ROBÓTICA

AULA 41

Primeiros Passos Módulo 4





Jogo da cobrinha - II

Diretoria de Tecnologia e Inovação

PARANÁ
GOVERNO DO ESTADO
SECRETARIA DA EDUCAÇÃO

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
Viviane Dziubate

Revisão Textual
Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação
Edna do Rocio Becker

Apoio Técnico
Equipe UFMS

2025

SUMÁRIO

Introdução **2**

Objetivos **2**

Roteiro da aula **3**

1. Contextualização 3

2. Montagem e programação 4

3. Feedback e finalização 12

Referências bibliográficas **14**

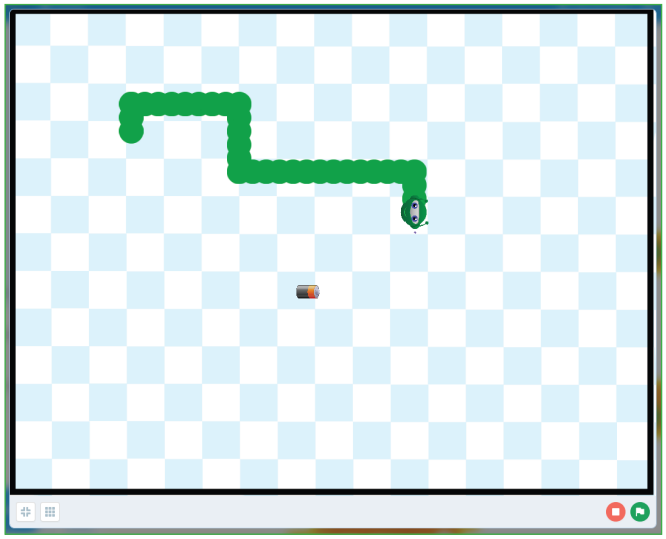
AULA 41

Jogo da cobrinha - II

Introdução

Na aula anterior, exploramos um dos clássicos arcade: o jogo da cobrinha!

Figura 1 - Jogo da cobrinha Robótica Paraná



Fonte: Seed/DTI/CTE.

Depois de aprendermos a dar vida à cobrinha e programar a comida, representada pela bateria, chegou a hora de tornarmos o jogo ainda mais interativo e divertido: controlar a cobrinha com um joystick físico!

Agora, vamos rever como o mundo virtual pode responder a ações do mundo real, entendendo que, por trás de um simples movimento do joystick, existem valores de sensores, coordenadas e tomadas de decisão programadas no código.

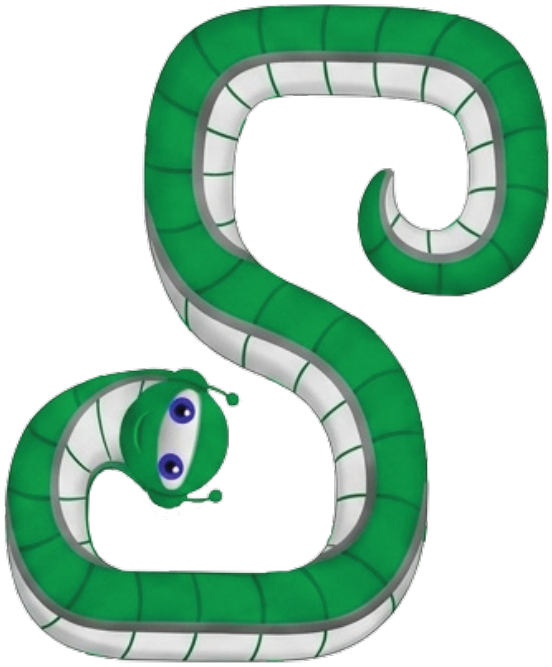
Esta aula marca a transição do jogo para uma experiência completa de interação, mostrando que a programação pode ultrapassar a tela e envolver componentes eletrônicos reais.

Objetivos desta aula

- Conectar e configurar o joystick no Arduino para controle do jogo da cobrinha;
- Adaptar o código do jogo da cobrinha para que os movimentos substituam as teclas do teclado;
- Revisar o funcionamento dos sensores analógicos para relacionar seus valores às direções X e Y do movimento.

Lista de materiais

- Notebook;
- mBlock com programação da Aula 40 - Jogo da cobrinha [parte I].



