**Chapter 1: Introduction of Project**

**1.1 Gaming Console**

“Gaming Console “this project is all about creating game and adding them in a one single running platform. With some natural graphics.it has two welcome screens first for listing project members and the other is main menu for the game. What have we done is creating 3 games and compressed into one console this console consist of games like Snake, Tic-Tac-Toe, Quiz.

Each and every game is independent of each other so their functions and the score are evaluated with user progress in the game. To end the game you need to press the ‘escape’ key to get to the main menu and to completely close the game you need to follow the instruction.

**1.1.1 Snake Game**

This **Snake Game** is a simple console application without graphics. In this project, you can play the popular “Snake Game” just like you played it elsewhere. You have to use the W-up, S-down, D-right or A-left arrows to move the snake.

Foods are provided at the several co-ordinates of the screen for the snake to eat. Every time the snake eats the food, its length will by increased by one element along with the score.

**1.1.2 Tic-Tac-Toe**

You have probably played the Tic-Tac-Toe game to pass time during school hours. It’s fun when you play with paper and pencil. Here, we have developed a **Tic Tak Toe game** – a simple console application without graphics.

It is the same nougats and crosses or the Xs and Os, the other names for Tic-Tac-Toe, you’ve played with paper and pencil. This mini game project is written in C language in a very simple manner; it is complete and totally error-free.

To win in this game a player’s character must be in 3 consecutive sequence in the game grid.

**1.1.3 Quiz**

This is a complete and error-free **Quiz Game in C** designed as a simple console application. In this game, a number of questions are asked, and the

User is awarded a score point in every correct answer and at last complete score is displayed.

In quiz game, questions are chosen in such a way that they cover all fields of a typical quiz contest. The user’s general knowledge is tested with quiz questions regarding science, technology, movies, sports, general health, geography and many more.

**1.2 Features of the Project**

* This project in c language of Gaming Console is a simple application with very basic graphics. In this project there’s two screen primarily the project main page that consists of group members name and the second is the main menu of the game.
* In the main menu, you can play popular games like snake, tic-tac-toe, and quiz.
* After selecting the game to play in every game there’s rules displayed on the screen firstly.
* In case of snake game you control the snake movement by pressing W-up, A-left, S-down, D-right
* Foods are provided at several coordinates of the screen for the snake to eat, every time the snake eats the food its length will be increased by one element(on its tail) along score
* In case of tic-tac-toe it can be played in multiplayer mode, in this gameplay there’s a grid of 3\*3 in which numbers are printed to grid numbers for the purpose of selecting a specific box
* A player can input number from 1 to 9 and put their character in that position.
* The first player to get 3 of her character in sequential order (up, down, across, diagonal) is winner.
* When all 9 squares are full and no player has there 3 character in continuous order, the game ends in tie.
* In case of quiz, several questions are displayed on screen player can answer them and find if they are correct or not. They are given 5 chances, if they answer 5 wrong question gameplay will end along with score.
* There is an emergency exit at every game which leads to the main menu of the game.
* Every game in this console is independent of each other along their functions.

**1.3 Requirements of the Project**

* Software Requirements

Operating System: - Windows 10  
Application Software: - Dev C++

Language: - C

* Hardware Requirements

Hard Disk: - 520GB

Ram: - 2GB

**Chapter 2: Present Work**

This project has been developed using c programming language. it is gaming console just like a play station where one can play multiple game as per his/her choice .it has a user friendly menu driven console. The complete code of the program is easy to understand for a genius student who has the knowledge of the c language as it contain simple logic with great application and not using much system and user define header files. we have made the best possible use of the function calling , looping etc. the header file used in this complete coding is listed below with short description :-

1. stdio.h:-it is the header file for the standard input output .

This is useful for getting input from the user (keyboard) and output result text to the monitor (screen). It is also needed for various other operation.

2. stdbool.h:- this header file allow us to use bool as a Boolean data types .

3.windows.h:- it is the base header file .it is contain declaration of all basic windows macros and different typedef screen “clear” function is present in this file only.

As soon as we run our code, we get a welcome screen with list of project members. Then it starts get loading for the next screen on the first screen itself we can have a look for great design build without any use of graphics. It also has a loading part which is create using looping concept.

It gives us a look of completely developed game which we used to play.

We have excellently used the getch() function to hold the program.

On the second screen which is the game menu screen we get the amazing graphics display developed through simple coding. Here we get there choice of games to play:-

Snake game, tic tac toe game and the quiz game.

