

Lógica de Programação: laços e listas com JavaScript

◆ Questão 1 (JavaScript)

Uma pessoa, após incluir a imagem da estrada como background, do carro e do personagem do jogo, adicionou o seguinte código para mover o personagem, quando as setas para cima e para baixo forem pressionadas:

```
function movimentaAtor(){  
  if (keyIsDown(UP_ARROW)){  
    yAtor = yAtor + 3;  
  }  
  if (keyIsDown(DOWN_ARROW)){  
    yAtor = yAtor - 3;  
  }  
}
```

Analisando o código acima, podemos afirmar que:

Respostas:

- Quando a tecla seta para cima for pressionada, o personagem irá para baixo.
- Quando a tecla seta para baixo for pressionada, o personagem irá para cima.

◆ Questão 2 (JavaScript)

Para fazer com que os carros voltem a aparecer na tela e passem a rodovia mais uma vez, uma pessoa desenvolveu o seguinte código:

```
function voltaPosicaoInicialDoCarro(){  
  if (xCarro < 50){  
    xCarro = 600  
  }  
  if (xCarro2 < 50){  
    xCarro2 = 600  
  }  
  if (xCarro3 < 50){  
    xCarro3 = 600  
  }  
}
```

Porém algo não saiu como a pessoa queria. Com base nessas informações podemos afirmar que:

Resposta: Os carros desaparecerão subitamente antes de ultrapassarem a extremidade esquerda da tela.

Orientação da Alura: Com base no código acima, se o valor da variável xCarro for menor que 50, ele volta para a posição inicial. O valor correto seria -50.

◆ Questão 3 (JavaScript)

Para evitar uma grande quantidade de variáveis para cada carro, criamos listas para armazenar essas informações. Sabendo que trata-se de uma estrutura de dados ordenados, observe a seguinte lista:

```
let palavras = ["Dia", "programar", "javascript", "de"]
```

Com base no exemplo acima, analise as afirmações abaixo e marque as verdadeiras:

Respostas:

- Para descobrir a quantidade de elementos de uma lista, podemos usar a palavra length.
- Podemos recuperar os valores dentro de uma lista através de índice.
- Podemos formar uma frase com o seguinte código:
console.log(palavras[0],palavras[3],palavras[1],palavras[2])

◆ Questão 4 (JavaScript)

Uma pessoa resolveu incluir o placar no seu jogo e criou o seguinte código:

```
function marcaPonto(){  
  if (yAtor < 15){  
    meusPontos += 1;  
    voltaAtorParaPosicaoInicial();  
  }  
}
```

Ao executar o jogo e chegar do outro lado, a pessoa não marcou nenhum ponto. Não encontrando o erro, decidiu compartilhar seu arquivo sketch.js, conforme ilustra o código abaixo:

```
function setup() {  
  createCanvas(500, 400);  
}  
  
function draw() {  
  background(imagemDaEstrada);  
  mostraAtor();  
  mostraCarro();  
  movimentaCarro();  
  movimentaAtor();  
  voltaPosicaoInicialDoCarro();  
  verificaColisao();  
  incluiPontos();  
}
```

Analisando os códigos acima, por que a pessoa não está conseguindo marcar pontos quando consegue atravessar a rua e chegar do outro lado?

Resposta: A pessoa não está chamando a função `marcaPonto()` em `draw()`, assim ela não será executada.

Orientação da Alura: Para conseguir pontuar, é necessário executar função `marcaPonto()`, dentro da função `draw()`.

◆ Questão 5 (JavaScript)

Para melhorar a leitura e compreensão do código, alteramos o nomes de algumas funções como colidiu para voltaAtorPosicaoInicial() e criamos a função pontosMaiorQueZero(), conforme as imagens abaixo:

```
function voltaAtorParaPosicaoInicial(){  
  yAtor = 366;  
}  
  
function pontosMaiorQueZero(){  
  return meusPontos > 0;  
}
```

Analisando as funções acima, podemos afirmar que:

Respostas:

- Para criar uma função que retorna um valor em Javascript, usamos a palavra return, seguida pelo valor que queremos retornar.
- Uma função em Javascript pode ter um retorno ou não.
- O retorno da função pontosMaiorQueZero() será true ou false.