AULA 41

Jogo da cobrinha - II

Roteiro da aula

1. Contextualização

Na primeira etapa do projeto, focamos no "coração" do jogo da cobrinha: a lógica de programação, as coordenadas no plano cartesiano e as variáveis para nosso jogo funcionar no mBlock. Agora, nosso desafio atinge mais um nível, no qual focaremos na conexão entre o mundo físico do joystick e o mundo digital do mBlock, transformando a sequência de comandos da parte I em uma nova experiência para o usuário.

Como já experimentamos em alguns outros projetos de Robótica Primeiros Passos, o joystick traz o conceito IHC (Interface Humano-Computador) e, diferente de um botão simples – como chave táctil ou push button – no qual tem-se um valor digital para ligado/desligado, a haste do joystick trabalha com dados analógicos, variando entre 0 e 1023. Como veremos na sequência desta aula, o centro desse controle fica em aproximadamente 512 e programaremos "zonas de ação" (usando os limites de 300 e 700) para o Arduino entender exatamente quando o jogador quer mudar de direção, garantindo a precisão do movimento.

Essa atenção ao desenvolvimento do jogo para que ele corresponda às ações de comando é um dos aspectos abordados pelo **Design & UX** (*User Experience*, que em inglês significa "experiência do usuário"), tema introduzido na Aula 30 - Desenhando soluções, do Módulo 2 de Robótica Primeiros Passos. Um bom desenvolvedor sabe que não basta um jogo funcionar, o jogo também precisa ser jogável e atraente. E, no ponto avançado em que chegamos, as questões de **Design & UX** são um marco, pois passamos a focar mais na integração e harmonia entre hardware, software e design tanto para projetos de entretenimento, como nosso jogo, quanto projetos que visam resolver problemas e trazer soluções reais!

Como em outros projetos, compreendemos que o desenvolvimento de protótipos envolve a versão inicial de um produto ou solução, no qual o objetivo é fazer a lógica funcionar. Porém, um protótipo pode ser difícil de usar, não agradar ou ser confuso para quem nunca o viu antes... pensando em nosso jogo, se a cobrinha se move e a pontuação é computada, o protótipo está "ok". Mas, como fazer com que o "ok" vire um "uau!"? Pense com seus colegas, em aplicativos ou jogos, o que vocês mais gostam? Que os jogos sejam intuitivos, respondam rápido aos comandos e o visual não seja cansativo? Isso é Design & UX!

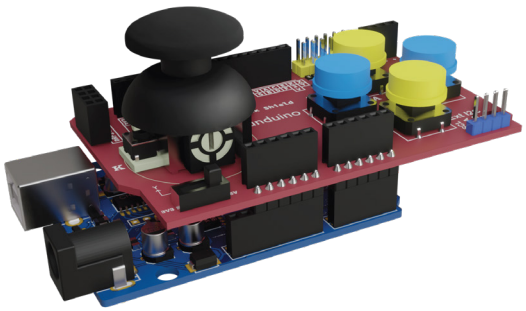
Na revisão da programação do jogo da cobrinha da aula anterior para implementação do joystick, incentivamos você e seus colegas a irem além do roteiro da aula para revisarem também o design do jogo para que, além do controle ser suave e preciso, novas cores sejam bem escolhidas, pensando na diversão do seu jogador e incentivo para que ele siga jogando.

2. Montagem e programação

Vamos nos basear no código de programação da aula anterior, copiando-o, para fazer a implementação do joystick ao nosso jogo da cobrinha.