The console is fully menu driven

On pressing 1 we are directed to the snake game.

On pressing 2 we are directed to the tic tac toe game.

On pressing 3 we are directed to the quiz game.

And on pressing 4 we can quit.

As we choose to play any game we are firstly welcomed and then we see game instructions and then games starts get loading. The journey till here is just like the other games we used to play.

After loading is done game starts and we can enjoys playing it now.

Fig: - 1

*Screen 1 of the application*

**2.1 Working Of the Driver Function (Main Menu)** :-

This part consist of welcome screen and menu screen too. For making, it work efficiently and user friendly we have used certain functions descriptions below:-

1. main() :- in the function, we have called screen1() and screen() function respectively and we have used getch() to get users choice for menu option for this, we have used a switch case for making it menu driven. To terminate the console, we haves used quit (0) function.

Switch case have been set in such a way that if we input any involved choice we returns to the same game menu screen.

2. screen 1(),screen 2() function :-this is a simple user define function for displaying welcome and menu screen , draw the beautiful graphics for better look .for this we have used some other user define function listed below:-

a) box():-the function takes argument of all integer types that are coordinates for drawing a rectangular or square boxes as per the use . We have just used looping concept to print a character “I” where ASCII value is 177 to make boxes.

b) gotoxy():-this function takes argument of all integer types to set the cursor anywhere in the console as per the use . In the turbo c++. We directly use gotoxy () library function. It uses a handle to write a console. GetStdHandle(STD\_ OUTPUT\_HANDLE, coord) is used to get handle which we use to write on console.

Set console cursor position then users that handle and coordinate cursor to specified position the screen/ console.

c) load2():- this function is used to print the loading part on the screen wherever used. It uses excellent concept of looping to delay certain micro seconds and display character one by one.

d) Caller ():- this function is almost same as main(). The only difference is that it skips screen1 that is welcome screen. In this function we don’t call the screen 1(). Because we don’t need to display the welcome screen again and again.

