ROBÓTICA

AULA 13

Primeiros Passos Módulo 3


Imagem gerada com I.A.

Jogo do pinhão - II

VERSÃO 2

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 de Miranda

DIRETOR DE TECNOLOGIA E INOVAÇÃO
Claudio de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS
Marcelo Gasparin

Produção de Conteúdo
Ailton Lopes
Andrea da Silva Castagini
Viviane Dziubate Pittner

Validação de Conteúdo
Cleiton Rosa
Darice Alessandra Deckmann Zanardini

Revisão Textual
Kellen Pricila dos Santos Cochinski

Projeto gráfico, diagramação e geração de imagens IA.
Edna do Rocio Becker

Ilustrações
Jocelin Vianna

Apoio Técnico
Equipe UFMS

2024

SUMÁRIO

AULA 13

Jogo do pinhão - II

Introdução

Na última aula, exploramos a natureza do Estado do Paraná e programamos um jogo no mBlock contemplando elementos como a gralha-azul e o pinhão.

Nesta aula, avançaremos no nosso projeto com a integração de um novo elemento à narrativa do jogo, trazendo mais desafios e tornando nosso projeto mais interessante! Além disso, ampliaremos o controle da nossa personagem com a possibilidade de uso de um joystick ao teclado. Preparados?

Imagem gerada com I.A.

AULA 13

Jogo do pinhão - II

Objetivos desta aula

- Conscientizar sobre a importância da preservação da mata de Araucárias e do Pinheiro do Paraná;
- Ampliar a programação do jogo interativo em mBlock, sobre a temática proposta;
- Adicionar novo personagem ao jogo, tornando-o mais desafiador;
- Adicionar um joystick para controle do jogo;
- Retomar o uso do joystick shield para controle de jogos.

Roteiro da aula

1. Conteúdo

Atualmente, jogos eletrônicos permitem que jogadores utilizem vários meios de entrada de comandos, desde teclados e controles até volantes e óculos de realidade virtual. Por que não tornar nosso jogo um pouco mais moderno por meio da possibilidade de o usuário conectar um controle ao computador e influenciar o movimento do personagem com o movimento dos sticks analógicos? Indo mais além, que tal aumentarmos a dificuldade do nosso jogo adicionando um inimigo? É isso que faremos agora!

Lista de materiais

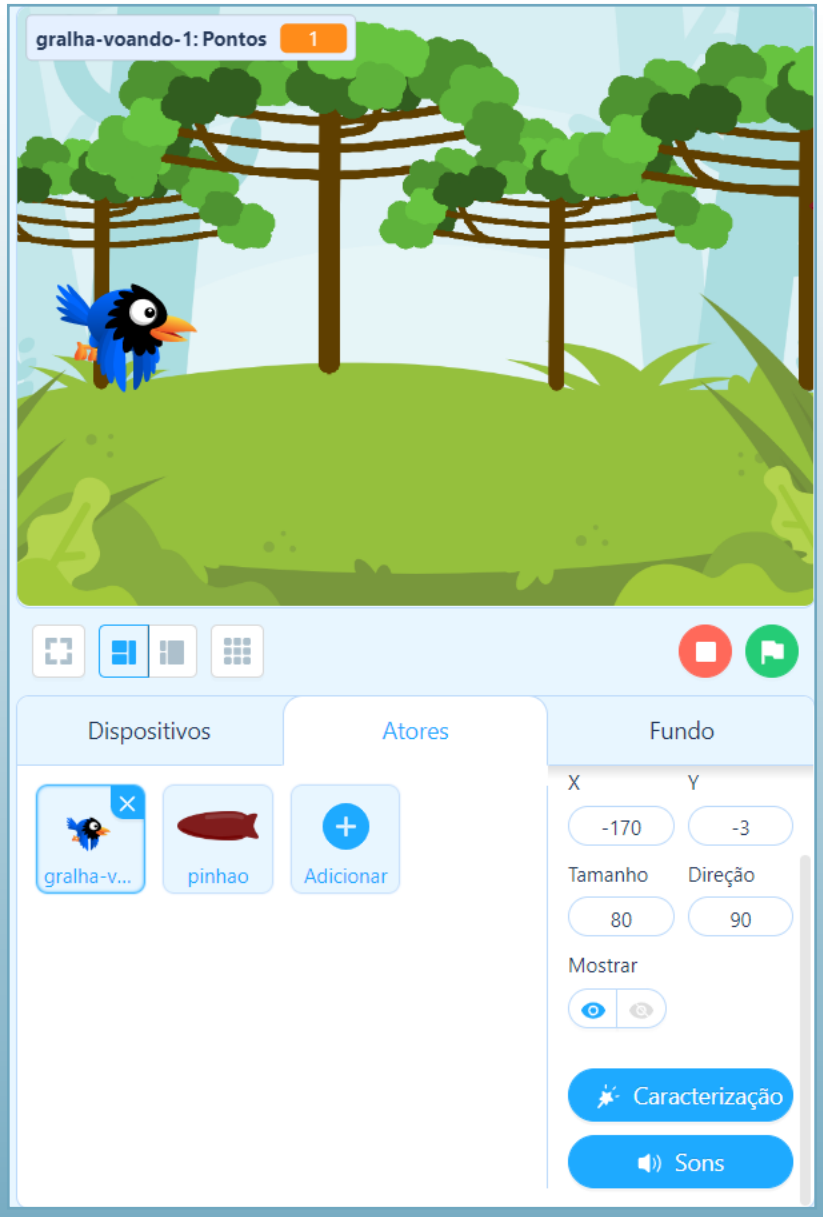
- Computador (com acesso à internet);
- Software MBlock;
- Arduino Uno;
- cabo USB;
- shield de joystick para o Arduino Uno;
- Imagens para utilizar no jogo (sprites da Águia).

Jogo do pinhão - II

2. Roteiro de Programação

Vamos dar continuidade partindo do código da aula anterior. Para isso, vamos adicionar um Arduino ao projeto. Primeiro, clique em "**Dispositivos**" no lado esquerdo da tela. Depois, clique no " + " azul. Veja a Figura 1.

Figura 1 - Menu de dispositivos.

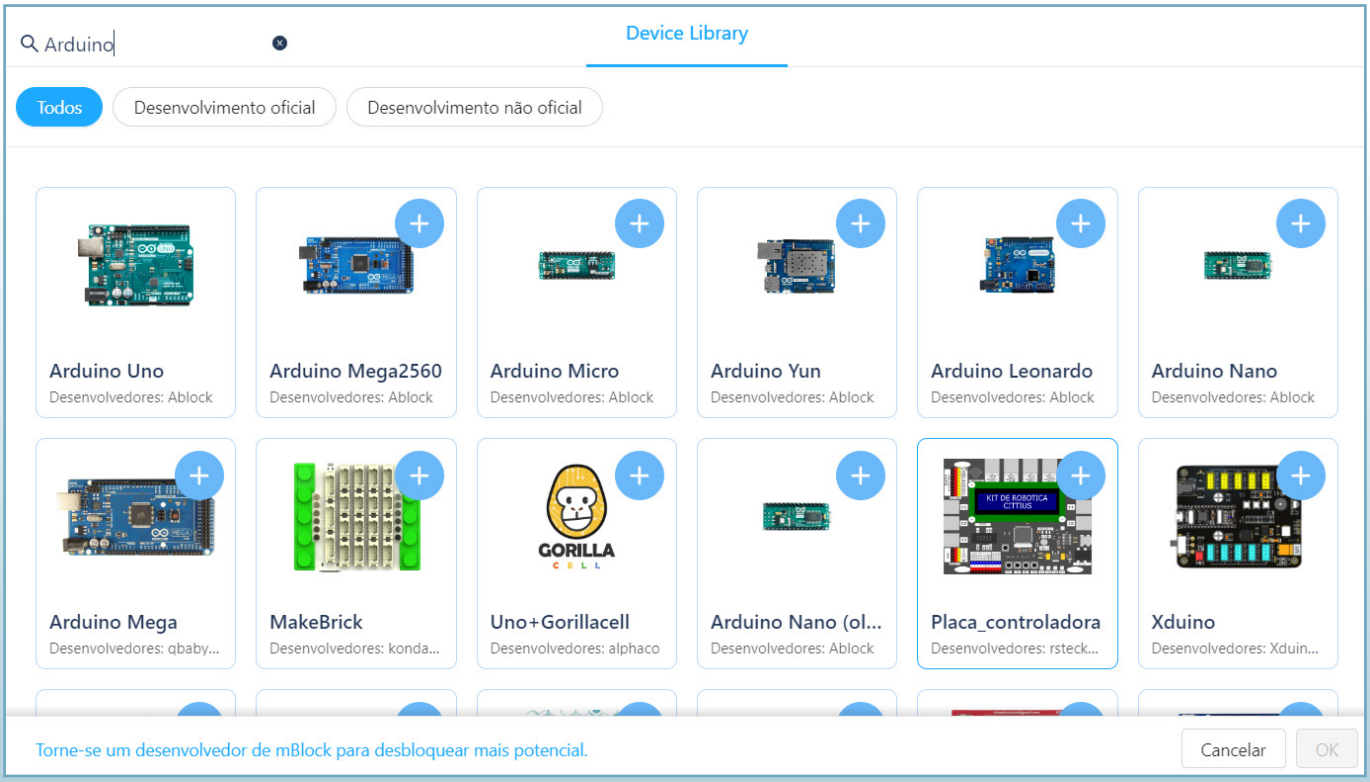


Fonte: mBlock, 2024 (adaptado).

Jogo do pinhão - II

Agora, na janela que se abriu, pesquise por "**Arduino**", selecione a primeira opção ("**Arduino Uno**", a placa média com um tom de verde) e clique em "**OK**" no canto inferior direito da janela. Veja a Figura 2.

Figura 2 - Lista de dispositivos disponíveis.



