### **GOVERNO DO ESTADO DO ESPÍRITO SANTO**

### **SECRETARIA ESTADUAL DE EDUCAÇÃO**

### **ENSINO MÉDIO TÉCNICO INTEGRADO INTEGRAL**

### **TURNO VESPERTINO INTERMEDIÁRIO – ANO LETIVO 2023**

### 

### **Valor: 5 pontos. - Nota obtida: \_\_\_\_\_ pontos. - Data: \_\_\_/06/24 – 2º Trimestre.**

### **ESTUDANTE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ SÉRIE: 3 INFO 1**

### 

### 

### **Teoria sobre Colisão**

#### **1. O que é Colisão?**

Colisão é o evento que ocorre quando dois ou mais objetos ocupam o mesmo espaço ao mesmo tempo. Em jogos, isso pode se referir a:

* A cabeça da cobrinha colidindo com sua própria cauda.
* A cobrinha colidindo com as bordas do campo de jogo.
* A cobrinha colidindo com objetos ou frutas no jogo.

#### **2. Tipos de Colisões em Jogos**

* **Colisões com Bordas:** Ocorrem quando um objeto atinge os limites do campo de jogo.
* **Colisões entre Objetos:** Ocorrem quando dois objetos no jogo se encontram. No caso do Snake, isso inclui a cabeça da cobrinha colidindo com a fruta ou com sua própria cauda.

#### **3. Técnicas de Detecção de Colisões**

* **Detecção de Colisão por Coordenadas:**
  + **Pontos:** Verifica se dois pontos ocupam a mesma posição.
  + **Retângulos:** Verifica se as áreas de dois retângulos se sobrepõem. Cada segmento da cobrinha e a fruta podem ser representados como retângulos ou pontos.

#### **4. Detecção de Colisões no Jogo da Cobrinha**

Vamos considerar a implementação em um jogo como o Snake, onde a cobrinha se move em uma grade bidimensional. A lógica básica envolve verificar se a posição da cabeça da cobra coincide com qualquer outra parte da cauda, com a borda do campo de jogo, ou com uma fruta.

**Colisões com Bordas:**

* A cobrinha colide com as bordas do campo de jogo quando suas coordenadas x ou y excedem os limites definidos.
* Por exemplo, se o campo tem uma largura (WIDTH) de 20 e uma altura (HEIGHT) de 20, uma colisão ocorre se x < 0, x >= WIDTH, y < 0, ou y >= HEIGHT.

**Colisões com a Cauda:**

* A cobrinha colide com sua própria cauda se a posição da cabeça (x, y) coincide com qualquer segmento da cauda.
* Isso pode ser verificado iterando sobre as posições armazenadas dos segmentos da cauda.

**Colisões com Frutas:**

* A cobrinha colide com uma fruta quando a posição da cabeça coincide com a posição da fruta. Nesse caso, a fruta deve ser "comida" e a cobrinha deve crescer.

#### **5. Exemplo Prático no Jogo da Cobrinha**

No exemplo de código fornecido anteriormente, a lógica de colisão pode ser descrita da seguinte forma:

**Colisão com Bordas:**  
if (x < 0 || x >= WIDTH || y < 0 || y >= HEIGHT)

gameOver = 1;

* Isso verifica se a posição x ou y da cabeça da cobra está fora dos limites permitidos.

**Colisão com a Cauda:**  
for (int i = 0; i < nTail; i++) {

if (tailX[i] == x && tailY[i] == y)

gameOver = 1;

}

* Esse trecho de código itera sobre cada segmento da cauda, verificando se a posição da cabeça coincide com qualquer segmento.

**Colisão com a Fruta:**  
if (x == fruitX && y == fruitY) {

// Código para gerar uma nova fruta e aumentar o tamanho da cauda

}

* Isso verifica se a cabeça da cobra está na mesma posição que a fruta, indicando que a fruta foi comida.