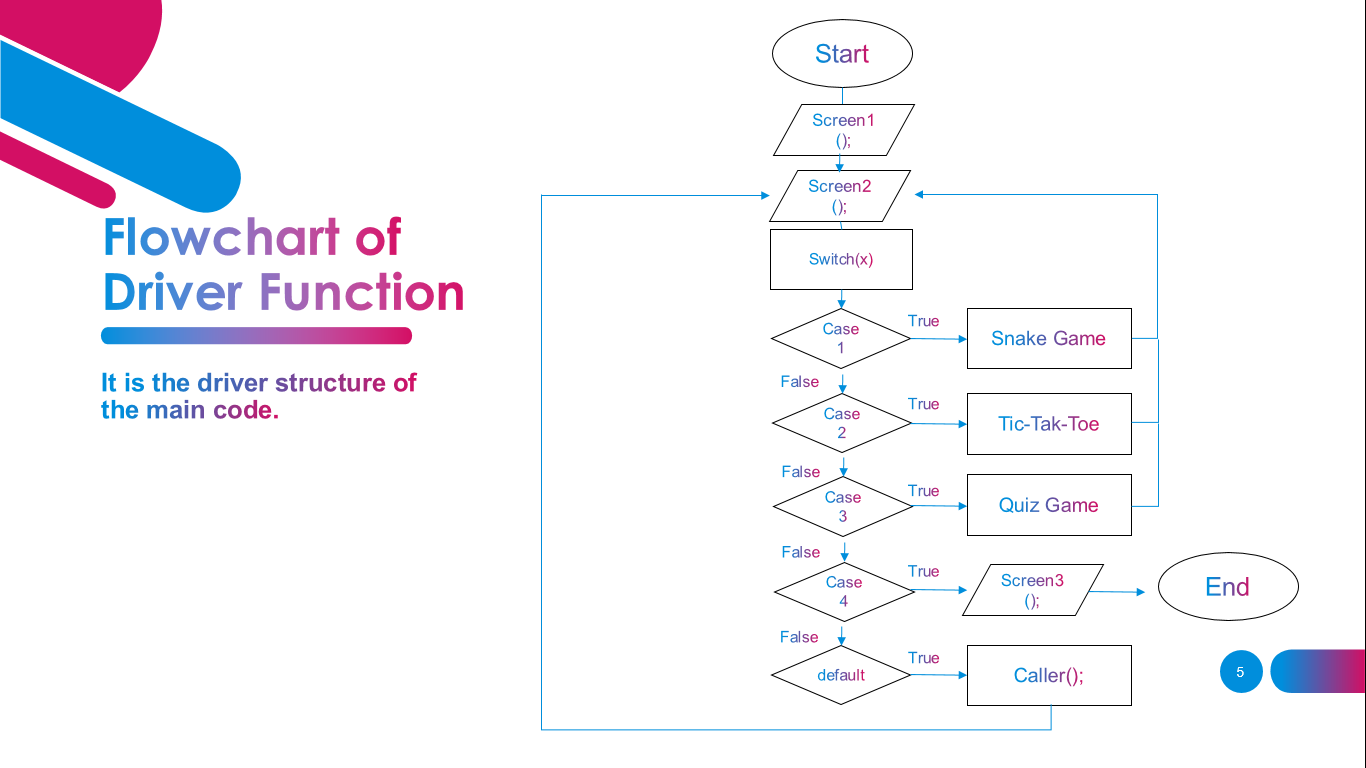


Fig -2

*Flowchart of Driver Function*

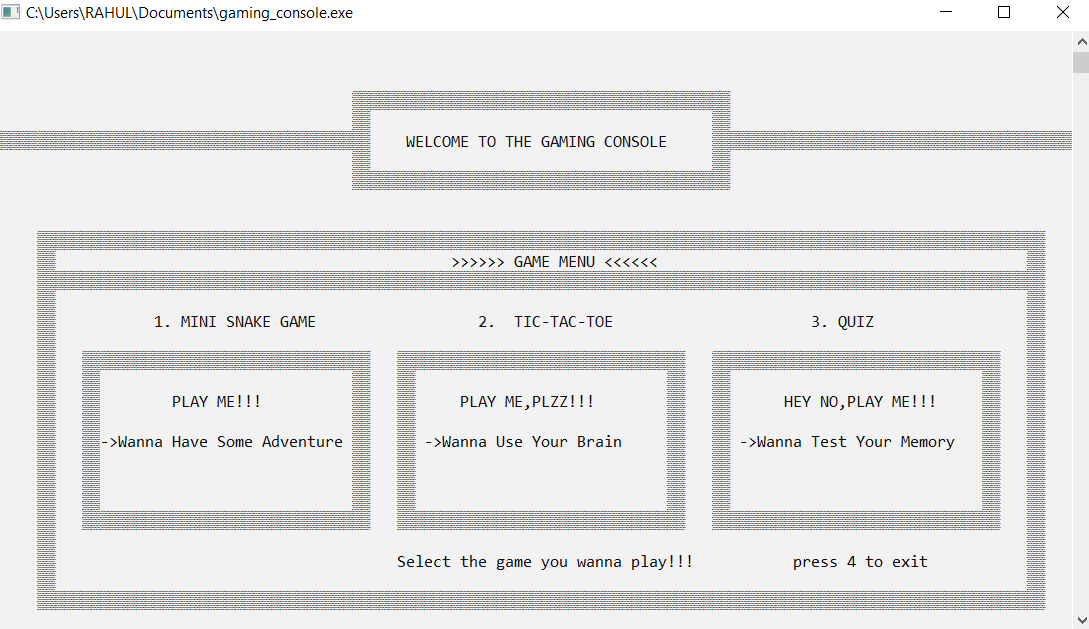


Fig:-3

*Screen2 of the application (Main Menu)*

**2.2 Working of Quiz Game:-**

This is the simple quiz game which we used to play anywhere. It has 12 questions in a row connected using if-else-if statement .the correct answer credit you 1 score each and after answering all the question you get grade according to the score obtained. on choosing 3rd option in the game we are directed to welcome screen of the quiz game and then instructions and then press any key to the continue after it, that game gets loading and then we have the quiz question to answer . We can leave and return to main menu anywhere in between the question round by pressing escape key. We have to use certain function in this game listed below:-

1 quiz():-the main function where we call certain other function required for the processing the game.

a) rules ():- it display the welcome screen and instructions for the game .

b) load ():-this is same function as load 2().the only difference is that we use different character to print the loading part.

c) round ():- in this game firstly we have to initialize an array of 12 elements for 12 question containing random numbers from 0 to 11 . And then use a while loop till the value of the count reaches equal to 11.using if statement and the help of randomly arranged elements we print question for the users randomly here. This is done to omit the rand () function which everyone uses. At last we print score and grade accordingly. And also we have called the caller function in this function where user inputs the choice as escape to exit the game.