Perceba que agora o controle é mostrado na lista da Figura 1. Clique nele e veja que o quadro de blocos está vazio novamente, precisamos criar os blocos para o controle. Antes disso, devemos habilitar o modo "**Viver**" no Arduino. A lógica aqui será simples: quando o mBlock detectar que o controle analógico foi movido para cima, a gralha irá se mover para cima; se o mesmo controle analógico se mover para baixo, a gralha irá se mover para baixo.

Jogo do pinhão - II

Figura 3 - Esqueleto dos blocos do controle.

Primeiro, vamos montar o esqueleto disso para que o Arduino seja executado apenas enquanto estamos jogando, veja na Figura 3.

Vamos criar uma variável (para todos os atores) de nome "**leituraY**" para armazenar a leitura do pino (pino analógico A1) do joystick, veja na Figura 4.

Figura 4. Variável "leituraY" e leitura do Joystick adicionadas.

AULA 13

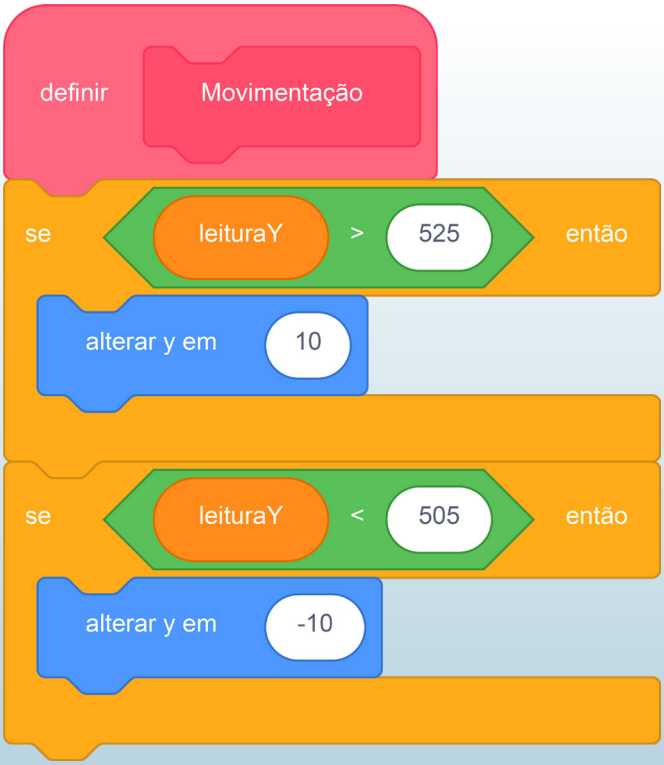
Jogo do pinhão - II

Para fazer a gralha se movimentar a partir da posição do joystick, clique em "**Atores**" e depois em "**gralha**" para visualizar os blocos referentes a esse ator. Agora, vamos criar nosso próprio bloco para armazenar os blocos que irão realizar o movimento da gralha. Assim, clique em "**Os Meus Blocos**" e depois em "**Criar um Bloco**", coloque o nome do nosso novo bloco de movimentação como mostrado na Figura 5.

Figura 5 - Criação do novo bloco "Movimentação".

Jogo do pinhão - II

Figura 6 - Construção do bloco "definir Movimentação".



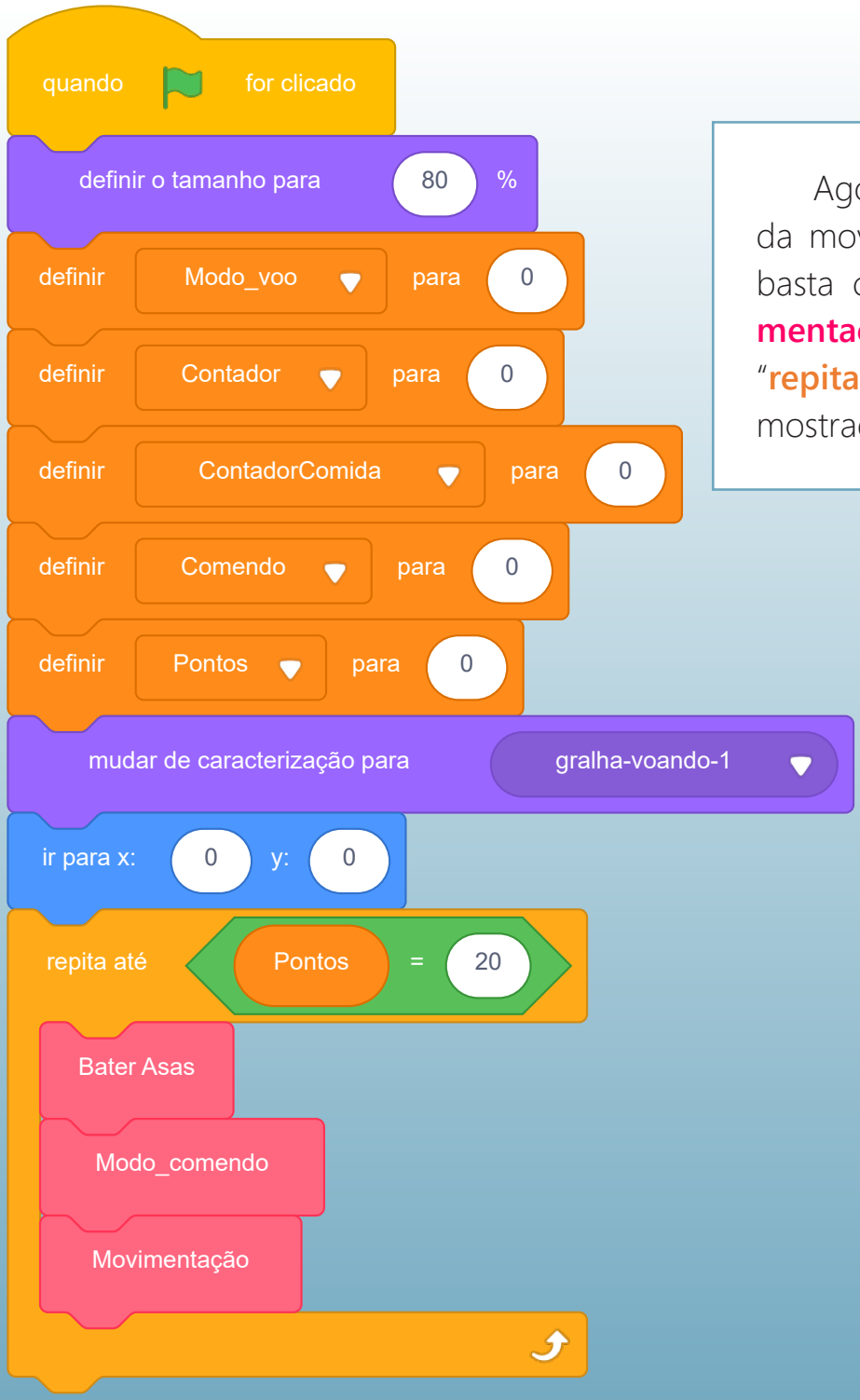
Clique em "**OK**" para concluir a criação do nosso bloco. Agora, no bloco "**definir Movimentação**" gerado em nossa "área de trabalho" no mBlock vamos adicionar os blocos apresentados na Figura 6.

Explicando o que está dentro das condições (blocos em verde) da Figura 6: no primeiro bloco "**se**" estamos perguntando ao Arduino se a leitura que ele obteve do joystick é maior que 525, o que significa que o analógico está para cima, assim a posição da "**gralha**" é alterada para cima (posição do **eixo Y**); no segundo bloco "**se**" estamos perguntando ao Arduino se a leitura que ele obteve do joystick é menor que 505, o que significa que o analógico está para baixo, assim a posição da "gralha" é alterada para baixo (posição do **eixo Y**).

Utilizamos esses valores na condição porque as leituras do analógico, quando ele está na sua posição padrão, varia um pouco, mas é praticamente 512, ou seja, a metade do valor máximo obtido pelas leituras das portas analógicas do Arduino que é de 512.

Jogo do pinhão - II

Figura 7 - Adição do bloco "Movimentação" ao "repita até" da "gralha".

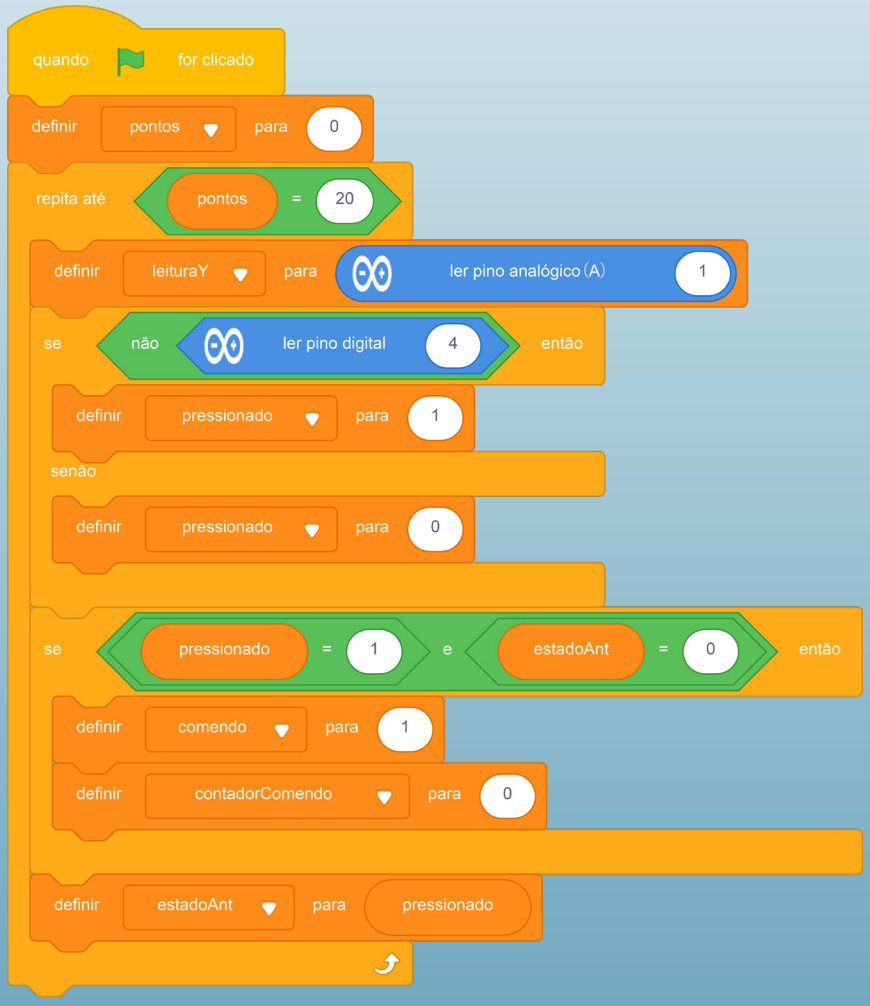


Agora, para concluir a parte da movimentação da "**gralha**", basta colocar o bloco "**Movimentação**" dentro do bloco "**repita até**" desse ator, como mostrado na Figura 7.