Figura 2 - Conexão do joystick no Arduino

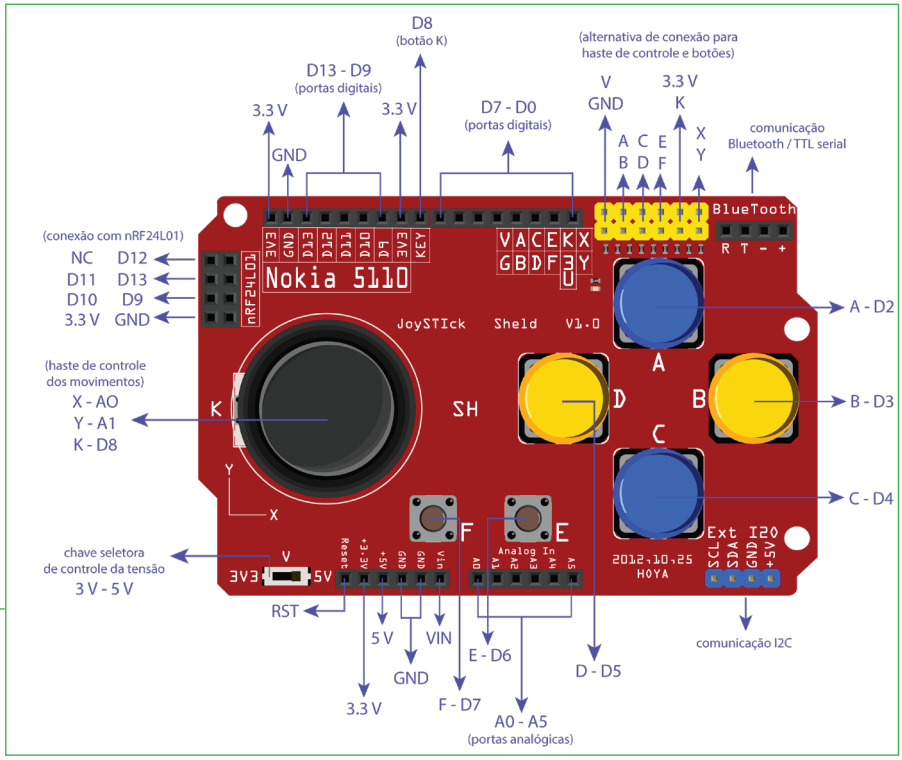
Para implementarmos o módulo joystick no nosso joguinho, precisamos trocar os códigos de controle de movimentação por teclado e mouse para as parametrizações do joystick. Mas, antes de tudo, vamos conectar o joystick no Arduino.



Fonte: Seed/DTI/CTE.

Figura 3 - Conexões do joystick ao Arduino

Considerando as conexões do joystick ao Arduino e que sua haste trabalha com valores analógicos, vamos conferir as portas que controlam os movimentos para cima e para baixo (eixo X) e os movimentos para os lados (eixo Y), essenciais para a movimentação da cobrinha.



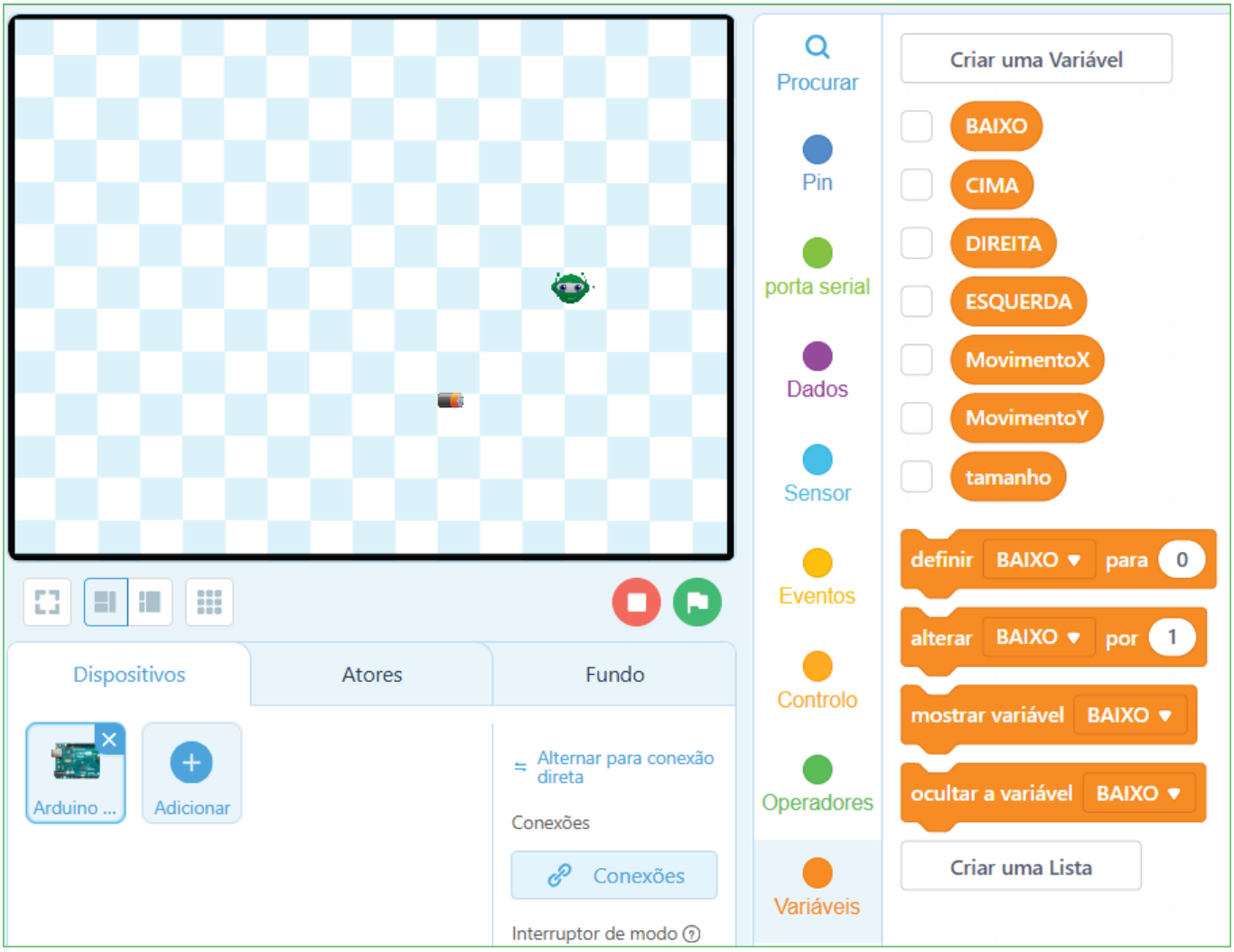
Fonte: Seed/DTI/CTE.

