

Aula 8

Quente frio dinâmico

► Unidade

Lógica de programação: criando arte interativa com p5.js

Questão 1 – Identificação de conceitos e comandos

Associe os conceitos da coluna A aos termos corretos da coluna B:

Coluna A

- a) **constrain()**
- b) **dist()**
- c) **random()**

Coluna B

1. Limita o valor dentro de um intervalo.
2. Calcula a distância entre dois pontos.
3. Gera um valor aleatório dentro de um intervalo.

Em seguida, assinale a alternativa correta:

- a)** a-1, b-2, c-3
- b)** a-2, b-3, c-1
- c)** a-3, b-1, c-2
- d)** a-1, b-3, c-2

Alternativa A, correta. O comando `constrain()` é usado para limitar um valor dentro de um intervalo. O comando `dist()` calcula a distância entre dois pontos no espaço. Já o comando `random()` gera valores aleatórios dentro de um intervalo especificado.

Alternativa B, incorreta. A função `random()` gera números aleatórios dentro de um intervalo, não os limita.

Alternativa C, incorreta. A função `constrain()` não é utilizada para gerar valores aleatórios. A função `constrain()`, na verdade, é usada para limitar valores dentro de um intervalo.

Alternativa D, correta. A função `dist()` não é responsável pela geração de valores aleatórios.

Questão 2 – Ajuste da dificuldade em um jogo interativo

No código a seguir, a dificuldade do jogo depende da variável distância, que calcula a distância entre o mouse e o objeto. Para tornar o jogo mais difícil, qual modificação você pode fazer no valor de distância?

```
let distancia = dist(mouseX, mouseY, x, y);
if (distancia < 3) {
  text("Encontrei!", 200, 200);
  noLoop();
}
```

Proponha a modificação para aumentar a dificuldade do jogo:

- a)** Diminuir o valor `distancia < 3` para `distancia < 1`.
- b)** Aumentar o valor `distancia < 3` para `distancia < 5`.
- c)** Remover a função `dist()` e utilizar valores fixos para `distancia`.
- d)** Alterar o limite de `random(-5,5)` para `random(-10,10)`.

Alternativa A, correta. Diminuir o valor para `distancia < 1` aumentaria a dificuldade, pois o jogador teria que estar ainda mais próximo do objeto para “vencê-lo”.

Alternativa B, incorreta. Aumentar o valor para `distancia < 5` facilitaria o jogo, pois o jogador não precisaria estar tão perto do objeto para ganhar.

Alternativa C, incorreta. Remover a função `dist()` eliminaria o cálculo da distância, prejudicando a funcionalidade principal do jogo.

Alternativa D, incorreta. Aumentar o intervalo do `random()` tornaria o movimento do objeto mais imprevisível, mas isso não está diretamente relacionado com o ajuste da dificuldade baseado na distância.

Questão 3 – Comportamento do comando constrain()

No código a seguir, os valores de x e y são atualizados por meio do comando **random()**.

```
x = x + random(-20, 20);
y = y + random(-20, 20);
x = constrain(x, 0, 400);
y = constrain(y, 0, 400);
```

Avalie se as afirmativas a seguir são verdadeiras (V) ou falsas (F) sobre o comportamento do código.

- () O comando **random(-20, 20)** pode fazer com que x e y assumam valores negativos ou maiores que 400.
- () O comando **constrain()** é usado para garantir que x e y permaneçam dentro do intervalo de 0 a 400, corrigindo valores que possam estar fora desse intervalo.
- () Se os valores de x e y estiverem dentro dos limites de 0 a 400, o comando **constrain()** não altera os valores.
- () Sem o comando **constrain()**, os valores de x e y nunca ultrapassariam os limites de 0 a 400 devido à função **random(-20, 20)**.

Escreva a sequência correta de letras nas linhas a seguir:

Sequência correta: V | V | V | F

Comentário: este exercício reforça o entendimento do comportamento dos comandos *random()* e *constrain()*. Ele ajuda o estudante a perceber a importância de *constrain()* para limitar valores dentro de um intervalo específico, especialmente quando os valores são alterados de maneira aleatória com *random()*.