Jogo do pinhão - II

Pronto, concluímos o sistema de movimentação da "**gralha**". No entanto, está faltando uma das ações mais importantes da "**gralha**", comer os pinhões. Para isso, vamos detectar se o usuário clicou no "**Botão C**" da nossa shield, ou seja, ocorreu um evento de alteração do estado de não pressionado para pressionado desse botão. Para isso, volte aos blocos do Arduino e crie as variáveis "**pressionado**" e "**estadoAnt**" (Para todos os atores). A variável "**pressionado**" irá receber o estado atual do botão e o "**estadoAnt**" receberá o estado anterior do botão, com esses valores é possível saber se o botão foi clicado ou está sendo pressionado, ou solto. Quando o evento de clique ocorrer queremos que a "**gralha**" coma o pinhão que está próximo a ela, desta forma, vamos trazer para o Arduino os blocos que eram executados quando a tecla espaço era clicada. Para isso, faça as alterações apresentadas na Figura 8.

Figura 8 - Evento de clique do botão.

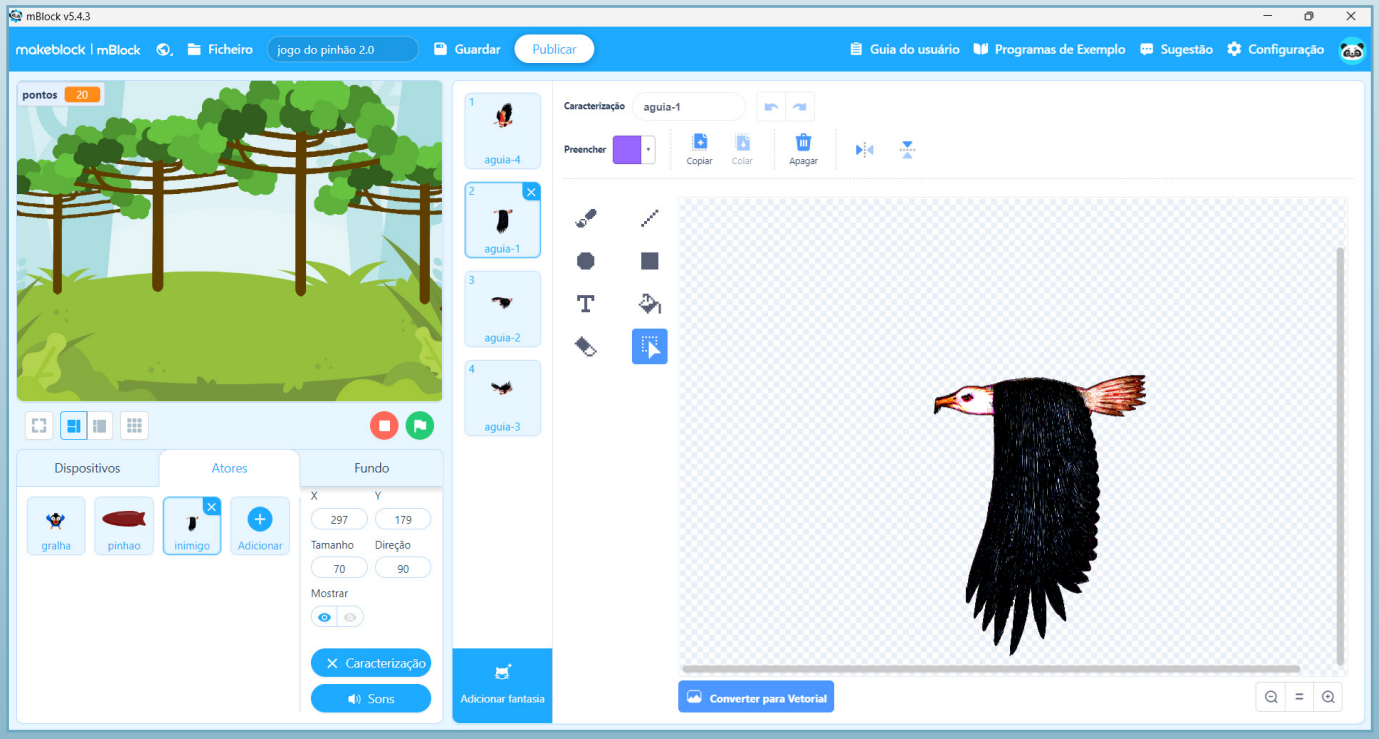


Jogo do pinhão - II

Desta forma, concluímos a parte de movimentação e de ações da "**gralha**" por meio de um Arduino com uma shield com joystick. Logo, jogar nosso jogo se tornou mais divertido e desafiador, mas, no mundo real, enquanto os animais se alimentam eles devem tomar cuidado com seus predadores, assim, adicionaremos um inimigo para que nossa "**gralha**" fique mais atenta na hora de se alimentar.

Para esta etapa, será utilizada uma lógica similar à aplicada à "**gralha**" para gerar sua animação de bater asas. Inicialmente, vá a aba de "**Atores**" e crie um novo ator como nome "**Aguia**". Faça o upload do arquivo "**aguia-1.png**" e depois adicione como fantasia para este ator os arquivos "**aguia-2.png**", "**aguia-3.png**" e "**aguia-4.png**" disponíveis na pasta "Imagens". A conclusão desses passos resulta na Figura 9.

Figura 9 - Criação do ator "Aguia" e suas fantasias.



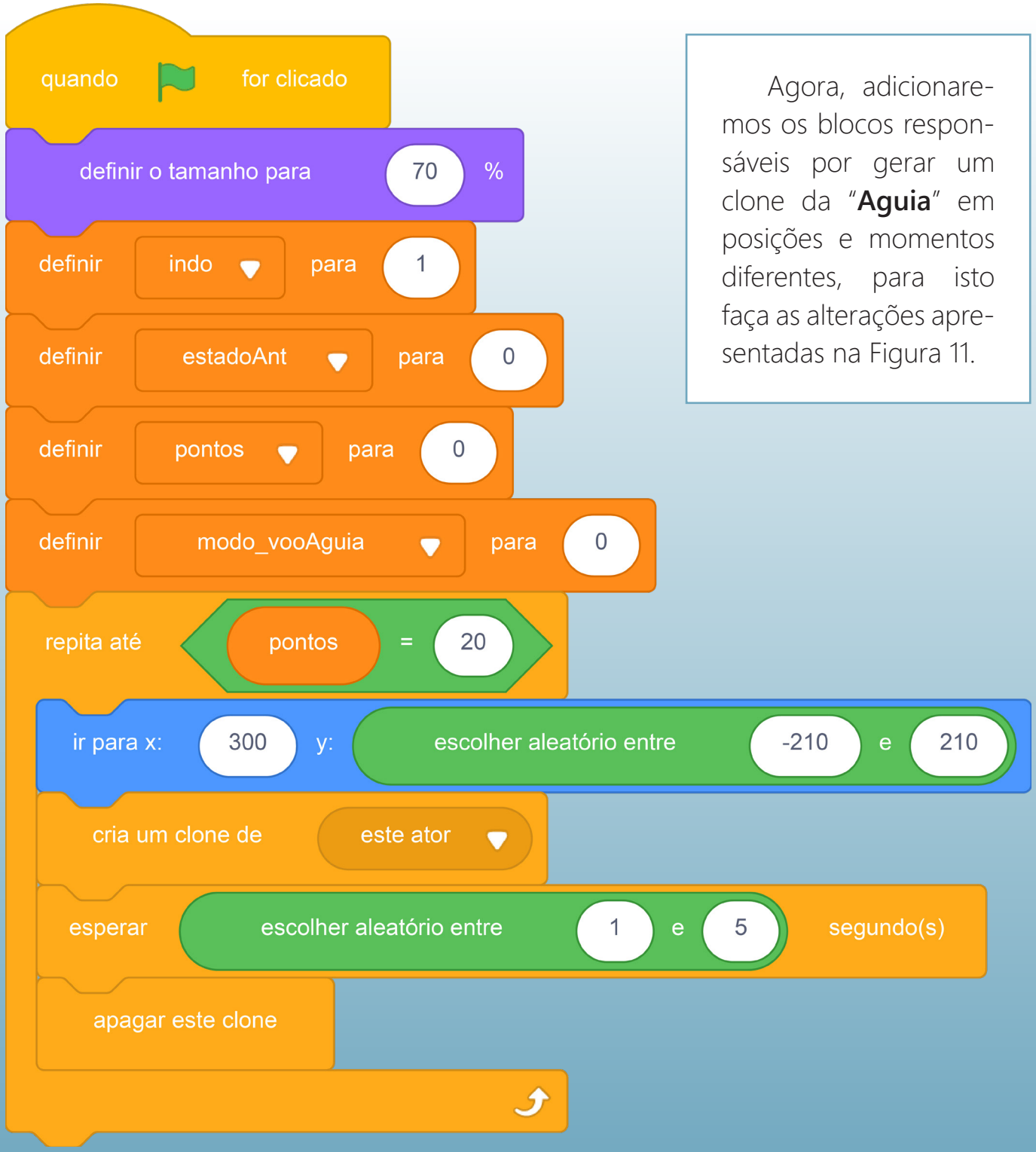
Jogo do pinhão - II

Figura 10 - Esqueleto dos blocos do ator "Aguia".

