**Snake game c programming**

#include <conio.h> #include <stdio.h> #include <stdlib.h> #include <unistd.h> int i, j, height = 20, width = 20; int gameover, score; int x, y, fruitx, fruity, flag; // Function to generate the fruit // within the boundary void setup() { gameover = 0; // Stores height and width x = height / 2; y = width / 2; label1: fruitx = rand() % 20; if (fruitx == 0) goto label1; label2: fruity = rand() % 20; if (fruity == 0) goto label2; score = 0; } // Function to draw the boundaries void draw() { system("cls"); for (i = 0; i < height; i++) { for (j = 0; j < width; j++) { if (i == 0 || i == width - 1 || j == 0 || j == height - 1) { printf("#"); } else { if (i == x && j == y) printf("0"); else if (i == fruitx && j == fruity) printf("\*"); else printf(" "); } } printf("\n"); } // Print the score after the // game ends printf("score = %d", score); printf("\n"); printf("press X to quit the game"); } // Function to take the input void input() { if (kbhit()) { switch (getch()) { case 'a': flag = 1; break; case 's': flag = 2; break; case 'd': flag = 3; break; case 'w': flag = 4; break; case 'x': gameover = 1; break; } } } // Function for the logic behind // each movement void logic() { sleep(0.01); switch (flag) { case 1: y--; break; case 2: x++; break; case 3: y++; break; case 4: x--; break; default: break; } // If the game is over if (x < 0 || x > height || y < 0 || y > width) gameover = 1; // If snake reaches the fruit // then update the score if (x == fruitx && y == fruity) { label3: fruitx = rand() % 20; if (fruitx == 0) goto label3; // After eating the above fruit // generate new fruit label4: fruity = rand() % 20; if (fruity == 0) goto label4; score += 10; } } // Driver Code void main() { int m, n; // Generate boundary setup(); // Until the game is over while (!gameover) { // Function Call draw(); input(); logic(); } }