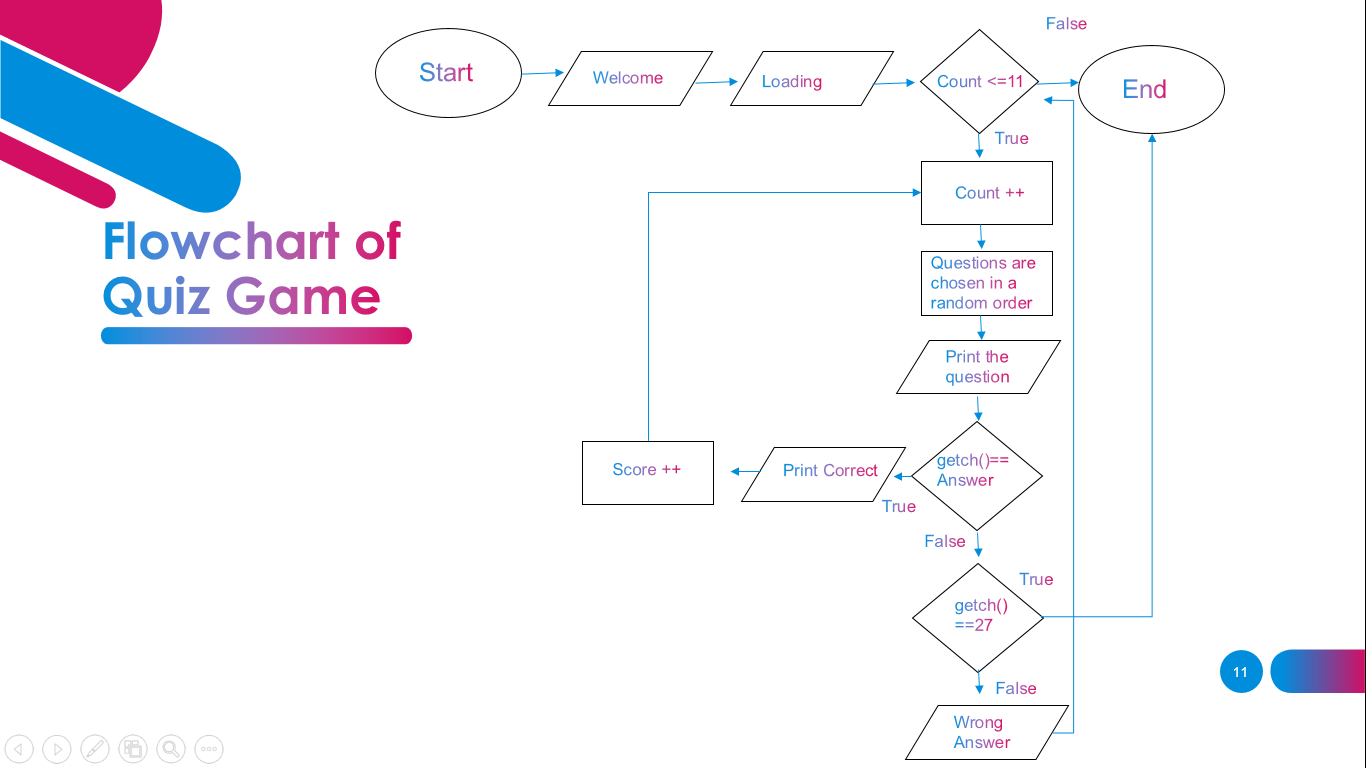


Fig: - 4

*Flowchart of Quiz Game*

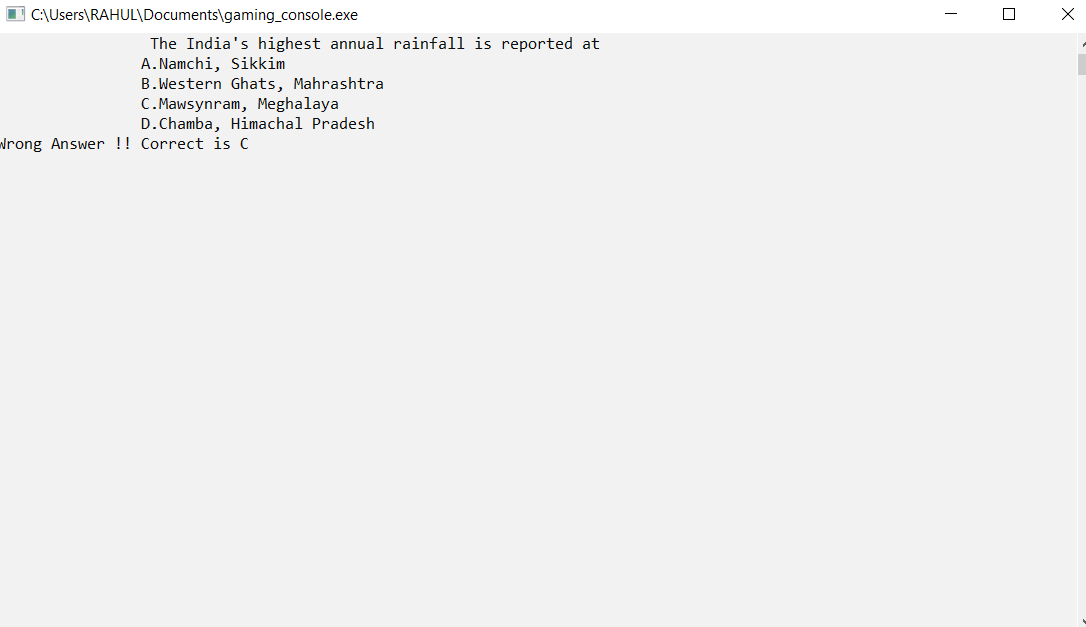


Fig: - 5

*Quiz Game*

**2.3 Working of the Snake Game**

This is the popular snake game which we used to play early in our childhood and still popular. Its working is not as simple as it looks while playing game. But through our code we are making it easier to understand its working concept. First of all, have used enumeration (enum), a user defined data type to make coding more intelligible. enum consist of a set of named constants called enumeration , when we need a predefine list values which do represent same kind of numeric or total data we use an enum . We use an enum when variable can take one out of small set of possible values.

There the enumeration has type name direction consist of 5 sets of possible values:- stop, left , right , up and down.

We have declared certain variables for different coordinates for snake head (x,y),fruit (fruit x, fruit y), tail length ,score and two arrays for tail of the snake .

It uses certain function listed below:-

1. snakegame ():- it is the main function to control this game here we call certain functions for the processing of the snake game. The function are listed below:-

a) print ():- this is the simple function just to print welcome and instruction for the game.

b) load ():-for the loading part mentioned above.

c)setup():- this is a simple function where we first initialize certain values such as ,score, direction, x-y coordinates of snake head and fruit .here we used rand() function to display fruit randomly on the console within the border we have used rand() function display such way that fruit get its coordinate within the border wall.

In the snake game () after all these function are executed we have used a while loop which terminates when it gets message that the game is over.

First of all an entering into while loop, its call the following function in the sequence below:-

Drawn():- this function used simple logic to draw border using looping concepts and display feed randomly each time the snake eats it up and the length of the snake increases by 1 .