Agora, vamos criar as variáveis para o comportamento da "Águia". Crie as variáveis "indo" e "modo_vooAguia" e construa o código como pode ser visto na figura 10.

Jogo do pinhão - II

Figura 11 - Gerador de clones da águia.

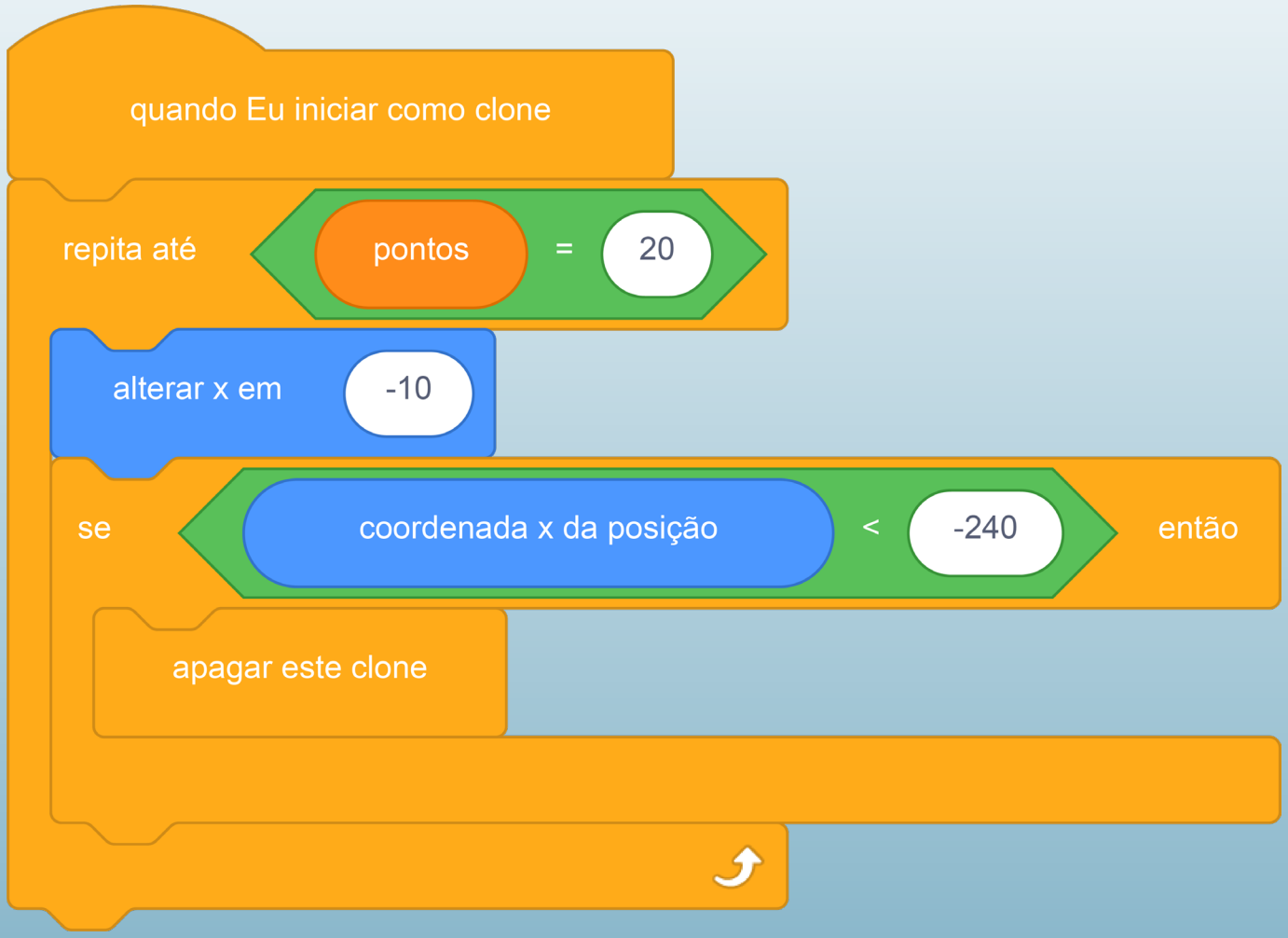


Agora, adicionaremos os blocos responsáveis por gerar um clone da "**Aguia**" em posições e momentos diferentes, para isto faça as alterações apresentadas na Figura 11.

Jogo do pinhão - II

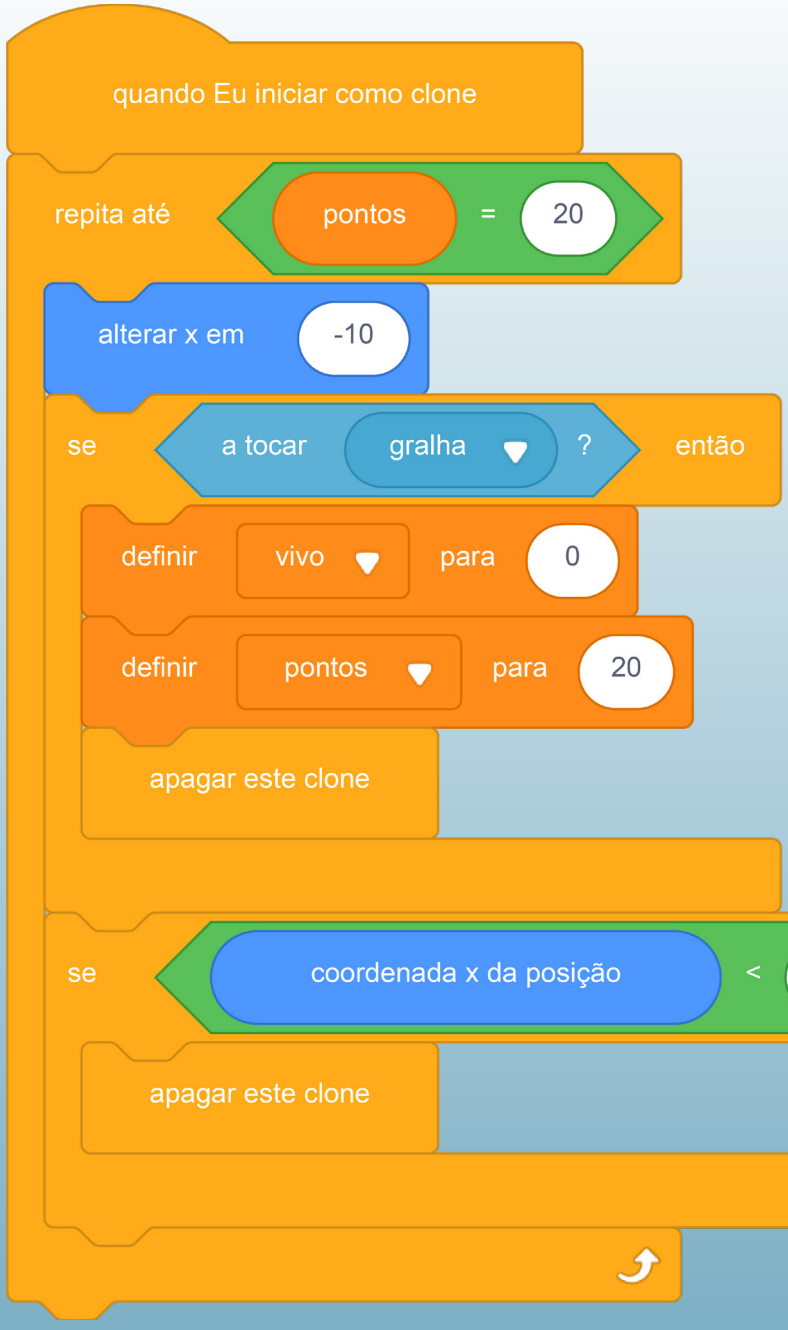
Com essa parte concluída, daremos início à movimentação e animação de cada **clone**. Para animação, desejamos que a "**Aguia**" se movimente no **eixo X** saindo do canto direito e indo até o canto esquerdo, onde irá desaparecer. Para isso, iremos utilizar o bloco "**Quando eu iniciar como clone**" e um "**repita até**" como pode ser visto na Figura 12.

Figura 12 - Movimentação dos clones.



Jogo do pinhão - II

Figura 13 - Evento de colisão entre "gralha" e "Aguia"



Agora, adicionaremos o evento de contato entre a "**Aguia**" e a "**gralha**". Quando isso acontecer, desejamos que o jogo pare e seja executada uma animação de derrota que será apresentada mais adiante. Por agora, para encerrar o jogo, crie a variável "**vivo**" e adicione o bloco "**se**" aos blocos apresentados na Figura 12 como mostrado na Figura 13.

