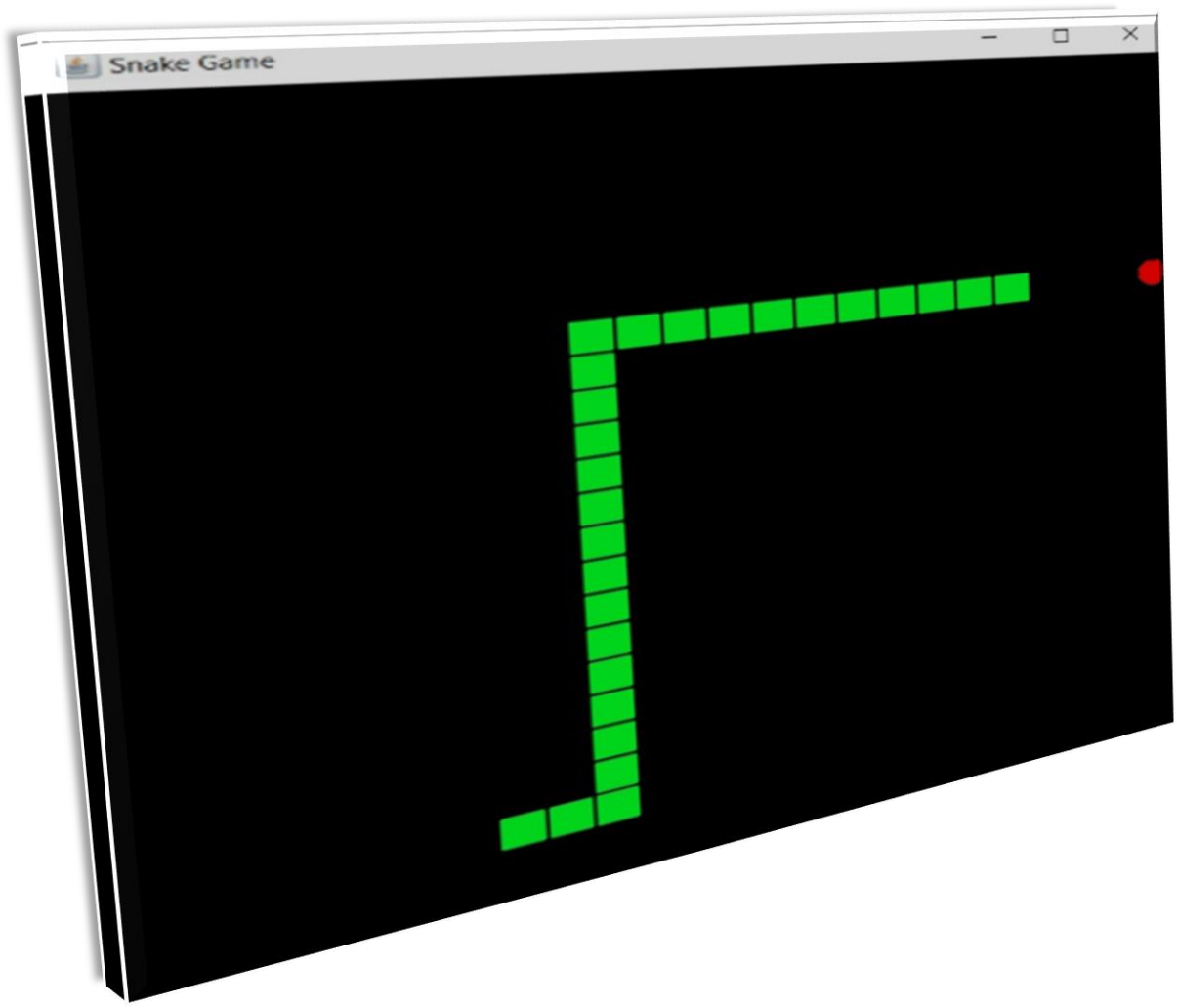
SNAKE GAME



CONTENTS

INTRODUCTION

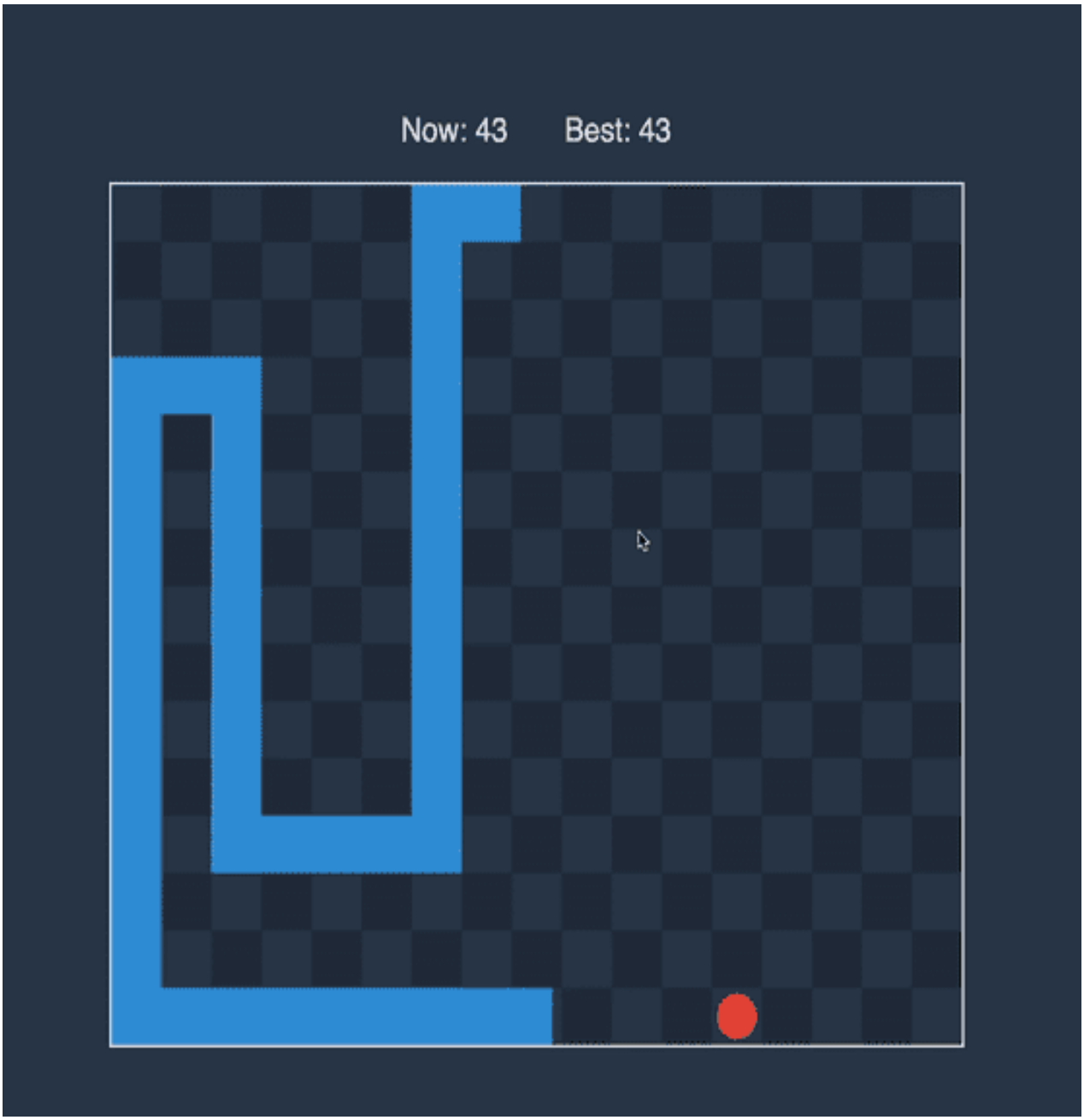
The Snake Game is a video game, in which the player maneuvers a dot or a line, which is as assumed the snake. This grows once it crosses an obstacle, which is assumed as food for the snake and the food is represented by a small dot. The food regenerates to a different location as and when the snake consumes it. Borders limit the movement of the snake, which, when simple, is a rectangle / square. If the snake hits the wall, then the game ends. The tricky bit is maneuvering the snake when it grows. At some point during the game, if the snake is in its own path, it collides with itself and the game ends.

This concept originated in 1976 in the two-player arcade game 'Blockade' from Gremlin Industries and because of the easy development of the game, hundreds of versions were made on different platforms. In 1982, an arcade game based on the movie 'Tron' featured a single player mode in the game. The turning point of the game is said to be in 1998 when a variant of the game was preloaded in all Nokia mobile phones. A resurgence of interest was seen for the game, as it was accessible to a larger number of people.

The steps to play the game are as follows:

- The Snake can be moved in the forward direction using the 'W' key and backward using 'S' key.
- To change the directions, the 'D' and 'A' keys are used to turn right and left respectively.
- The snake moves at a fixed speed
- The food for the snake appears in the form of a dot. When the snake consumes this, it grows longer.
- Once the snake grows, the food is regenerated to a different location.

- Borders restrict the movement of the snake within the screen, which, normally, is a simple square or a rectangle.
- If the snake happens to collide these borders, the game ends.
- When the snake gets long, there are chances it may come in the path of itself. When this happens, it collides with itself and the game ends.



PROBLEM STATEMENT AND OBJECTIVE

Problem statement:

Snake game is a computer action game, whose goal is to control a snake to move and collect food in a map

The traditional snake game does not offer much challenge to its players. It is impossible to bring out the best playing skill of players unless a challenge is offered to them. The players loose interest in the game after playing it for some time due to lack of challenge. Hence, when people hear about snake game, they quickly form an image of old age arcade game. The absence of powerful graphical resources and weak support for networked application in the programming language is one of the factors that has delayed the development of an interesting and challenging version of snake game.