We have used nested loop in this. The outer loop is for row (height) and lines for column (width). For drawing vertical border we first check for whether it is first column or not and then for last column. And then use move next line and the same process.

Here the cursor moves line by line within the border .if the coordinates of the cursor matches the coordinates of the snake head then we print character for the snake head that is “@”.and if the function coordinates of cursor matches the coordinates of the food then we print food character ‘F’.

In the else for we used a for loop to print the snake body and used a Boolean flag to check whether we print the body or not because we have to print a blank space if it does not eats any food. We have also printed the score each time the snake eats up the food.

Input ():- this function used to get user to get user input during the gameplay and set the direction for snake to move .first of all it will check that if any key is pressed or not .we have used a switch case that checks what user the inputted for the direction .It was 5 choices ‘a’, ‘d’, ‘w’, ‘s’, ‘x’ which controls the direction of snake and ‘x’ is to terminate the game.

Logic ():-as form the name itself we get know that this function includes the logic of the whole game. It is the important function of this snake game. This function is used for making the snake head and body move on the console during the gameplay. This function containing the terminating condition to end the game. The max length of the snake can be upto100 because we get only 100 spaces for an array in which we store its coordinate .firstly in this function we store the coordinate of the snake body in an array and swap the values accordingly .values will be swapped only if snake eats

The fruit and its length increases.

Then we have used switch statement to make snake body move along with its head accordingly to the direction set by player/user. Each time the function is called the snake moves first position in it’s respectively .then check for the game over condition. The first if statement checks whether snake has hit the wall or not .if it hits the wall the Boolean value of game over changer to true secondly , we have checked a for loop along with if statement weather snake head eaten its body or not.