AULA 41

Jogo da cobrinha - II

Agora, partindo da cópia da programação da Aula 40 - Jogo da cobrinha [parte I], vamos configurar os parâmetros de controle! Primeiro, verifique a criação das variáveis **MovimentoX** e **MovimentoY** para a programação do Arduino e do ator em complemento às variáveis da versão anterior.

Figura 4 - Variáveis MovimentoX e MovimentoY

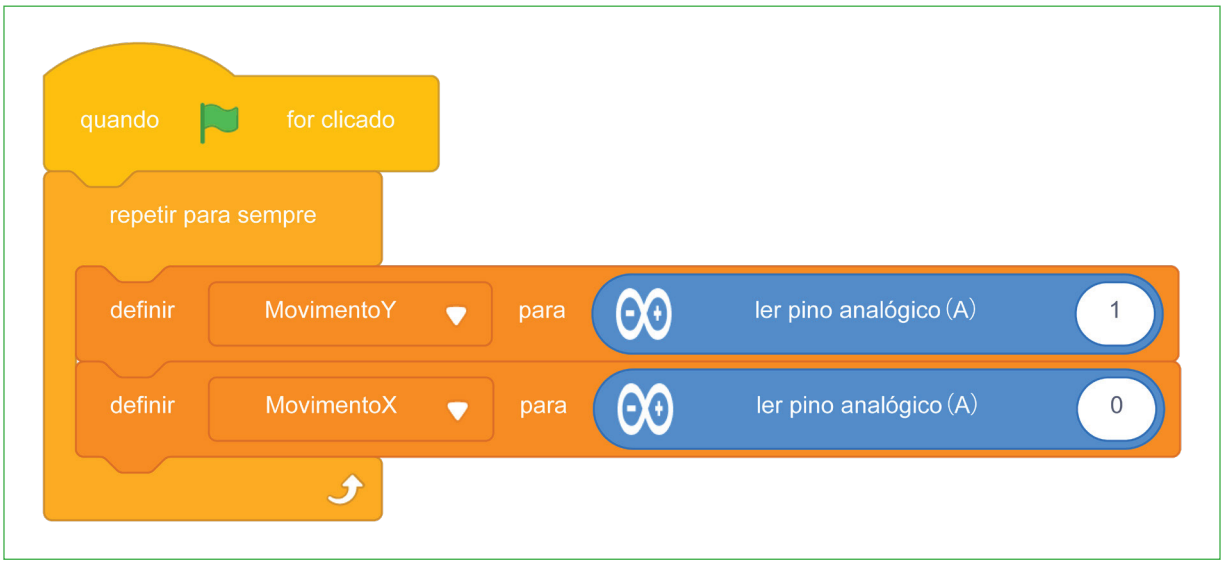


Fonte: Seed/DTI/CTE.