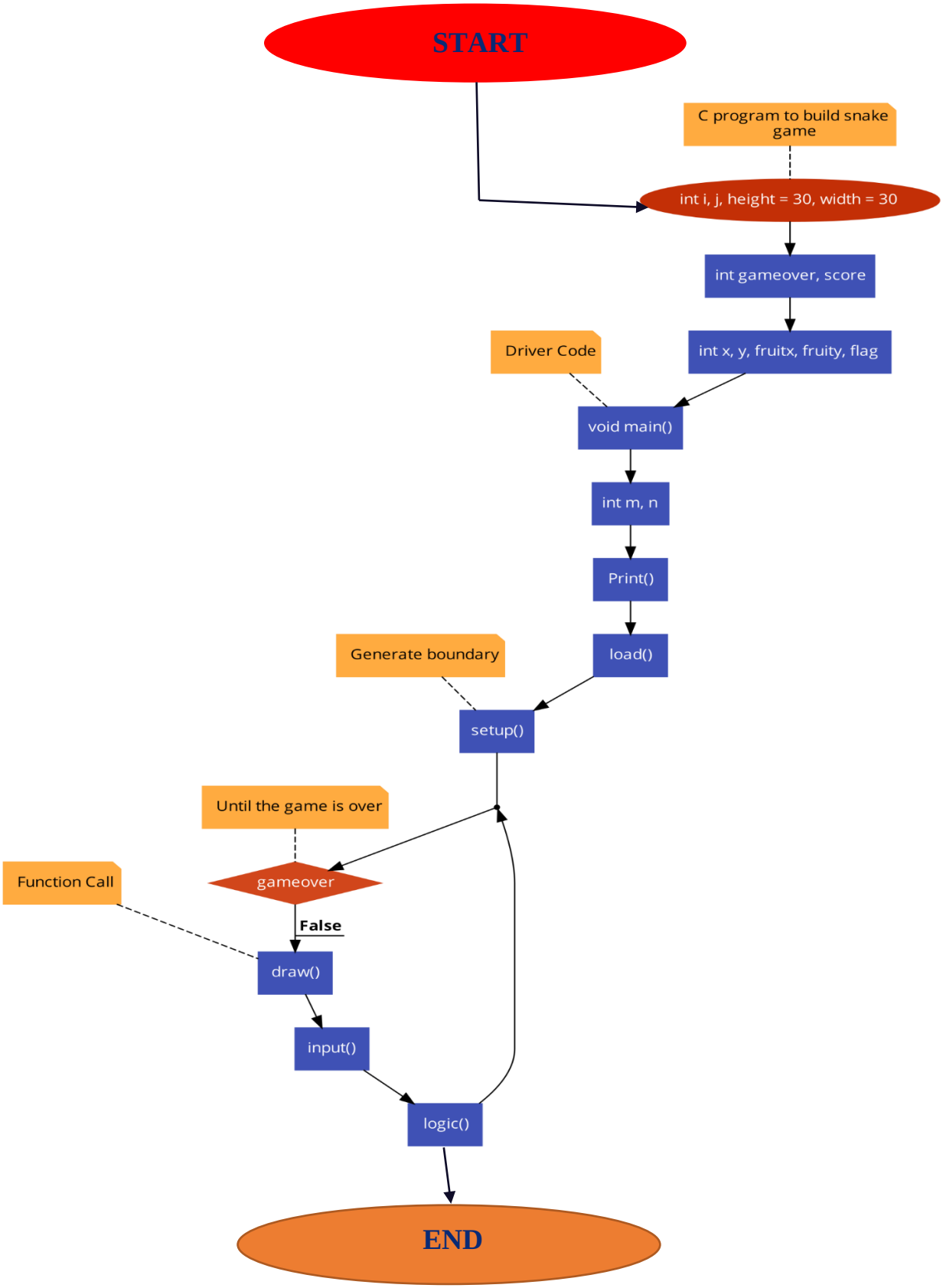
Objectives:

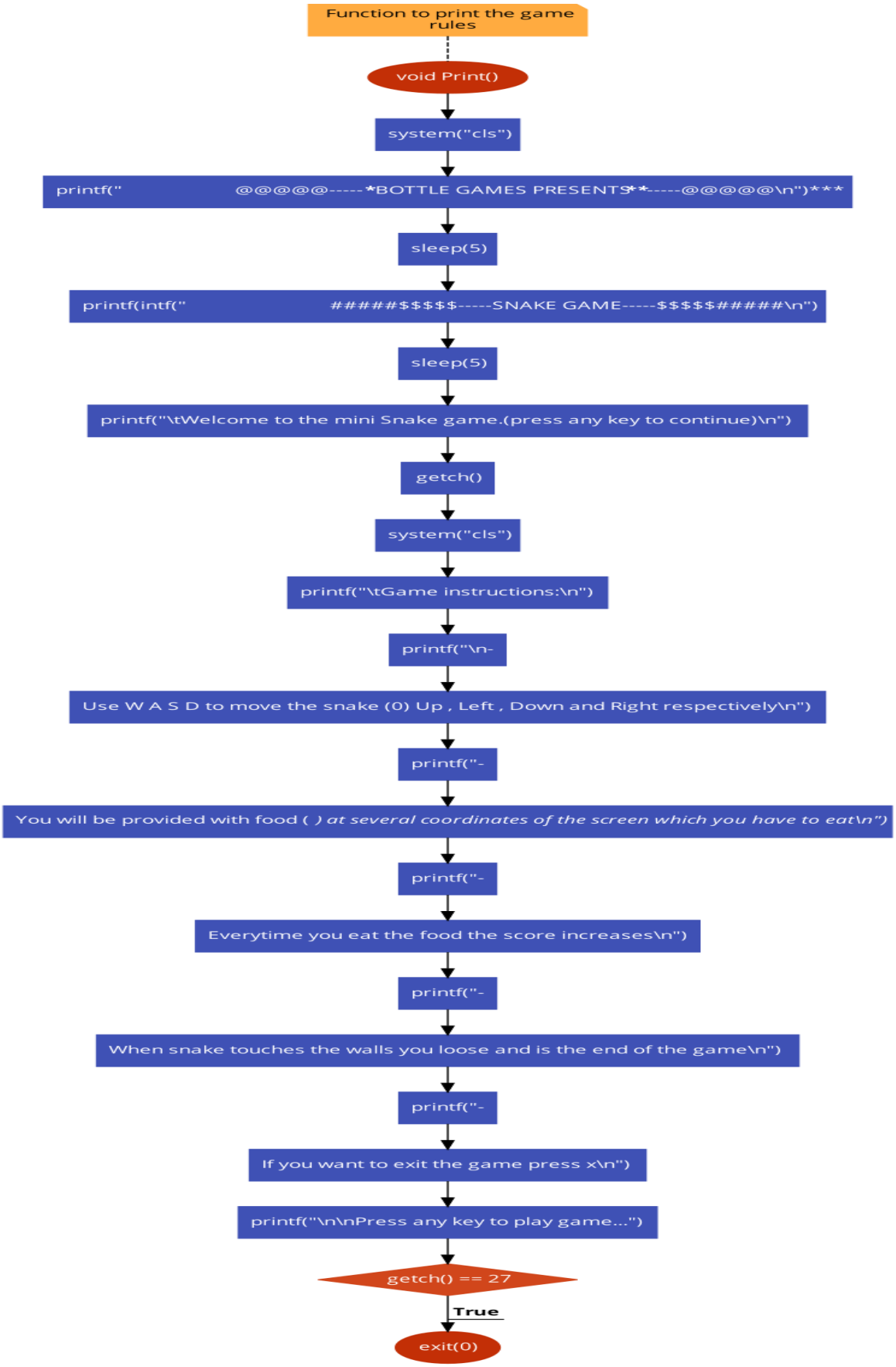
The objective is to design a snake game, which provides the following functionalities:

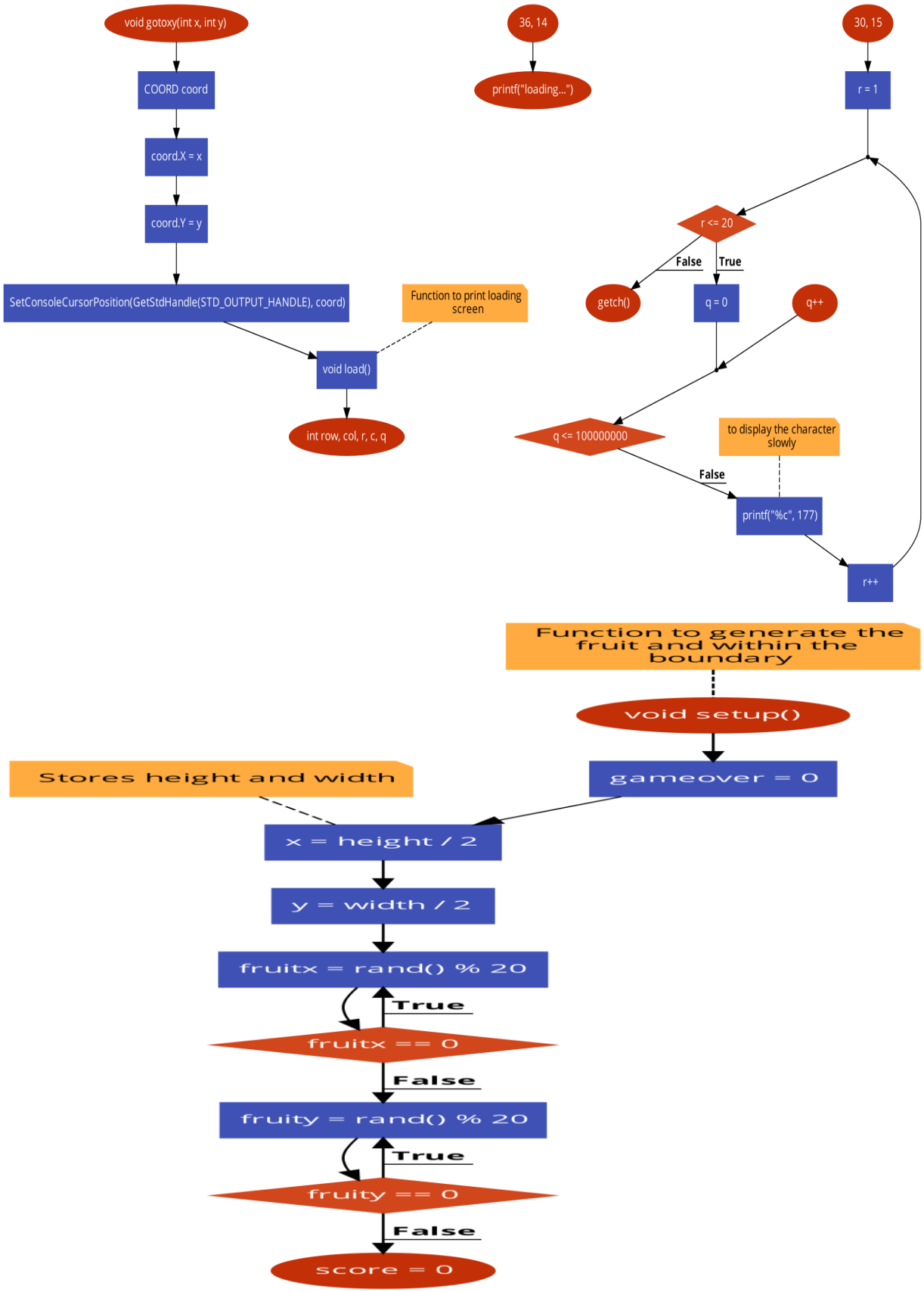
- To provide a stress-busting experience for the player
- To make the Snake game simple and more interesting for the player
- To make the game playable anytime, anywhere and from any device
- The compiler can easily execute the code