**Jogo da cobrinha**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

#define WIDTH 20

#define HEIGHT 20

#define UP 72

#define DOWN 80

#define LEFT 75

#define RIGHT 77

int gameOver, score;

int x, y, fruitX, fruitY, flag;

int tailX[100], tailY[100];

int nTail;

void setup() {

gameOver = 0;

x = WIDTH / 2;

y = HEIGHT / 2;

label1:

fruitX = rand() % 20;

if (fruitX == 0)

goto label1;

label2:

fruitY = rand() % 20;

if (fruitY == 0)

goto label2;

score = 0;

}

void draw() {

system("cls");

int i, j;

for (i = 0; i < WIDTH; i++) {

for (j = 0; j < HEIGHT; j++) {

if (i == 0 || i == WIDTH - 1 || j == 0 || j == HEIGHT - 1) {

printf("#");

} else {

if (i == x && j == y)

printf("O");

else if (i == fruitX && j == fruitY)

printf("F");

else {

int isTail = 0;

for (int k = 0; k < nTail; k++) {

if (i == tailX[k] && j == tailY[k]) {

printf("o");

isTail = 1;

}

}

if (!isTail)

printf(" ");

}

}

}

printf("\n");

}

printf("Score: %d\n", score);

}

void input() {

if (\_kbhit()) {

switch (\_getch()) {

case 'w':

flag = UP;

break;

case 's':

flag = DOWN;

break;

case 'a':

flag = LEFT;

break;

case 'd':

flag = RIGHT;

break;

case 'x':

gameOver = 1;

break;

}

}

}

void logic() {

int prevX = tailX[0];

int prevY = tailY[0];

int prev2X, prev2Y;

tailX[0] = x;

tailY[0] = y;

for (int i = 1; i < nTail; i++) {

prev2X = tailX[i];

prev2Y = tailY[i];

tailX[i] = prevX;

tailY[i] = prevY;

prevX = prev2X;

prevY = prev2Y;

}

switch (flag) {

case UP:

y--;

break;

case DOWN:

y++;

break;

case LEFT:

x--;

break;

case RIGHT:

x++;

break;

}

if (x < 0 || x > WIDTH || y < 0 || y > HEIGHT)

gameOver = 1;

for (int i = 0; i < nTail; i++) {

if (tailX[i] == x && tailY[i] == y)

gameOver = 1;

}

if (x == fruitX && y == fruitY) {

label3:

fruitX = rand() % 20;

if (fruitX == 0)

goto label3;

label4:

fruitY = rand() % 20;

if (fruitY == 0)

goto label4;

score += 10;

nTail++;

}

}

int main() {

setup();

while (!gameOver) {

draw();

input();

logic();

Sleep(100); // sleep(10) in linux

}

return 0;

}

**Questões de revisão de conteúdo**

1. O que é uma colisão em jogos e por que é importante detectá-la?
2. Quais são os tipos comuns de colisões em jogos?
3. Como a detecção de colisão por coordenadas funciona no jogo da cobrinha?
4. Por que é necessário verificar colisões entre a cabeça da cobra e sua própria cauda?
5. Como o código fornecido detecta uma colisão entre a cobra e as bordas do campo de jogo?
6. Explique a lógica usada para detectar uma colisão entre a cabeça da cobra e sua cauda.
7. Como o código detecta quando a cobra come uma fruta?
8. Qual é a função do array tailX e tailY no código?
9. Por que é necessário atualizar as coordenadas dos segmentos da cauda a cada movimento?
10. Como o código inicializa a posição da cobra e da fruta no campo de jogo?
11. Explique o propósito da função setup() no código.
12. Qual é o papel da função draw() no jogo?
13. Como a função input() contribui para a jogabilidade?
14. Por que o código usa a função \_kbhit() dentro da função input()?
15. Explique a utilização de switch na função input().
16. Qual é a função do comando Sleep(100) no código?
17. Por que é necessário incluir a biblioteca conio.h no código?
18. Como a variável nTail é usada no código?
19. Explique o propósito dos rótulos label1 e label2 no código.
20. Qual é a lógica para mover a cobra em diferentes direções?