Jogo da cobrinha - II

Como o módulo joystick utiliza sinais analógicos para se comunicar com o Arduino, programaremos o eixo X (movimentação horizontal) para ser lido pela entrada **A0** e o eixo Y (movimentação vertical) pela entrada **A1** e essa leitura ocorrerá continuamente assim que a bandeira verde for clicada, dando início ao jogo. Ah, lembre-se que, como utilizaremos o Arduino integrado ao mBlock no modo viver, a programação da placa inicia pelo bloco < **Quando bandeira verde for clicado** >.

Figura 5 - Configuração da leitura do módulo joystick na programação do Arduino



Fonte: mBlock.

Com a conexão do joystick pronta e a programação do ator "Arduino" também, vamos entender como o Arduino interpreta os movimentos do joystick, em substituição ao teclado que utilizamos na programação da aula anterior. Os valores captados nas entradas analógicas variam em uma escala de 0 a 1023, o que posiciona o centro (quando a haste do joystick está parada) em aproximadamente 512. Para que o jogo identifique um comando intencional para cima, baixo, esquerda ou direita, definiremos as "zonas de ação": utilizaremos o valor **300** para detectar movimentos em uma direção e **700** para a direção oposta. Dessa forma, garantimos que o personagem só se mova quando você realmente deslocar o joystick, evitando que o jogo reaja a pequenos balanços acidentais enquanto o controle estiver na posição central.

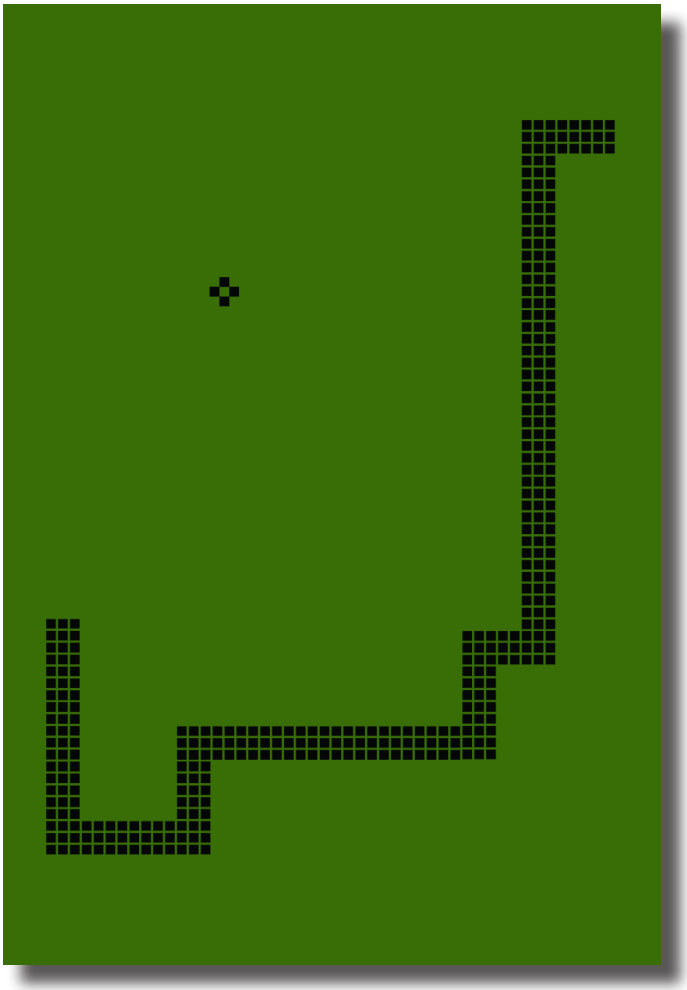
Seguindo com base na cópia da programação da Aula 40, continuamos com as definições de exibição dos atores "pilha" e "cobrinha", realizando a verificação da posi-

Jogo da cobrinha - II

ção anterior da cobrinha para evitar que ela volte instantaneamente para a direção que antes estava indo.