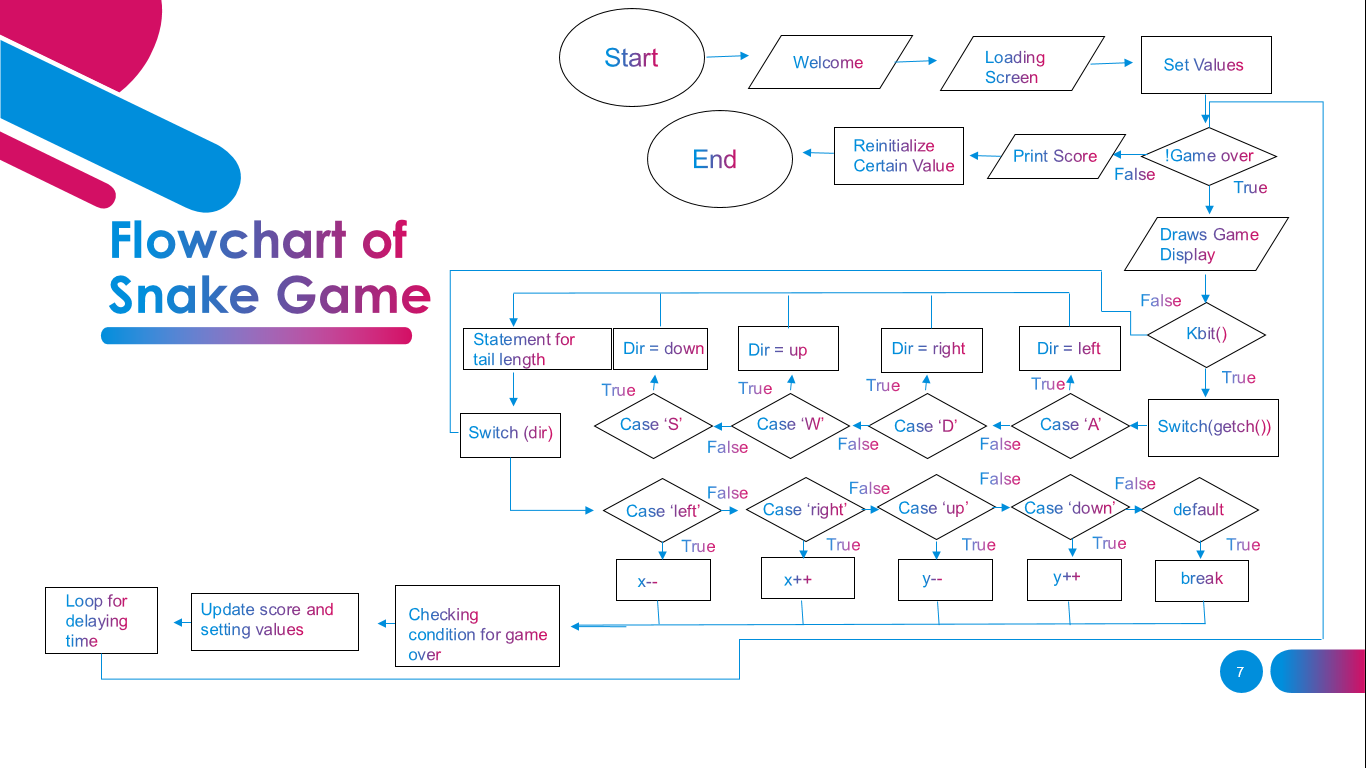
Thirdly it checks weather snake had eaten fruits or not if it has eaten up the fruits or not .if it has eaten up the fruit then we increase the score along with tail length (snake body) and set the values for fruit to display again on a random position.

After this function ends we have to use for loop with the while loop just to delay certain time for the further execution of the code. This is done to slow down the movement of the snake .we know that process of function calling it very fast, thus we need to slow down this fast process.

This while loop keeps on calling the function again and again until the terminating condition is reached.

After that when game ends final score is display and then press any key to return to main menu.

We also have the choice to exit the game in between the gameplay.

Fig:-6

*Flowchart of Snake Game*



Fig: - 7

*Snake Game*

**2.4 Working of the Tic-Tak-Toe Game**:-

The code of this has been developed using simple array matrix that we have studied earlier.