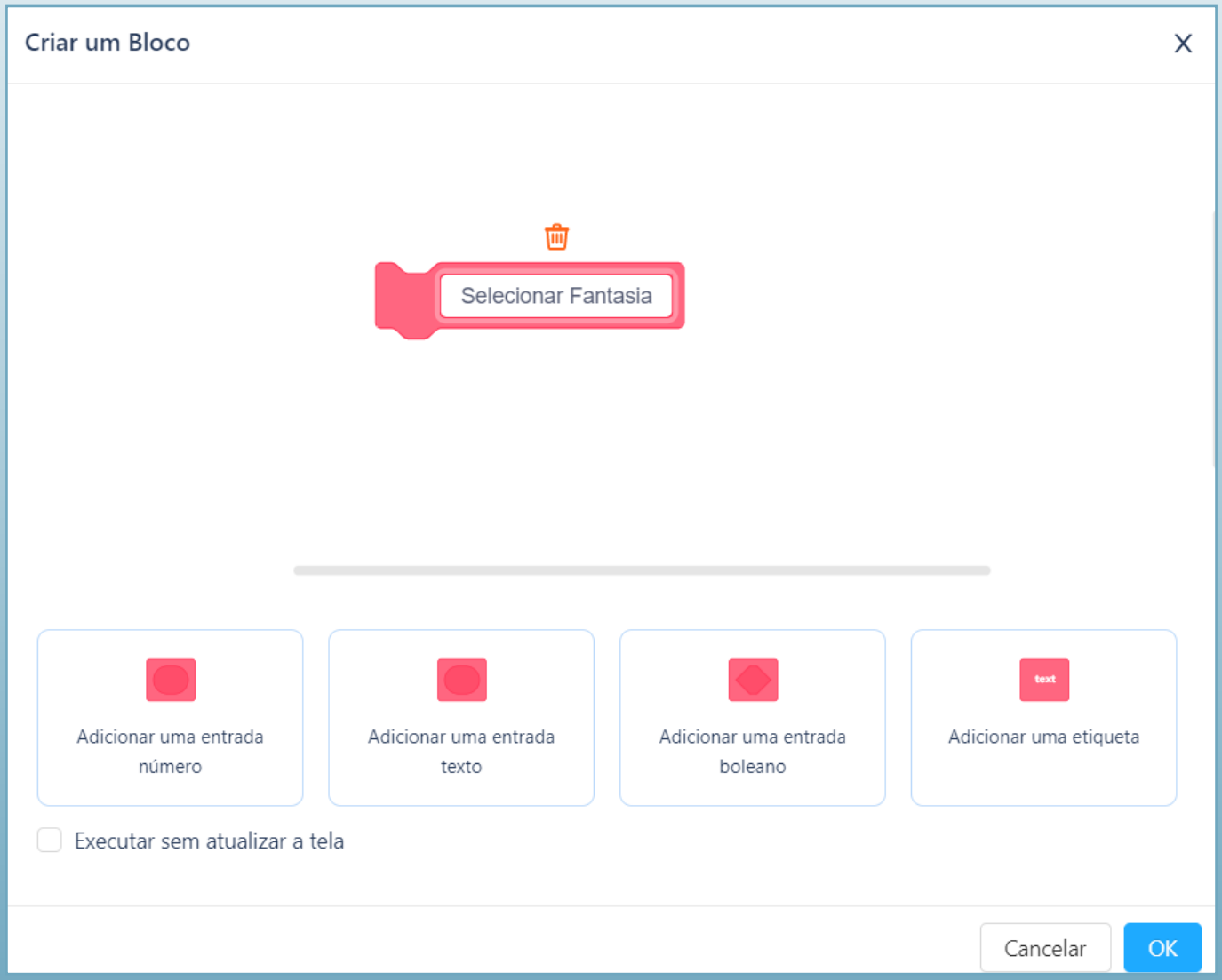
AULA 13

Jogo do pinhão - II

Com os passos até então descritos, **já é possível jogar interagindo com os pinhões e com as águias**, no entanto, para deixá-la mais bonita, colocaremos uma **animação** na águia.

A ideia dessa animação é semelhante à utilizada na "**gralha**", mas como aqui temos mais sprites do que antes, necessitamos de uma lógica mais refinada. Inicialmente, utilizaremos a variável "**modo_vooAguia**" para saber qual sprite apresentar no momento. Para isso, vá à opção "**Os Meus Blocos**" e crie um bloco com o nome "**Selecionar Fantasia**" como mostrado na Figura 14.

Figura 14 - Criação do bloco "Selecionar Fantasia".



Jogo do pinhão - II

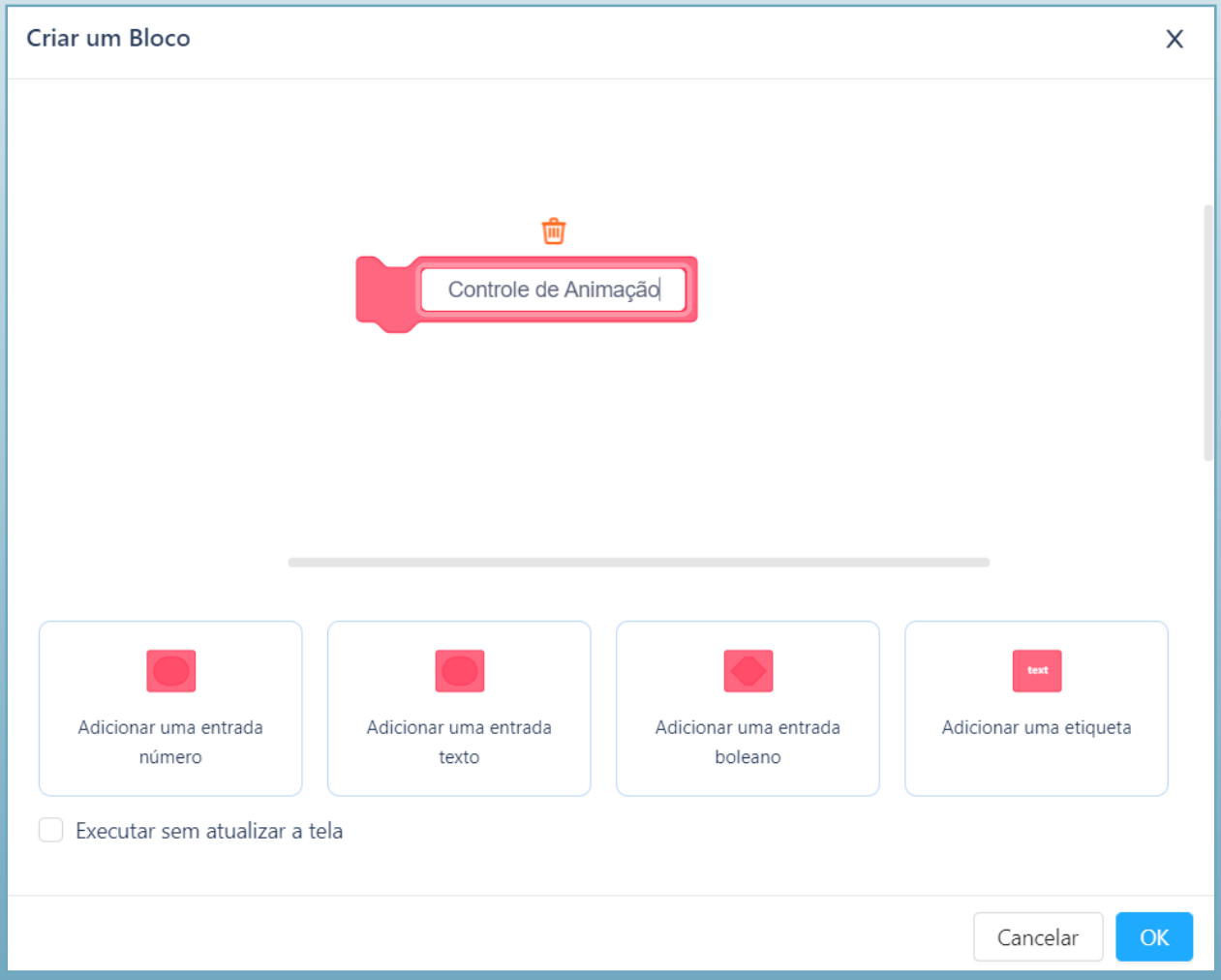
Figura 15 - Implementação da lógica no bloco "Selecionar Fantasia".

Com nosso bloco de selecionar fantasias criado, vamos implementar a lógica descrita anteriormente utilizando a variável "**modo_vooAguia**", para isso, utilizaremos blocos "**se**" seguidos um do outro implementados dentro do bloco "**Selecionar Fantasia**" como mostrado na Figura 15.

Jogo do pinhão - II

Para alterarmos o valor da variável "**modo_vooAguia**" e, consequentemente, mudar a fantasia da "**Aguia**" gerando a animação, utilizaremos as variáveis "**indo**" e "**contadorAguia**" (deve ser criada agora). A lógica aqui é similar à utilizada na "**gralha**", mas como temos mais fantasias para a "**Aguia**", devemos sair da primeira fantasia e ir até a última e depois voltar da última até a primeira enquanto o jogo se mantém executando. Para isso, utilizamos o "**contadorAguia**" que vai alterar o valor de "**modo_vooAguia**" em "**+1**" se a variável "**indo**" for igual a **1**, ou em "**-1**" se a variável "**indo**" for igual a **0**. Para implementar essa lógica, devemos primeiramente criar outro bloco. Repita os passos mencionados anteriormente e crie o bloco "**Controle de Animação**" como mostrado na Figura 16.

Figura 16 - Criação do bloco "Controle e Animação".



Jogo do pinhão - II

Figura 17 - Atualização da lógica do "Controle de Animação".

Agora, para aplicar a lógica explicada anteriormente, faça as alterações no bloco "**Controle e Animação**" apresentadas na Figura 17.

Jogo do pinhão - II

Figura 18 - Adição dos blocos criados ao clone da águia.

Agora, adicione os blocos que criamos no bloco "**Quando eu Iniciar como clone**" da águia, como mostrado na Figura 18.

Jogo do pinhão - II

Figura 19. Inserção do bloco de Parada e alteração da velocidade da águia.