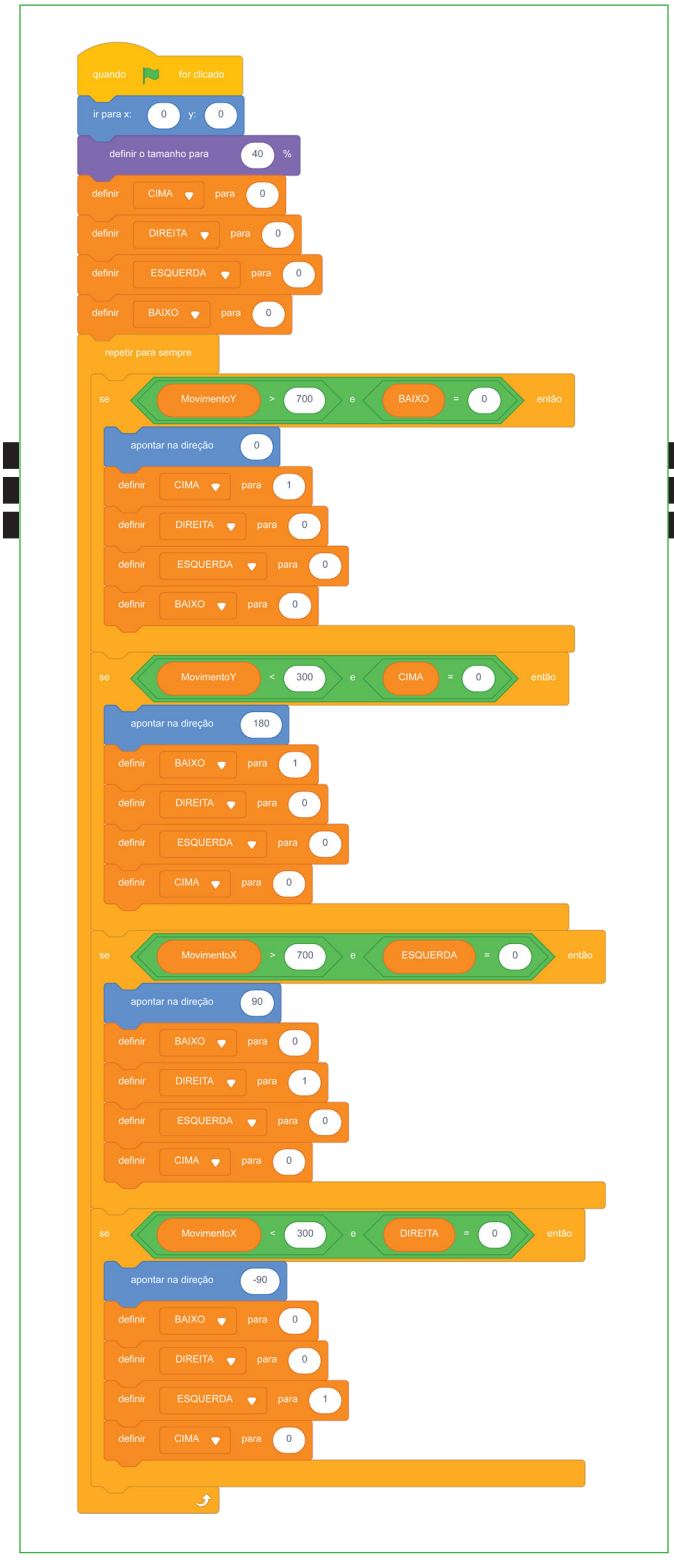
E para alterarmos o controle do jogo pelo teclado para o joystick, precisaremos mudar apenas os blocos dos operadores de detecção booleana (que apenas verificam se uma tecla está ou não pressionada) por operadores matemáticos de comparação, como "maior que" (**>**) ou "menor que" (**<**). Como o joystick opera com sinais analógicos que variam na escala de 0 a 1023, deixamos de usar uma resposta direta de "verdadeiro ou falso" (1 ou 0) para monitorar intervalos numéricos: o código agora precisa "decidir" a direção do movimento baseando-se em se a leitura do pino analógico ultrapassou o limite de 700 ou caiu abaixo de 300. Essa mudança exige que a lógica de controle seja inserida no laço de repetição contínuo < **repetir para sempre** >, permitindo que o Arduino compare constantemente a posição física do joystick com esses valores de referência para acionar a direção correta da cobrinha no plano cartesiano.

