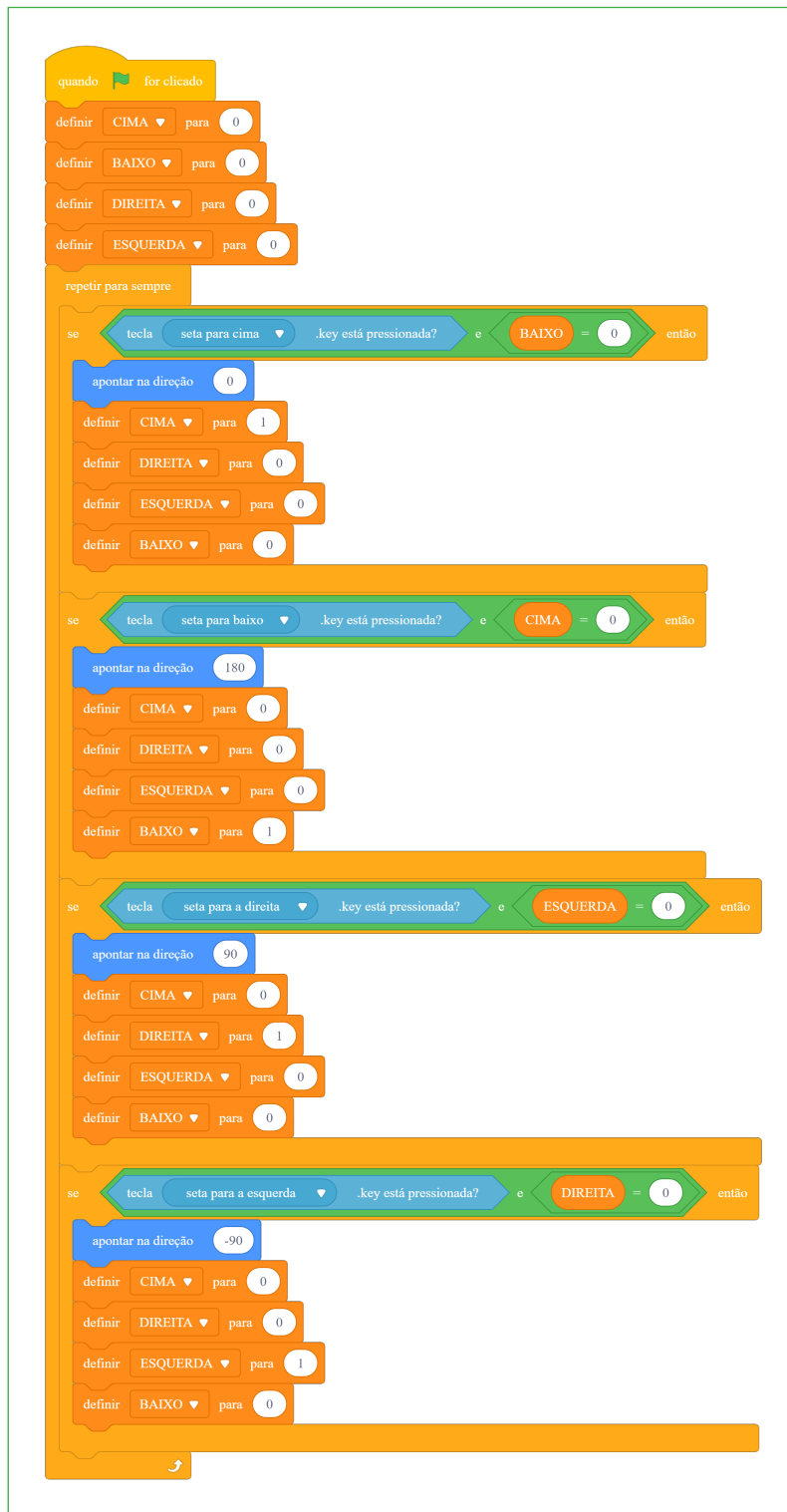
Se <**seta para cima está pressionada**> e a cobrinha não estava indo para baixo <**BAIXO =**>, então, <**apontar na direção 0**> quer dizer que a cobrinha pode subir;

Defina CIMA como **1** e o restante como **0**;

Se <**seta para baixo está pressionada**> e a cobrinha não estava indo para cima <**CIMA =**>, então, <**apontar na direção 180**> quer dizer que a cobrinha pode descer;

Defina **BAIXO** como **1** e o restante como **0**;

De forma análoga, faça essa mesma verificação para seta esquerda (<**apontar na direção -90**>) e seta direita (<**apontar na direção 90**>), isso serve para direcionar a cobrinha por onde ela pode andar e impede que, se ela estiver indo para baixo, simplesmente suba, encoste em si mesma e morra toda hora. Veja o código completo nas figuras abaixo.