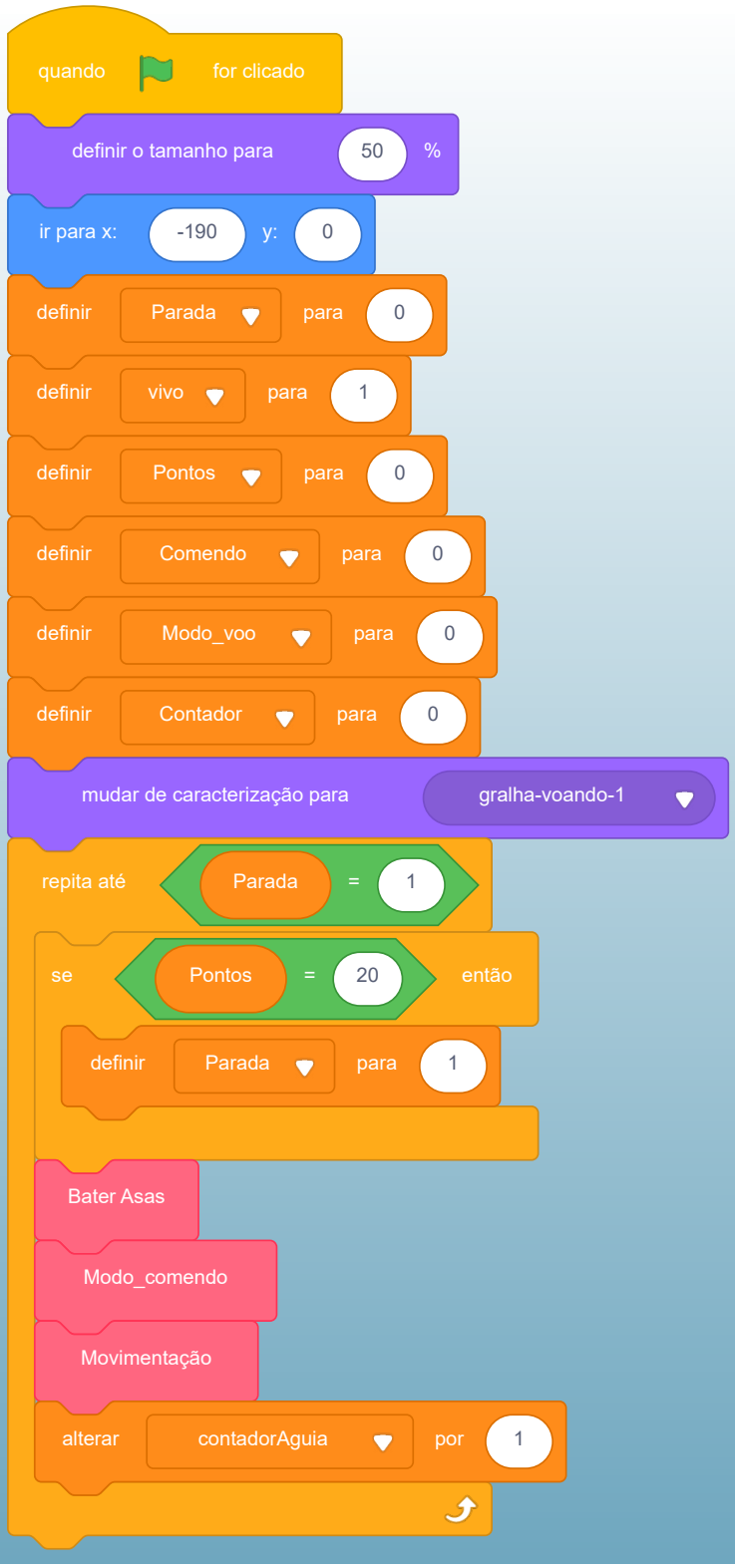
Vamos aproveitar que estamos fazendo alterações nos blocos desse grupo "quando eu iniciar como clone" e vamos alterar a velocidade da águia. Para isso, altere o valor em "alterar x em" para –5.

Também vamos criar uma variável chamada "Parada" para que quando a águia toque a gralha, o jogo seja interrompido e finalizado. Faça as alterações conforme a figura 19:

Fonte: os autores, 2024.

Jogo do pinhão - II

Figura 20 - Inclusão da atualização da variável "contadorAguia" e outras alterações.



Para que nosso jogo funcione da forma que esperamos, é necessário atualizar o valor de contador, para isso vá ao ator da gralha e, após o bloco Movimentação, adicione o bloco que altera o valor de "**contadorAguia**" em **1** como apresentado na Figura 20.

Aproveite para alterar o valor da gralha para 50% ou outro valor que seja do seu agrado e também altere o valor do bloco azul onde colocamos a gralha mais à esquerda da tela x: - 190. Como estamos utilizando o bloco personalizado chamado "Movimentação", não são mais necessários os blocos de seta para cima alterando o valor do y para dar o movimento da gralha. Você poderá apagar de acordo com a figura ao lado.

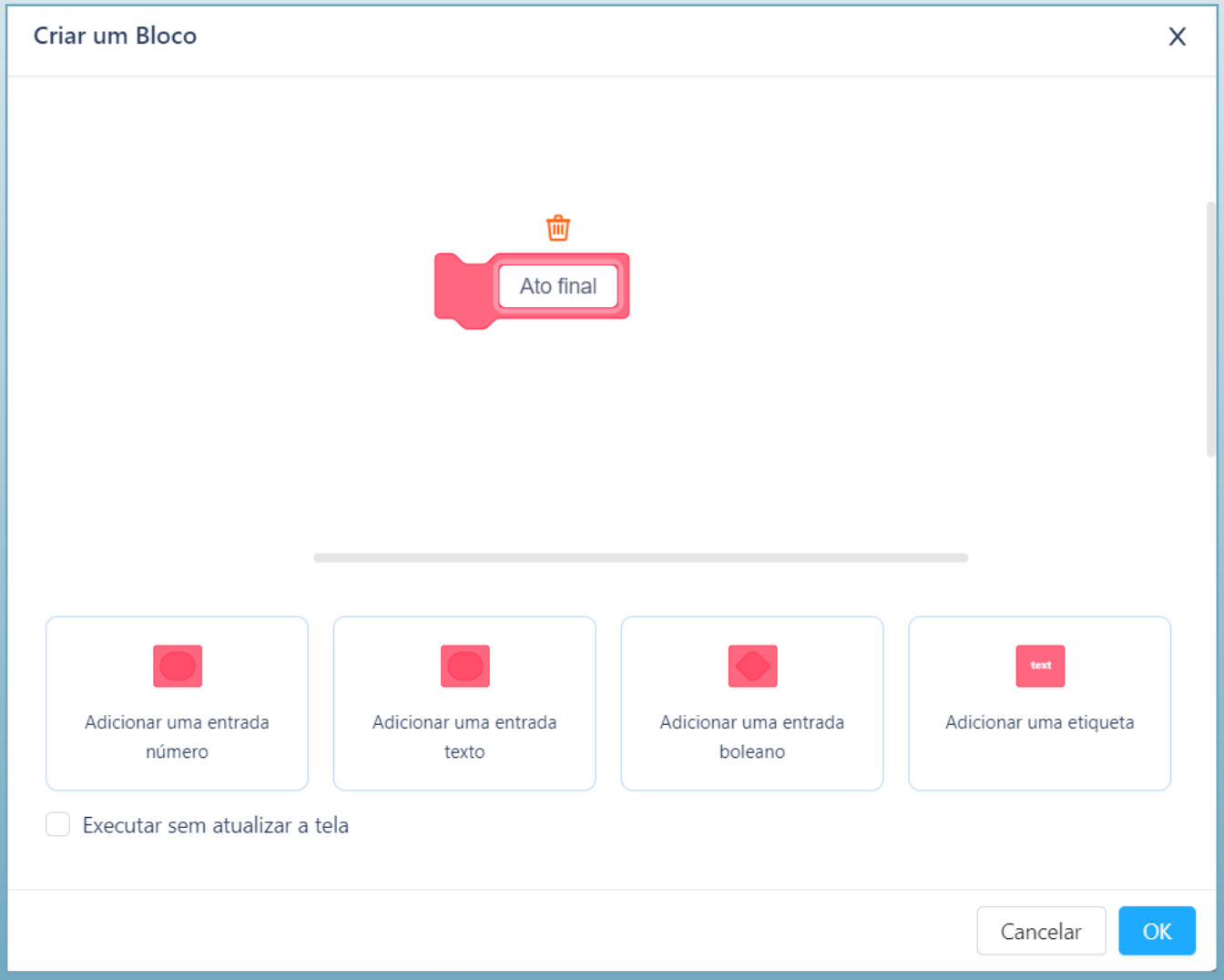
Vamos aproveitar também para inserir a variável "**Parada**" dentro do loop configurando o término do jogo caso a gralha seja atingida pela águia. Altere o final do código onde refere-se aos pontos = 20 numa estrutura de "**repita até**" e troque para um bloco de "se/então". Faça as alterações conforme a figura 20.

Jogo do pinhão - II

O motivo para adicionarmos a alteração do valor do "**contadorAguia**" nos blocos da "**gralha**" é que se adicionássemos esse bloco nos blocos dos clones da "**Aguia**" quando tivéssemos mais de um clone na tela do jogo, as animações seriam alternadas de uma forma inesperada.

Por fim, vamos executar a animação de derrota da "**gralha**" caso ela seja tocada por alguma "**Aguia**", para isso, faça o upload da imagem "**gralha-morrendo.png**" como fantasia para nosso ator "**gralha**". Com a nova fantasia adicionada, crie o bloco "**Ato final**", nesse bloco estará o que a "**gralha**" deve fazer caso ela vença ou perca o jogo. A criação desse bloco é apresentada na Figura 20.