Para concretizar essa transição dos controles do nosso jogo, substituiremos os blocos < **se tecla [seta] pressionada?** >, que antes disparavam os movimentos para cima, para baixo, para a esquerda e para a direita pelo teclado, por quatro novas estruturas de decisão que monitoram os eixos físicos do joystick: < **se MovimentoY > 700** > e < **se MovimentoY < 300** > para definir os movimentos verticais e < **se MovimentoX > 700** > e < **se MovimentoX < 300** > para definir os movimentos horizontais. Essa substituição é o que permite ao código "traduzir" a inclinação física da alavanca em comandos de direção para a cobrinha, transformando a lógica do teclado em uma experiência de controle analógico real.



Jogo da cobrinha - II

Figura 6 - Alteração do controle da cobrinha em < se .... então >



Fonte: mBlock.

Jogo da cobrinha - II

Além dessas alterações, não faremos no roteiro da aula mais nenhuma mudança no restante do código, mas segue o convite para vocês **pensarem em outras cores para o jogo, alterando-as**... então, prontos para o jogo?

Confira como ficou a programação completa do jogo da cobrinha com joystick pelo mBlock online e faça o primeiro teste!

Ao testar seu jogo, considere um check-up de desenvolvimento:

Check-up do desenvolvimento

🎨 CORES (visibilidade)

[ ] A cobrinha se destaca do fundo?
[ ] A comida é fácil de encontrar?
[ ] As cores ajudam a visão?

🕹️ CONTROLES (joystick)

[ ] Para cima é para cima? (lógica)
[ ] A cobra responde rápido ao comando?
[ ] O joystick está firme?

✨ FEEDBACK (reação)

[ ] A cobra mudou de tamanho ao comer?
[ ] O jogador percebeu que ganhou ponto?
[ ] O jogo acaba com feedback ou animação?

🚀 DESAFIO (evolução)

[ ] A pontuação está funcionando?
[ ] A velocidade aumenta com o tempo?
[ ] O jogo é divertido para um colega?

Agora que os primeiros testes foram concluídos, que tal mergulharmos no mundo do Design & UX? Programar é dar vida aos projetos, e o design é o que torna essa vida agradável para quem joga ou vivencia a experiência do seu projeto.

AULA 41

Jogo da cobrinha - II

Ao programarmos a interface do mBlock e alterarmos nosso jogo para a utilização do joystick, pensamos em alguns pontos que podem ser avaliados também pelos seus colegas, como usuários, durante os testes do jogo. Então, enquanto seus colegas testam seu jogo, observem juntos estes três pontos fundamentais:

Contraste: as cores ajudam a distinguir os elementos? Por exemplo, uma cobra verde limão em um fundo preto é muito fácil de ver. Se as cores forem muito parecidas, o jogo se torna injusto. Como está o seu?

Feedback visual: o jogador recebe um sinal claro quando a cobrinha come? Pode ser um aumento no tamanho, uma piscada de luz ou uma mudança rápida de cor. O importante é o usuário sentir que sua ação teve resultado!

Ergonomia e resposta: joystick responde rápido ou existe um "atraso"? Em jogos de reflexo, a velocidade de resposta é o que separa a diversão da frustração.

Um bom designer sempre se pergunta se o jogador entende o que deve fazer sem precisar de um manual! E, para isso, cada estudante pode adotar um "check list" – agora do design – ao testar e avaliar o jogo do colega:

Design e desenvolvedor:		
Critério de Design & UX	**Avaliação (0 a 5 ⭐)**	**O que pode melhorar?**
Visibilidade Consigo distinguir facilmente a cobrinha, a comida e o fundo pelas cores?		
Controle O movimento da cobrinha através do joystick é intuitivo e rápido?		
Feedback O jogo dá um sinal claro (crescimento ou som) quando a cobrinha come?		
Interface Se a pontuação é alterada, ela é fácil de ler enquanto se joga?		
Diversão O jogo é atraente e desafiador, sem ser frustrante?		

Jogo da cobrinha - II



Conforme as avaliações pelo check-list, faça os ajustes necessários no seu código para deixar a experiência mais incrível e convide novos colegas para o desafio!