Fonte: mBlock.

Estabelecido o ponto de início e a movimentação, vamos para o corpo da cobrinha.

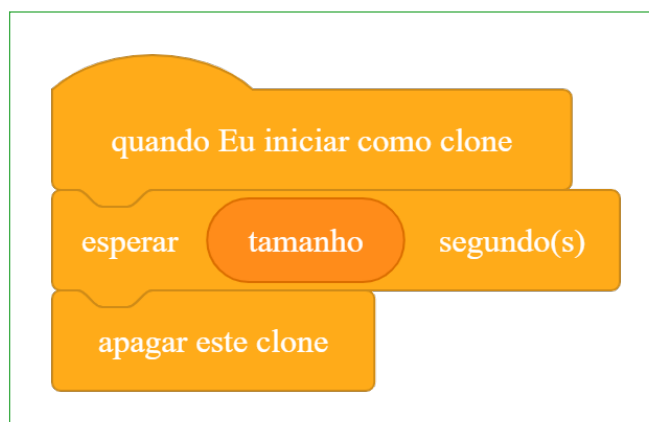
Figura 09 - Aumentando o corpo da cobrinha



Fonte: mBlock.

Figura 10 - Bloco auxiliar para movimentar e aumentar o corpo da cobrinha

Se deixarmos o código assim, o corpo da cobrinha irá crescer infinitamente e não se movimentará, portanto, precisamos que, quando ele inicie como clone, ele espere uma quantidade de tempo específica, no nosso caso, será o tamanho, então conforme ele aumenta de tamanho, ele vai se movimentando, depois de esperarmos esse tempo, para dar a impressão de movimento, iremos apagar o clone.

