

# Raquete do oponente e placar

## Unidade

### Lógica de programação: desenvolvendo um jogo estilo Pong

## Questão 1 – Pontuação

Para desenvolver um algoritmo de pontuação do jogo Pong, é necessário avaliar se a bolinha realmente passa sem que seja rebatida pela raquete. Para descrever essa situação em código, qual é o principal valor de referência que devemos avaliar para gerar essa pontuação?

- ☐ a) Devemos avaliar se a bolinha passa pela raquete ao verificar se a posição y da bolinha está fora do alcance da raquete.
- ☐ b) Verificar se a coordenada x da bolinha ultrapassa as coordenadas x das raquetes dos jogadores para determinar a pontuação.
- ☐ c) Verificar a velocidade da bolinha para determinar se ela foi rebatida pela raquete.

**Alternativa A**, incorreta. Para rebater a bolinha, a coordenada é importante, porém, não é por meio dela que desenvolvemos o algoritmo de pontuação.

**Alternativa B**, correta. Lembre-se: sempre que, com coordenadas negativas, a referência muda, por essa razão, precisamos alterar as condições lógicas de maior (>) para valores de posição X maior que zero (no caso, maior que 229), e menor (<) para valores de posição X menor do que zero (no caso, -229).

**Alternativa C**, incorreta. Embora a velocidade da bolinha possa mudar ao ser rebatida pela raquete, esse não é o principal valor de referência para determinar se a bolinha passou sem ser rebatida.

## Questão 2 – Sistema de vidas

Vamos adicionar um oponente final no nosso jogo do Scratch? Para fazer isso, você deve seguir algumas instruções:

- Esse oponente deve possuir uma barra de vida e precisa ser atingido várias vezes para ser derrotado;
- Para confrontar o oponente final, a pessoa jogadora deve usar uma lança que pegou no chão.

Como estruturar esse código dentro do Scratch? Ordene **todos os blocos** a seguir.

se <<tocando em [oponente]> e <tocando em [lança]>>> então

mude [vida] para [100]

quando bandeira for clicada

adiciona [-10] a [vida]

sempre

Escreva a sequência correta de blocos nas linhas a seguir:

---

---

**Sequência correta:** quando bandeira for clicada | mude [vida] para [100] | sempre | se <<tocando em [oponente]> e <tocando em [lança]>>> | então | adiciona [-10] a [vida]

**Comentário:** ao iniciar o jogo com o clique na bandeira, você configurou o oponente final com uma barra de vida de 100. Com a condição de verificar se a personagem está tocando tanto no oponente quanto na lança, você assegurou que apenas o contato com a arma possa reduzir a vida do oponente.

## Questão 3 – Contando animais

Amanda está fazendo um jogo de fazendinha no Scratch, cujo objetivo é contar os animais. Ela colocou sensores para saber quando um animal entra e fez uma cerca que aparece na tela no ponto X = 228. Cada vez que um animal passa por lá, o sensor soma um ponto na contagem.

Ajude Amanda a montar esse código ordenando **todos os blocos** a seguir para que ele conte o número de animais que entraram na fazenda.

adiciona (1) a <minha pontuação>

quando bandeira verde for clicada

para sempre

espere até que <posição x> > <228>

Escreva a sequência correta de blocos nas linhas a seguir:

---

---

**Sequência correta:** quando bandeira verde for clicada | para sempre | espere até que <posição x> > <228> | adiciona (1) a <minha pontuação>

**Comentário:** lembre-se de que o sensor precisa reconhecer quando um animal passa pela cerca virtual. Certifique-se de usar um bloco que detecte essa passagem e outro que some um ponto à contagem cada vez que isso acontecer.