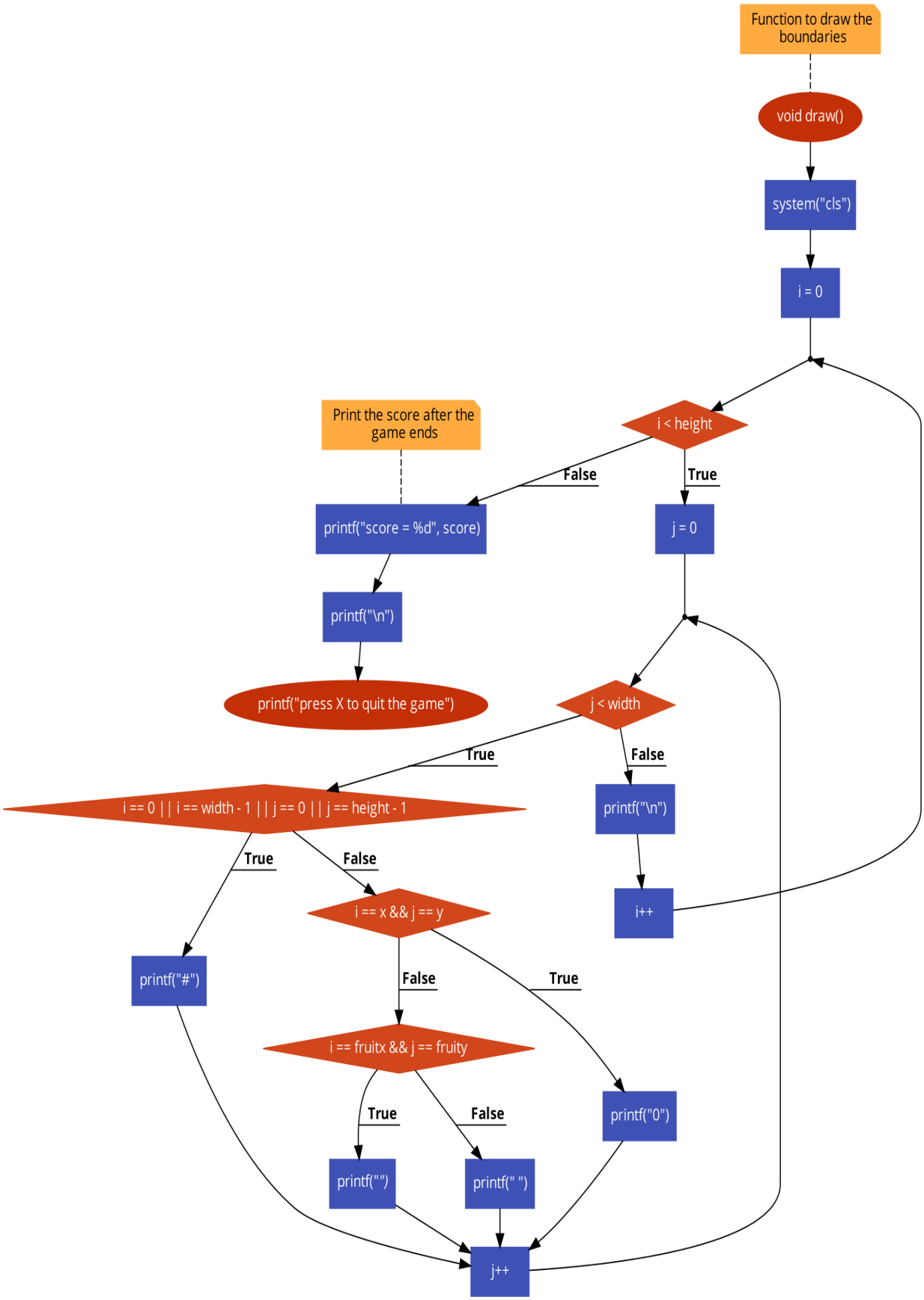
METHODOLOGY

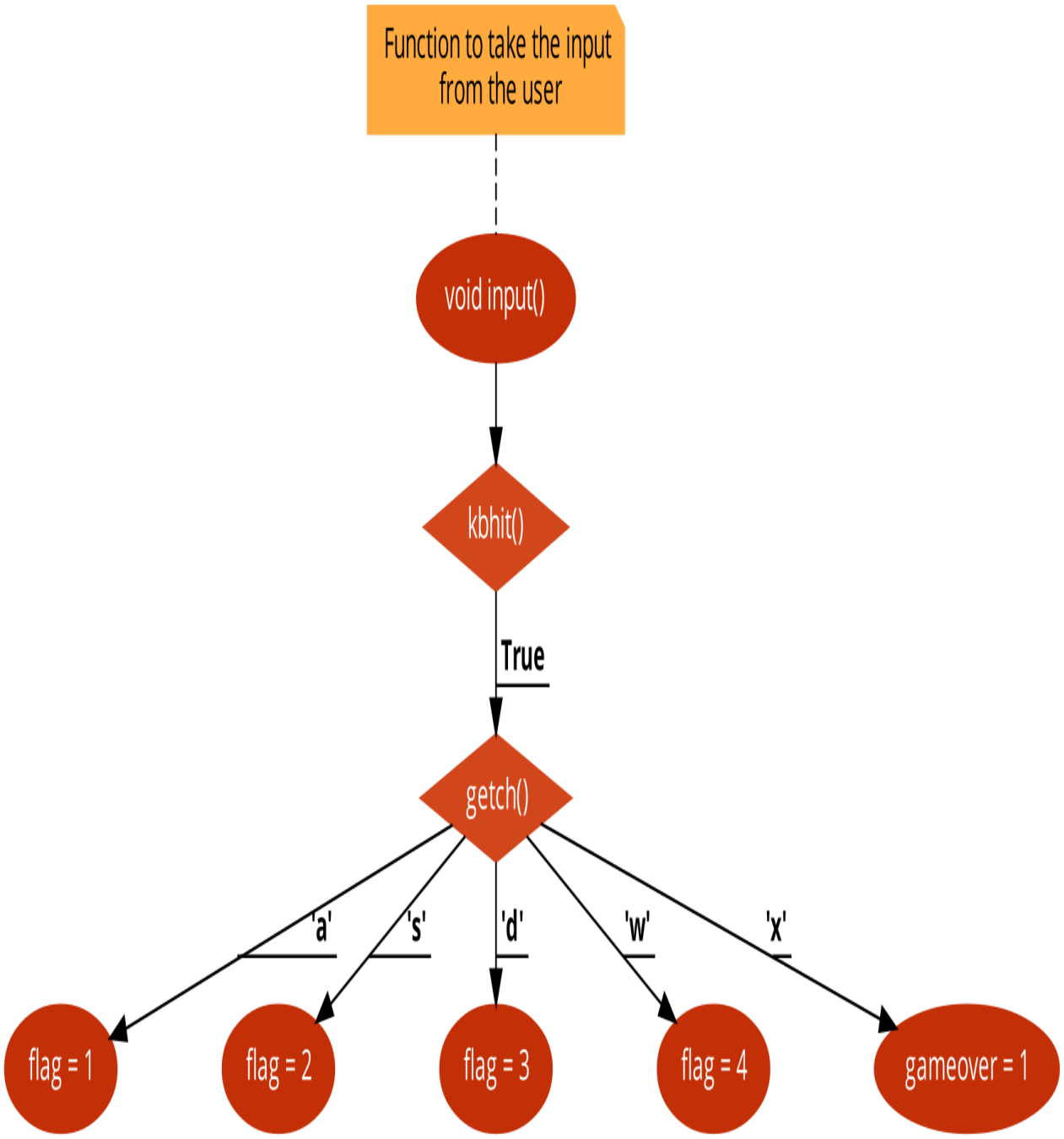
Flowchart:

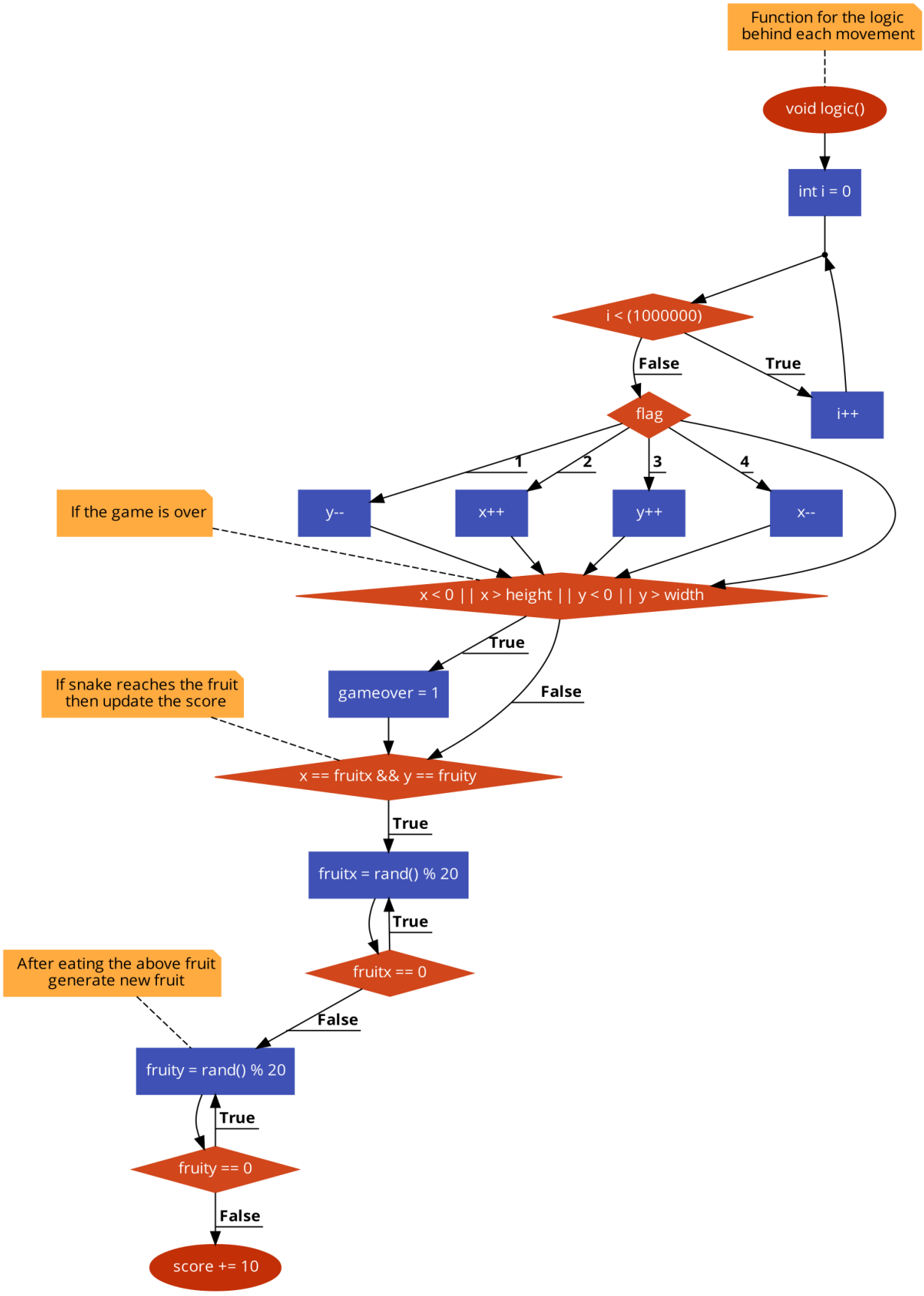



Function to print the game rules
void Print()
system("cls")
printf(" @@@@@-----*BOTTLE GAMES PRESENTS**-----@@@@@\n")***
sleep(5)
printf(intf(" ######$$$$$-----SNAKE GAME-----$$$$$######\n")
sleep(5)
printf("\tWelcome to the mini Snake game.(press any key to continue)\n")
getch()
system("cls")
printf("\tGame instructions:\n")
printf("\n-
Use W A S D to move the snake (0) Up , Left , Down and Right respectively\n")
printf("-
You will be provided with food ( ) at several coordinates of the screen which you have to eat\n")
printf("-
Everytime you eat the food the score increases\n")
printf("-
When snake touches the walls you loose and is the end of the game\n")
printf("-
If you want to exit the game press x\n")
printf("\n\nPress any key to play game...")
getch() == 27
True
exit(0)


void gotoxy(int x, int y)
COORD coord
coord.X = x
coord.Y = y
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord)
Function to print loading screen
void load()
int row, col, r, c, q
36, 14
printf("loading...")
30, 15
r = 1
r <= 20
False
True
getch()
q = 0
q++
q <= 100000000
to display the character slowly
False
printf("%c", 177)
r++
Function to generate the fruit and within the boundary
void setup()
gameover = 0
Stores height and width
x = height / 2
y = width / 2
fruitx = rand() % 20
True
fruitx == 0
False
fruity = rand() % 20
True
fruity == 0
False
score = 0