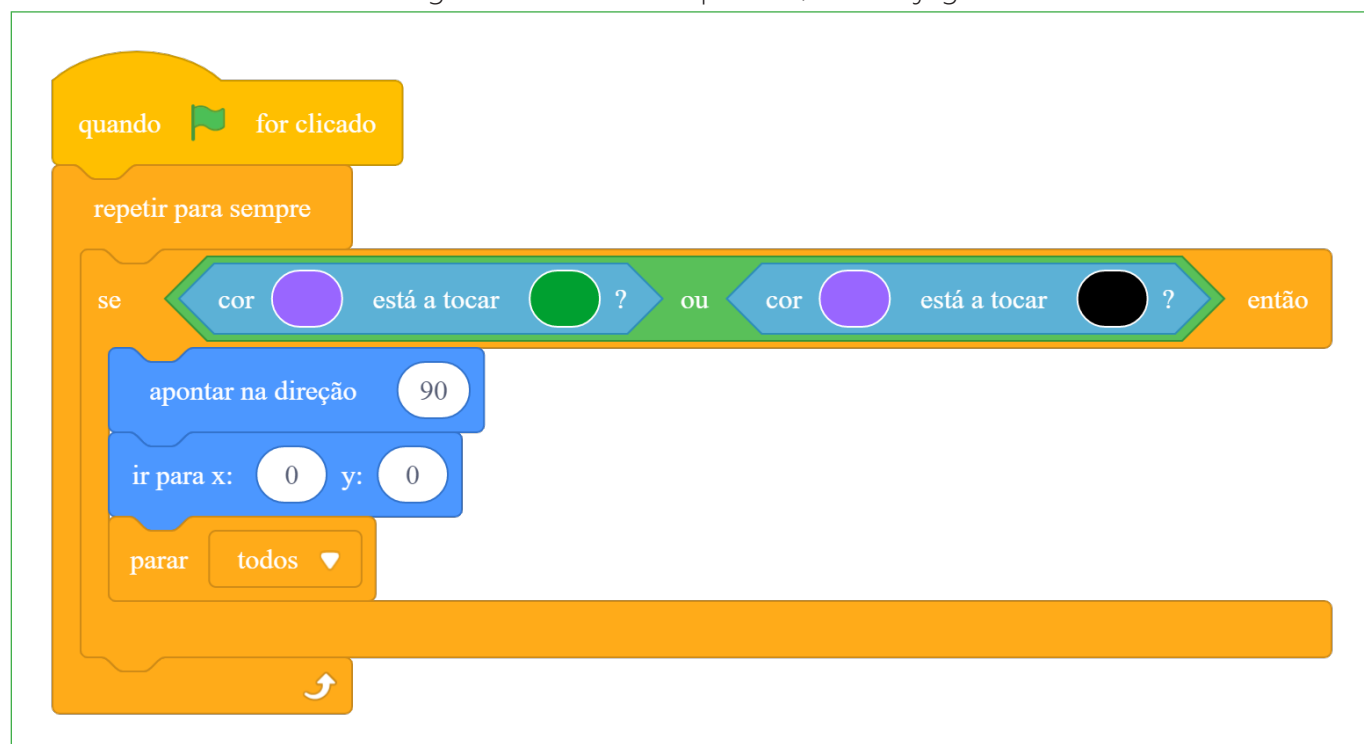


Fonte: mBlock.

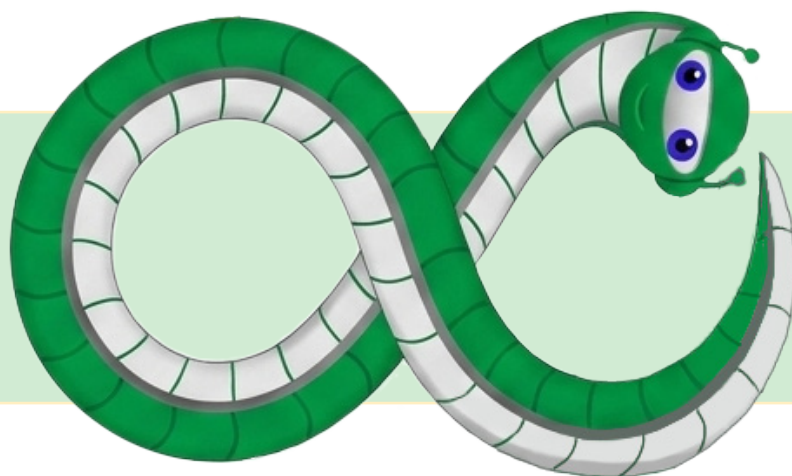
Jogo da cobrinha - I

O jogo termina quando a cobrinha toca o próprio corpo ou a borda do cenário, é por isso que configuramos a cobra de forma que sua cabeça sempre aponte para a direção que ela está indo, dessa forma, é possível verificar quando aquele pontinho que desenhamos ao lado dela logo no início, toca em qualquer coisa que não deve e assim o jogo encerra.

Figura 11 - Critério de parada / fim de jogo



Fonte: mBlock.



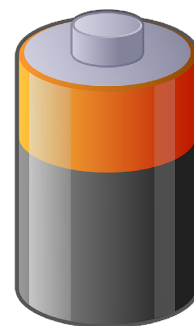
Programação em Blocos - Comida

Para dar continuidade à criação do jogo, vamos programar a comida da cobrinha. Como escolhemos a mascote Bino para representar o personagem principal, decidimos utilizar pilhas como a comida do jogo – mas você pode escolher qualquer outro alimento que combine com a mascote.

Para baixar a imagem da comida, acesse a pasta desta aula no Drive, entre na pasta “**Imagens**” e baixe o arquivo “**pilha.png**”. Em seguida, no mBlock, vá até a aba “**Atores**”, clique em “**Adicionar**” e selecione, do seu computador, a imagem que acabou de baixar.

Nesta imagem, não é necessário fazer nenhuma configuração adicional. Podemos seguir diretamente para a programação em blocos.