A mecânica do jogo (avançar, crescer e recomeçar) se relaciona à nossa jornada de aprendizado em Robótica. Então, aqui já vale os parabéns pela conclusão do último projeto do Módulo 4 de Robótica Primeiros Passos! Este jogo da cobrinha é um símbolo perfeito da nossa trajetória: assim como a cobrinha, nós fomos avançando e ampliando nosso repertório a cada passo. E, se em algum momento, para conquistas futuras, errarmos ou o desafio parecer difícil demais e precisarmos recomeçar, lembre-se: cada nova tentativa começa com o conhecimento que você já conquistou!





Jogo da cobrinha - II

Desafios:

O que você e seus colegas mudariam no visual ou nos controles para tornar o jogo mais divertido e dinâmico para quem joga?

Como podemos tornar o jogo mais emocionante? Tente programar para a cobrinha aumentar gradualmente a velocidade (diminuindo o tempo de espera entre os movimentos) conforme a pontuação sobe e um som para cada ação concluída.

Existe um modo de encurtar o código principal do ator, dividindo-o em subcódigos. Que tal explorar com seus colegas a criação de funções?

E se...

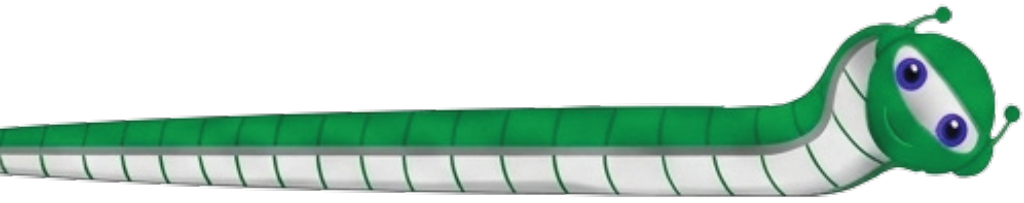
O projeto não funcionar?

- Verifique o código atribuído aos atores e cenário.
- Confira as conexões do joystick ao Arduino e os parâmetros utilizados na programação.

3. Feedback e finalização

Reflita, com seus colegas, sobre a experiência do jogo! O que foi mais fácil ou mais difícil nesta etapa de traduzir os movimentos do joystick para a tela? Ao testar uma partida com o joystick, o jogo reagiu como vocês esperavam quando a cobrinha comeu a comida ou houve algum comportamento surpresa que precisou de ajuste?

Assim como no decorrer dos nossos projetos de Robótica vamos refinando montagens e programações para melhorias, precisamos pensar também no público-alvo do nosso projeto, ou seja, quem vai utilizar o jogo. Olhando para o Design & UX, pensem: como as cores escolhidas após seus últimos ajustes ajudam na jogabilidade, facilitando para o jogador distinguir os elementos? Além disso, o controle pelo joystick está 'confortável' e intuitivo? Um bom designer sempre se pergunta se o jogador entende o que deve fazer sem precisar de um manual! E, para isso, cada estudante pode adotar um "check list de design" e testar o jogo do colega:

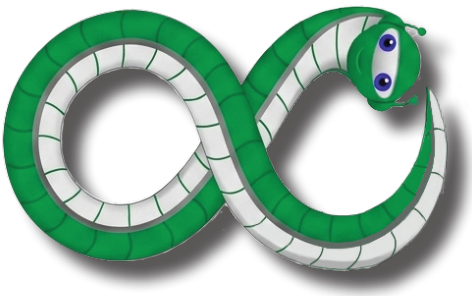




Jogo da cobrinha - II

Nome do design e desenvolvedor:		
Nome do jogador teste:		
Critério de Design & UX	**Avaliação (0 a 5⭐)**	**O que pode melhorar?**
Visibilidade Consigo distinguir facilmente a cobrinha, a comida e o fundo pelas cores?		
Controle O movimento da cobrinha através do joysti-ck é intuitivo e rápido?		
Feedback O jogo dá um sinal claro (crescimento ou som) quando a cobrinha come?		
Interface Se a pontuação é alterada, ela é fácil de ler enquanto se joga?		
Diversão O jogo é desafiador sem ser frustrante?		

Parabéns pela dedicação e não apenas conectar e programar, mas em transformar!



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



DTI - DIRETORIA DE TECNOLOGIA E INOVAÇÃO

PARANÁ
GOVERNO DO ESTADO
SECRETARIA DA EDUCAÇÃO