Function to draw the boundaries
void draw()
system("cls")
i = 0
i < height
False
True
Print the score after the game ends
printf("score = %d", score)
printf("\n")
printf("press X to quit the game")
j = 0
j < width
True
False
i == 0 || i == width - 1 || j == 0 || j == height - 1
printf("\n")
i++
True
False
i == x && j == y
printf("#")
False
True
i == fruitx && j == fruity
printf("0")
True
False
printf("")
printf(" ")
j++


Function to take the input from the user
void input()
kbhit()
True
getch()
'a'
's'
'd'
'w'
'x'
flag = 1
flag = 2
flag = 3
flag = 4
gameover = 1


Function for the logic behind each movement
void logic()
int i = 0
i < (1000000)
False
True
i++
flag
1
2
3
4
y--
x++
y++
x--
If the game is over
x < 0 || x > height || y < 0 || y > width
True
False
gameover = 1
If snake reaches the fruit then update the score
x == fruitx && y == fruity
True
fruitx = rand() % 20
True
fruitx == 0
After eating the above fruit generate new fruit
False
fruity = rand() % 20
True
fruity == 0
False
score += 10

Algorithm:

STEP 1: START

STEP 2: Include all the necessary header files.

STEP 3: Declare variables i, j, gameover, score, x, y, fruitx, fruity, flag, height=30, width=30.

STEP 4: Create a function named void Print() to display the first part of the game which includes name of the game and instructions on how to play the snake game. Make use of sleep() function to create a delay between the transition

STEP 5: Create a function void gotoxy() which assigns the cursor location according to the x and y coordinate mentioned

STEP 6: Create a function void load() to make loading screen in the mentioned coordinate on the screen

STEP 7: Create a function void setup() to generate the fruit within the assigned boundary using rand() function.

STEP 8:Create a function void draw() to develop the boundaries for the assigned height and width values using for loop and print the boundaries .also print score column to print the score

STEP 9: Create a function void input() to take keyboard input from the user using kbhit() function. Using switch case navigation of the snake is done according to the user input through keyboard.

STEP 10: Assign the cases accordingly. W is assigned to move the snake UP. S is assigned to move the snake DOWN. A is assigned to move the snake LEFT. D is assigned to move the snake RIGHT and X is assigned to end the game

STEP 11: Create a function void logic() to perform all the operations accordingly. Create a delay time for the snake movement and using flag variable increment or decrement the x and y positions using switch case.

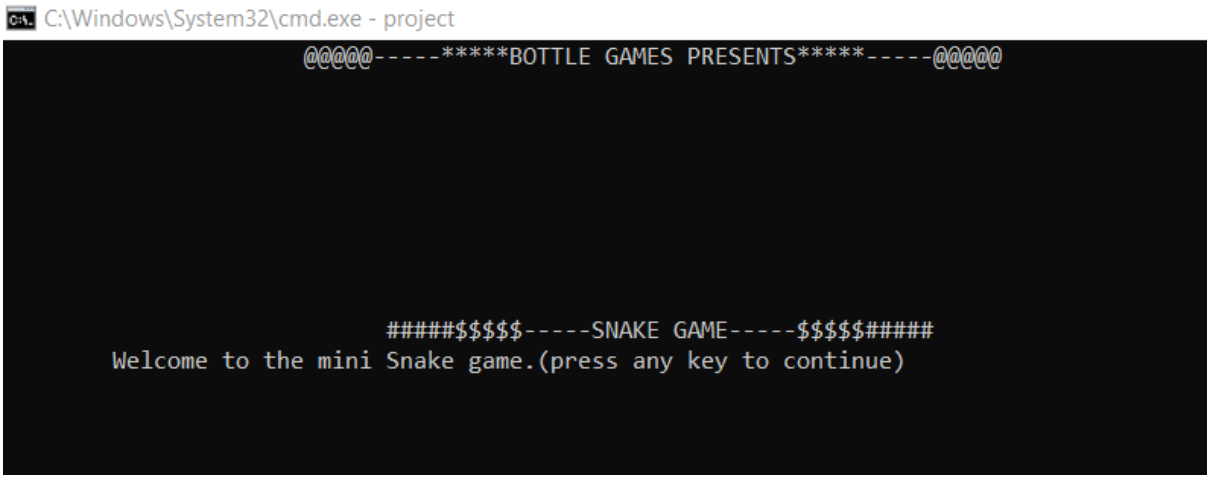
STEP 12: Using if condition check for the game over condition else update the score if the snake consumes the fruit accordingly and regenerate the fruit if it is consumed

STEP 13: Create a function void main() and call all the functions in order of their execution and put draw() , input() and logic() function in a while loop giving while(!gameover) to run the game continuously until the game ends