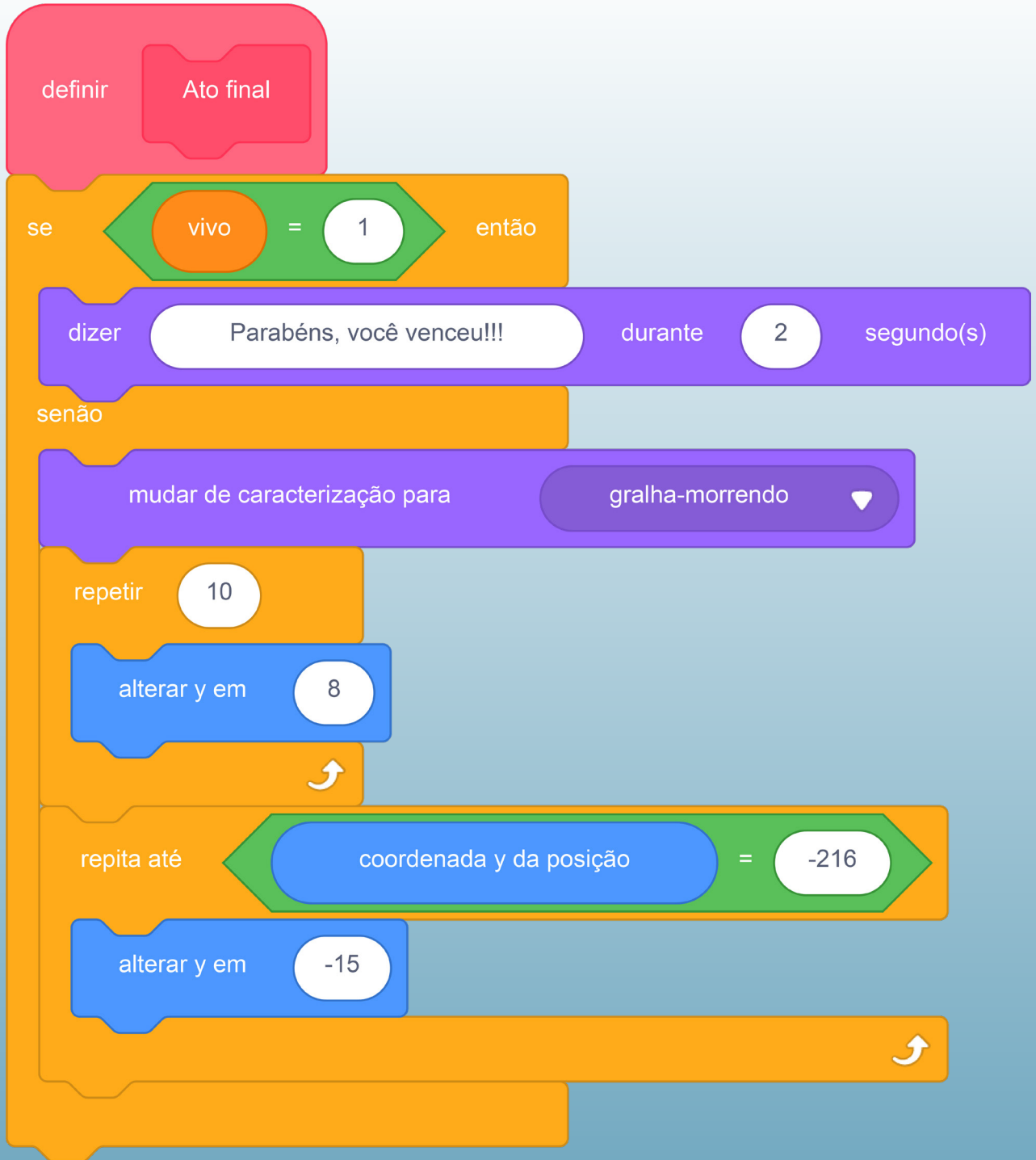
Figura 21 - Criação do bloco "Ato final".



Jogo do pinhão - II

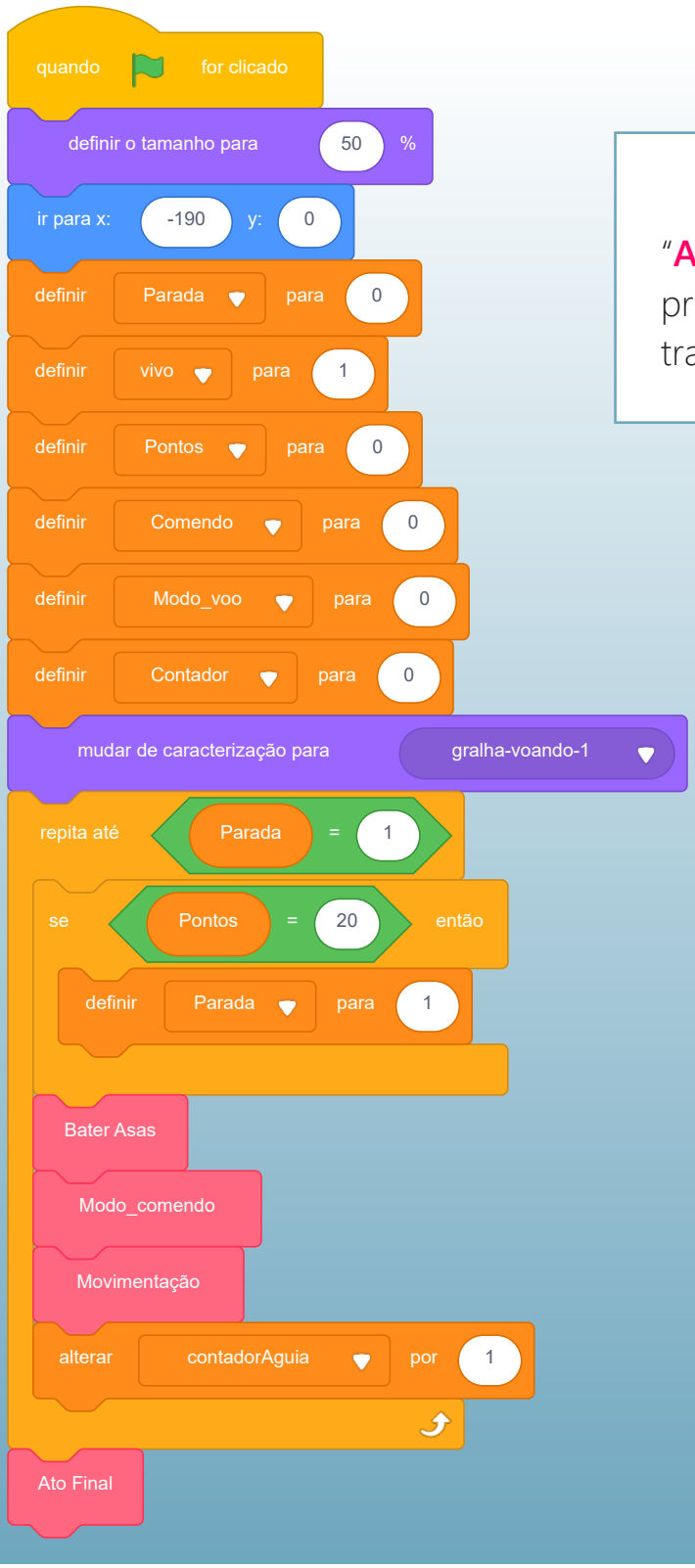
Agora, atualize o bloco "**definir Ato final**" como apresentado na Figura 22.

Figura 22 - Adição da lógica no "Ato final".



Jogo do pinhão - II

Figura 23 - Adição do "Ato final" à "gralha".

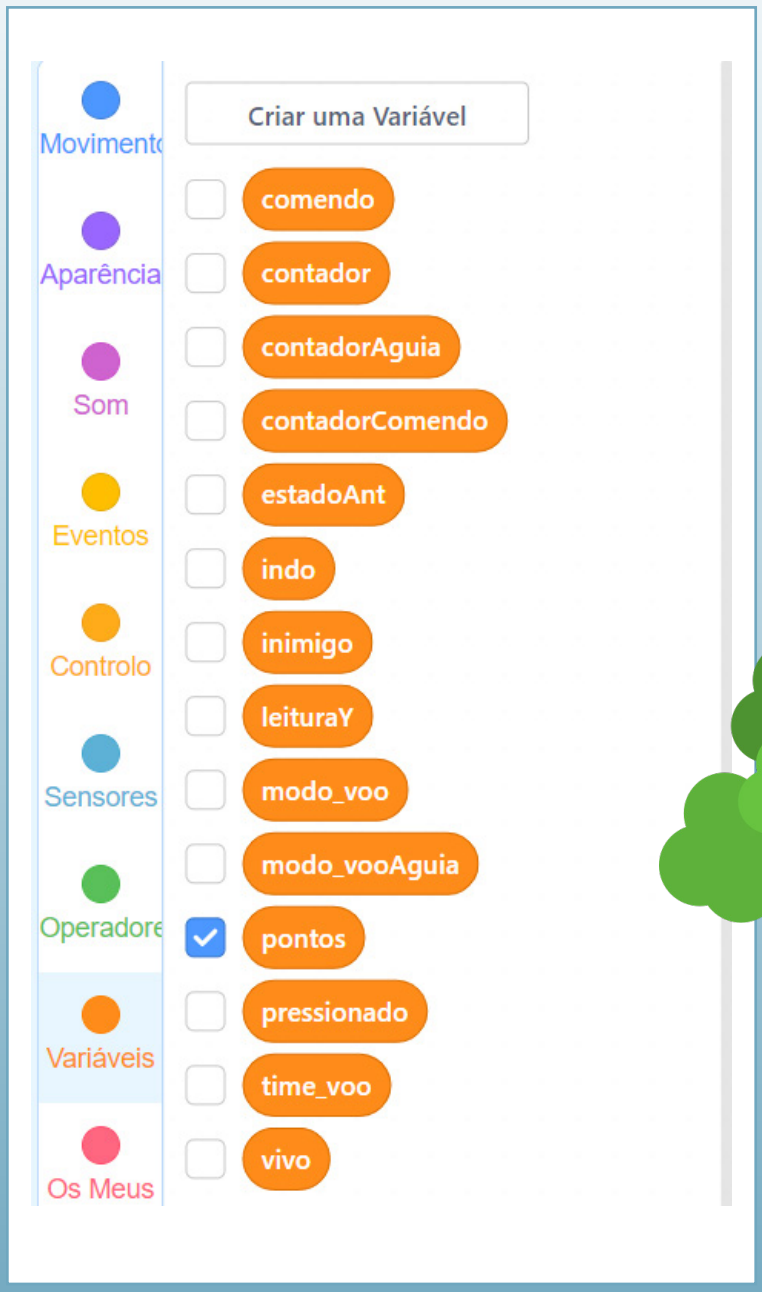


Por fim, basta adicionar o bloco "**Ato final**" ao conjunto de blocos principais da "**gralha**" como mostrado na Figura 23.