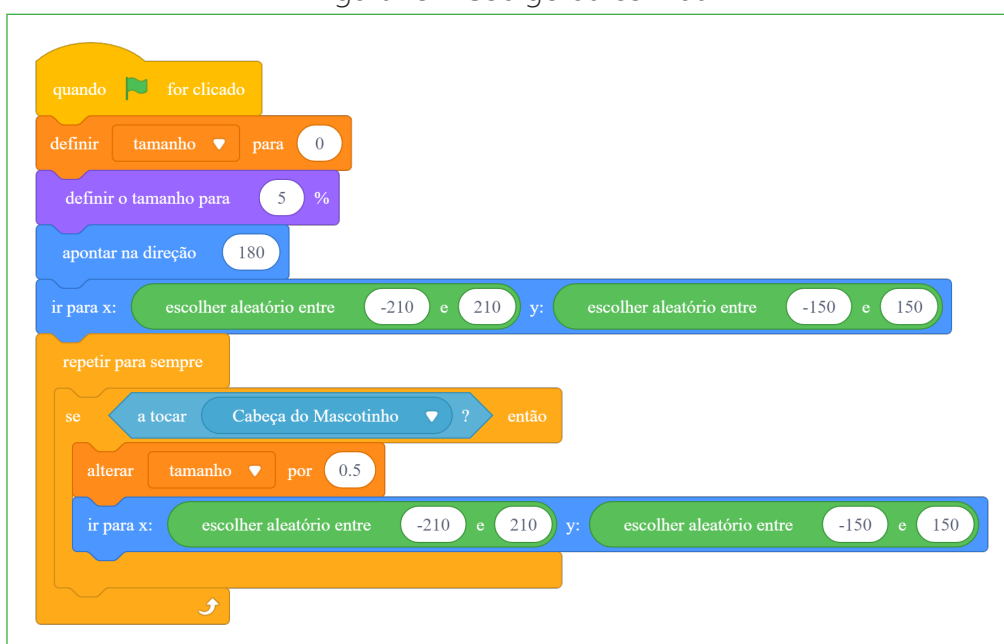
Figura 12 – Pilha



<**Quando bandeira for clicado**> defina o tamanho da cobrinha para **0**, o tamanho da pilha para metade do rosto da cobrinha, nesse caso, **5%**, e coloque a pilha na posição horizontal, então peça que toda vez que iniciar o jogo, a pilha assuma uma posição aleatória na tela.

E então repita para sempre, se a pilha tocar a cabeça da cobrinha, altere o tamanho em **0.5** e vá para uma nova posição aleatória. Veja o código completo na imagem abaixo.

Figura 13 – Código da comida



Fonte: mBlock.



Para finalizar, escolha um cenário para o seu jogo, como sugestão, está disponível na pasta desta aula um fundo claro quadriculado para que você possa fazer o download ("fundoCobrinha.png") e usar em seu próprio código, lembre-se que você deve escolher um fundo de cor diferente do ponto colorido ao lado da cabeça da sua cobrinha para que os comandos não entrem em conflito.

E finalizamos a nossa versão de um jogo clássico!!!

[Confira a programação completa do jogo no mBlock Online!](#)

Na próxima aula, a diversão vai tomar conta da aula! Reúna-se com seus amigos e simule um torneio para ver quem consegue fazer mais pontos.

Desafios

Faça os primeiros testes de jogabilidade, observando o jogo e seus detalhes. Assim como em nossos projetos, no qual um erro é desafiador e exige um reinício, caso a cobrinha encoste, o jogo recomeça.

Que tal criar uma pontuação para o jogo e níveis de dificuldade de acordo com a criatividade de cada um?

E se...

O projeto não funcionar?

Verifique o código atribuído aos atores e cenário.

3. Feedback e finalização

Ao final da aula, execute e teste o próprio jogo, observando:

- Se a cobra se move corretamente e cresce ao comer;
- Se a comida reaparece em locais aleatórios;
- Se as colisões com as bordas estão funcionando.
- Promova uma breve roda de conversa para reflexão:
- Qual parte da programação foi mais difícil de fazer funcionar?
- O que vocês fariam diferente se pudessem continuar o jogo?

Essas discussões nos ajudam a relacionar o conteúdo prático à lógica de programação e a desenvolver autonomia e pensamento crítico e podem auxiliar a pensarmos em uma versão 2 para o nosso jogo, proposta da próxima aula! Até lá!

REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27, mai. 2024.

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)
FACULDADE DE COMPUTAÇÃO (FACOM)

PROFESSORES

- Amaury Antônio de Castro Junior
- Anderson Corrêa de Lima
- Glauder Guimarães Ghinozzi
- Graziela Santos de Araújo
- Said Sadique Adi

ESTUDANTES

- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Caetano de Medeiros Santana - Sistemas de Informação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Filipe de Andrade Machado - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Guilherme Siqueira Fiani - Engenharia de Software
- Jenniffer Oliveira Checchia - Ciência da Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Pedro Paulo de Oliveira Andrade - Ciência da Computação
- Vinicius Wagner da Silva - Engenharia de Software

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Enzo Enrico Giacomini Piolla
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Regeane Vaz Guedes
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.
Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - CompartilhaIgual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