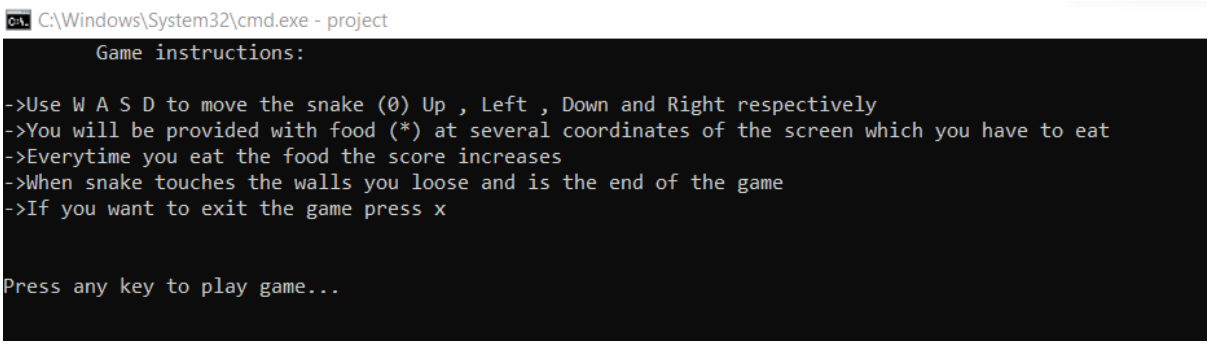
STEP 14: END

IMPLEMENTATION & OUTCOME

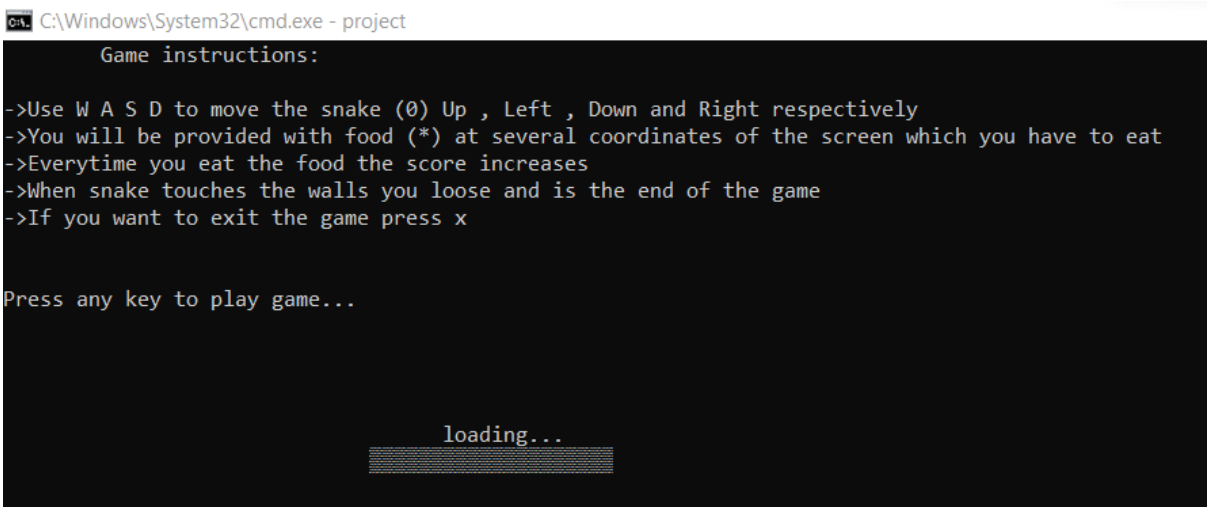
1) Display the welcome page of the game



2) Display the rules of the game

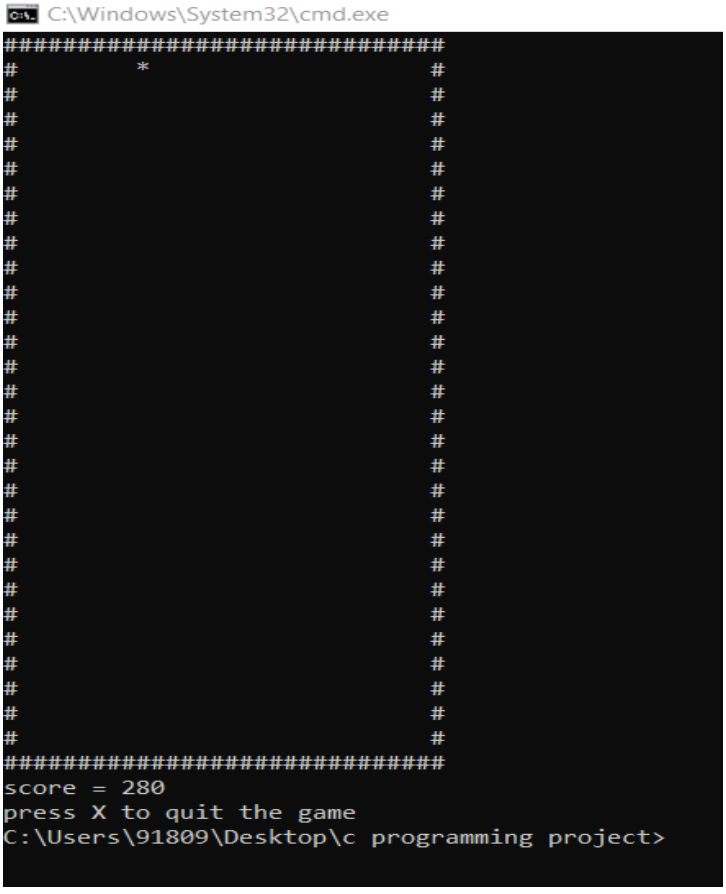


3) Once user presses, any key loading screen is displayed



4) Display the game screen for the user

5) Display the score once the game is over



References

[1] https://www.studytonight.com/c-projects/snake-game-project-using-c-language

[2] https://www.niit.com/india/knowledge-centre/Game-development-in-C

[3] https://www.cnet.com/culture/how-to-win-at-snake/

[4] https://www.kosbie.net/cmu/fall-10/15-110/handouts/snake/snake.html#:~:text=In%20the%20game%20of%20Snake,as%20possible%20before%20that%20happens.

[5] https://theprint.in/features/nokias-snake-the-mobile-game-that-became-an-entire-generations-obsession/462873/#:~:text=The%20concept%20of%20Snake%20originated,a%20solid%20line%20behind%20them.

[6] https://en.wikipedia.org/wiki/Snake_(video_game_genre)