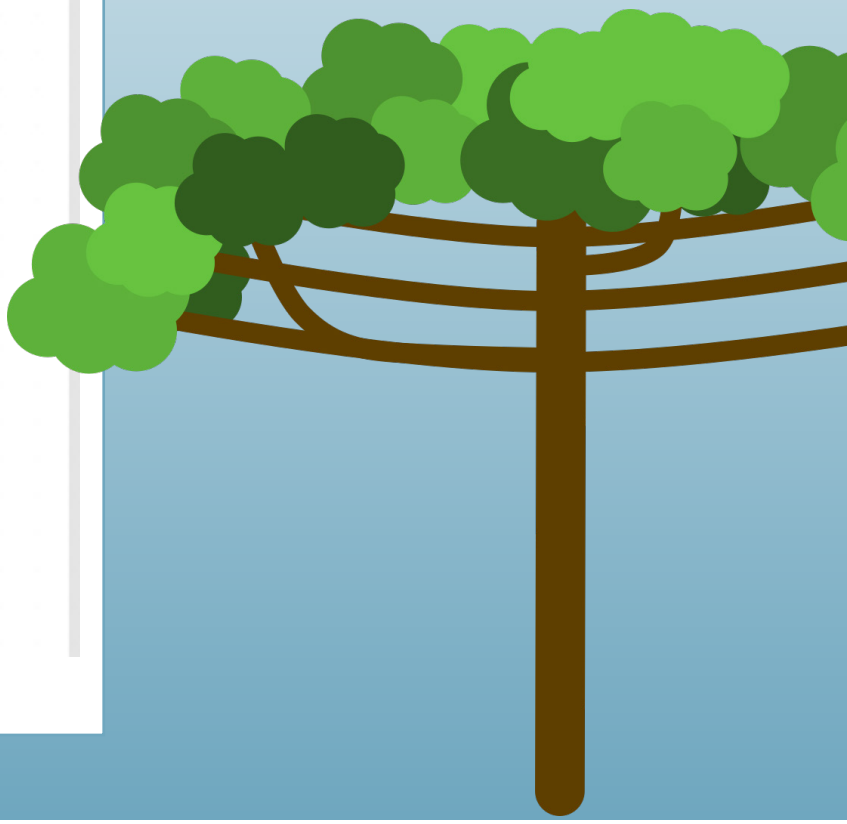
Jogo do pinhão - II

Clicando na opção "Variáveis" para visualização de todas as variáveis que criamos ao longo do projeto, desmarque a caixa de seleção de todas, exceto a da variável "**pontos**" como mostrado na Figura 24.

Figura 24 - Selecionando "pontos" para verificação do seu valor.



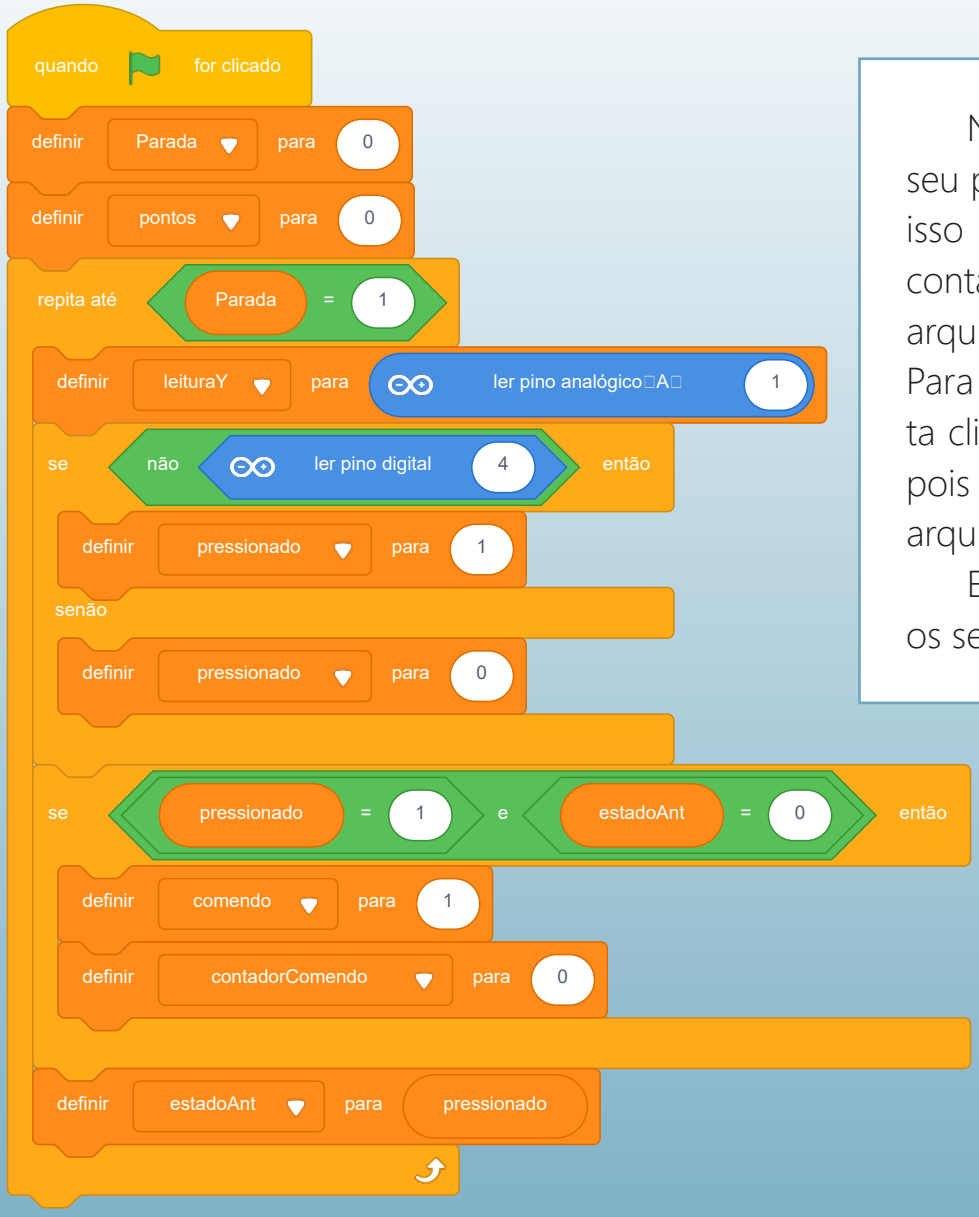
Fonte: mBlock, 2024 (adaptado).



Jogo do pinhão - II

Para finalizar, vamos nos voltar para a aba "Dispositivos" onde temos o "Arduino". Uma vez que foi criada a variável "Parada" para sinalizar o término do jogo, teremos que fazer algumas alterações no código. Segue o código alterado.

Figura 25. Código do dispositivo Arduino.



Fonte: os autores, 2025.

Não esqueça de **salvar** o seu projeto. Você pode fazer isso **on-line** através da sua conta no mBlock ou salvar o arquivo local no computador. Para carregar o arquivo, basta clicar em "**Ficheiro**" e depois em "**Abrir**" e localizar o arquivo no computador.

Explore o jogo e convide os seus amigos para jogar!

Jogo do pinhão - II

Desafio

Que tal você incluir alguns sons no jogo? Pense em fundo musical e sons que os personagens poderão exibir. Explore a ferramenta de sons do mBlock e busque na internet por repositórios musicais que tenham sons e músicas de licença aberta para o uso não comercial. Tente incluir ao código, teste e compartilhe suas alterações com os colegas.

3. Feedback e Finalização:

O seu jogo agora está funcionando definitivamente! Ele possui personagem principal, inimigo, objetivo e até mesmo pontuação. A respeito do jogo:

- Que tipo de melhoria você faria no personagem principal? Discuta com os seus colegas.
- Que tipo de desafio você incluiria no jogo para torná-lo mais complexo?

Após esses passos, reflita:

- Ao pensar em melhorias para o personagem principal, qual foi a sua inspiração?
- Ao imaginar as melhorias que o inimigo poderia ter, qual foi a sua inspiração?
- Na sua opinião, qual é o melhor tipo de jogo: um que tenha personagem principal, inimigo e objetivo a ser cumprido ou outro que tenha mais de um personagem que jogam juntos para cumprir um objetivo sem haver algum conflito com um inimigo?
- Um jogo pode ser uma ferramenta de aprendizado? Você se recorda de ter utilizado algum jogo eletrônico e esse estar relacionado com algum conteúdo que tenha visto em sala de aula?

Referências

MCROBERTS, Michael. **Arduino básico.** São Paulo, SP: Novatec, c2011. 453 p. ISBN 9788575222744.

WIKIAVES. **Cyanocorax**. Disponível em: https://www.wikiaves.com.br/wiki/cyanocorax. Acesso em: 07 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

- Arthur Henrique Andrade Farias - Ciência da Computação
- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Jenniffer Oliveira Checchia - Ciência da Computação
- Leonardo Vargas de Paula - Sistemas de Informação
- Marcos Gabriel da Silva Rocha - Engenharia de Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Nathanael Martins Wink - Ciência da Computação
- Victor Luiz Marques Saldanha Rodrigues - Ciência da Computação

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)

COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva

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.