The elements of matrix is initialized with numbering from 1 to 9.this is done to make the game user friendly so that player/user don’t get any trouble in playing the game. The tic-tac-toe function is the main function for this game code in which we have called certain after function below its description:-

1) tttrule ():- This function is used to print welcome and game instructions screen.

2) load ():- This function is used for printing loading screen.

Then we have a while loop that terminates when the condition for win or draw is satisfied. A player wins if their respective character is plotted 3 terms in continuous along the two diagonal or along the rows or columns when all 9 grids are full and no player has their respective 3 characters in continuous order, the game gets draw.

Inside the while loop here we call following function listed below:-

1. Drawtt():-this is a simple function to draw the main display of the gameplay.
2. Turn ():-this is the main function of the game code. Firstly. This function sets turn for the two players.

It only decides or process that which player will input their character first and then next. It is done through if statement where it checks whose turn is now. Then this function takes the choice of the player. If player press “escape” key then the game ends and we return to main menu.

After this we have the switch statement to get the player choice numbering from 1 to 9. The player choose row and column to print their respective character on the game board. If choice is not between 1 to 9 then it display “Invalid choice” message.

We have used if else if statement that further checks whether the choose player is already booked or not end set the next player turn accordingly. Then we have one variable “complete” that is used to indicate in the code that one of the player has won and another named “time2” to indicate the game has been drawn whichever condition matches it prints the message respectively. If the wining or draw condition is not matched then while loop again calls the draw function and repeats the same procedure again.

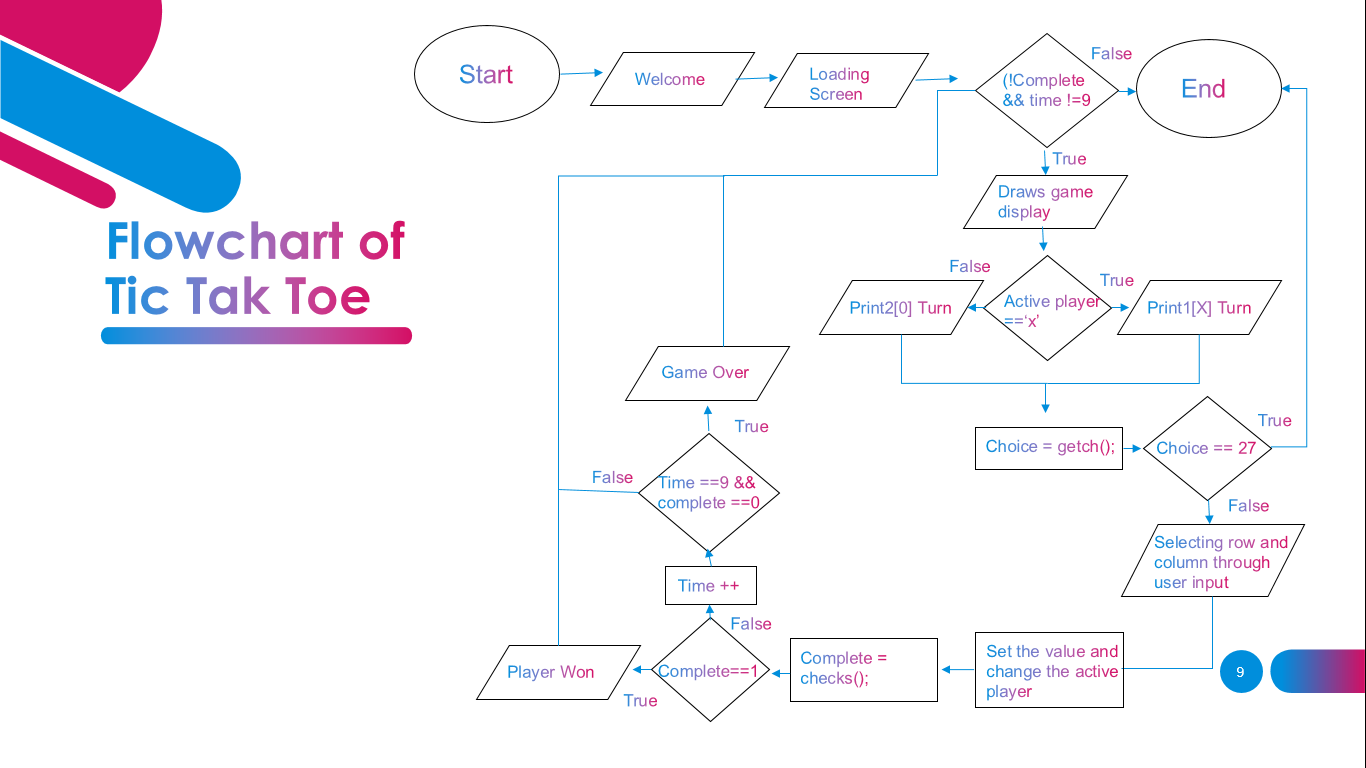


Fig: - 8

*Flowchart of Tic-Tak-Toe*

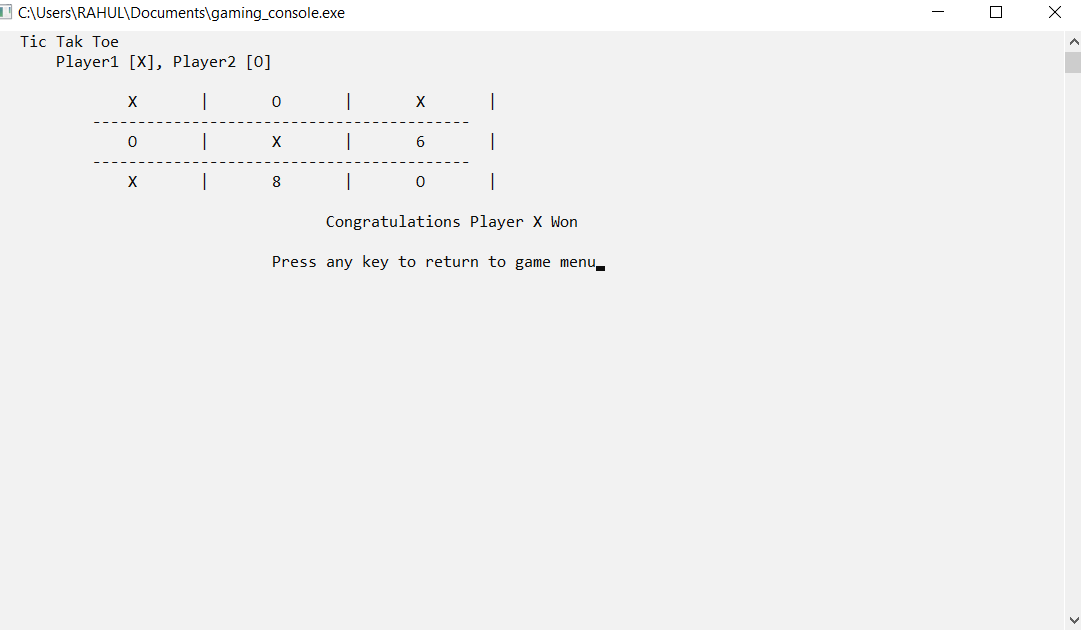


Fig: - 9

*Tic-Tak-Toe*

**Chapter 3: Result and Discussion**

While making of the project on “Gaming Console” we made our progress by solving a number of problems we went through. Solution to each problem by our self was most important part of the project and this provided us a lot of experiences, which will surely help us in the future.

Through this project we come to know about certain header files and few inbuilt functions such as kbit(); function. And also we developed our concept of using handler in c language which proved to be very helpful throughout the code and for future scope too.

We could easily set the cursor position through the handler concept which helped us in drawing the framework and graphics for the whole gameplay we also learnt some excellent looping and function calling concepts.

We got the concept of creating user defined data types through enumerations to make code more intelligible. Some important things that we learned include designing a good program architecture and converting real life situation into and efficient code and how to write a user friendly, good looking, easily readable and understandable as well as time and memory efficient code.

At last we would like to conclude that this project was very useful for us and it added immense knowledge in us

**Chapter 4: Future Scope**

In this project because it is our first core project after learning c. we can say we made our best to this module run perfectly in our system with low pc requirements.

1. We can add some better graphics in it.

2. We can also add A.I (artificial intelligence) functionality in the game.

3. We can make it more productive (commercial wise) by adding user’s choice game in it.

4. We can make launch it web platform.

5. We can make it file oriented.

6. Object oriented features can also be added.

7. We can also process user data to see past score.

**Chapter 5: References**

**1. Yashavant P.Kanetkar (2007)** *Let Us C.* BPB PUBLICATION, New Delhi 2007.

**2. Brian W. Kernighan and Dennis M. Ritchie,** *The C Programming Language,* Prentice Hall of India., Pearson Education India