### "Slithering Adventures: A Graphical User Interface (GUI) Snake Game"

### A PROJECT REPORT

***Submitted by***

**Aravinth Raj [192210197]**

**Praison Sam Asirvatham [192210260]**

*Under the guidance of*

## Dr. E.Anbalagan

***in partial fulfillment for the completion of course CSA0888- PYTHON PROGRAMMING FOR NATURAL LANGUAGE PROCESSING***



**SIMATS ENGINEERING**

**THANDALAM**

**MARCH 2024**

## BONAFIDE CERTIFICATE

Certified that this project report titled “**Classic Snake Adventure: Python Edition**” is the Bonafede work of “**Aravinth Raj D[192210197] , Praison Sam Asirvatham[192210260]**” who carried out the project work under my supervision as a batch. Certified further, that to the best of my knowledge the work reported herein does not form any other project report.

Date: Head of the Department

# ABSTRACT

# The Nokia-inspired Snake game implemented on the 8051-platform utilizing the AT89551 Microcontroller integrates an 8x8 LED dot matrix display and a five-way key interface. This classic game experience is enhanced with features such as a dedicated delay settings key, enabling players to easily adjust the snake's speed. The game's core functionality relies on the microcontroller's ability to interpret input from the switches, which are akin to buttons on a mobile device, for controlling the snake's movements – left, right, up, and down. Python is employed in the development process to translate the game logic into code, enabling the display of the snake and fruits on the 8x8 dot-matrix and facilitating interaction through the switches.

# Expanding upon the traditional Snake game, this project seeks to introduce new elements to elevate the gaming experience. Incorporating computer-controlled intelligent opponents adds a layer of challenge for players, as they must navigate the game while contending with dynamically moving adversaries. Additionally, the multiplayer feature enhances the game's social aspect, allowing multiple players to engage in competitive gameplay over a network. By embracing these innovations, the project aims to push the boundaries of the classic Snake game, transforming it into a more dynamic and engaging experience.

# Through this project, advanced concepts such as networking and artificial intelligence are explored within the context of a familiar and accessible game format. The simplicity of the Snake game serves as an ideal platform for experimentation and learning, providing opportunities to delve into complex topics while retaining an enjoyable and intuitive gameplay experience. By venturing into new dimensions of gameplay and functionality, this project aims to captivate players and showcase the potential for innovation within the realm of classic arcade games.

## Table of Contents

#### Title Page

[CERTIFICATE **1**](#_TOC_250010)

[ABSTRACT **2**](#_TOC_250008)

[TABLE OF CONTENTS](#_TOC_250007) **3**

[LIST OF TABLES **5**](#_TOC_250006)

[LIST OF FIGURES **6**](#_TOC_250005)

[CHAPTER 1: INTRODUCTION 8](#_TOC_250004)

1 Introduction to the Project

[CHAPTER 2: LITERATURE REVIEW 9](#_TOC_250003)

* 1. [Literature Survey](#_TOC_250002)
  2. [Modules](#_TOC_250001)

2.2.1 PyGame

2.2.2 Python time module

2.2.3 Python random module

[CHAPTER 3: PROBLEM FORMULATION 10](#_TOC_250000)

* 1. Description of Problem Domain
  2. Problem Statement
  3. Objective

**CHAPTER 4: SYSTEM DESIGN 13**

* 1. Software Design
  2. Functional Requirements
  3. Non – Functional Requirements
  4. **Software Requirement Specification 14**
     1. SRS
     2. Problem/Requirement Analysis
     3. Requirement Specification
     4. Role Of SRS
  5. **Hardware And Software Requirements 15**
     1. Software Requirements
     2. Hardware Requirements

**CHAPTER 5: IMPLEMENTATION 16**

* 1. Python
  2. PY-GAME
  3. Functional [Description](https://www.studocu.com/in?utm_campaign=shared-document&utm_source=studocu-document&utm_medium=social_sharing&utm_content=mini-project-report-111)
  4. Implementation Steps
  5. Working Algorithm

**CHAPTER 6: RESULT AND DEMONSTRATION 20**

**6.TEST CASES**

**CHAPTER 7: CONTRIBUTION OF PROJECT TO THE SOCIETY 22**

**CHAPTER 8: CONCLUSION, LIMITATION AND FUTURE SCOPE 23**

**REFERENCES 24**

## List Of Figures

**Title** **Page**

**Fig** **3.1.1**: Representation of Snake and Apple 10

**Fig** **3.1.2**: Snake Stuck Against the Wall 11

**Fig** **6.1.1**: In – Game Screenshots 21

**Fig 6**.1.2: Q – Quit or C – Play Again 21

## List of Tables

**Title** **Page**

Table 5.1: Python Version 17

Table 5. Function Description 18

# Chapter 1: INTRODUCTION

Playing games is fun and exciting. It gives us relief from stress and unwinds from our stressful work. Many of us spend our free time or others that use most of their time in playing and exploring new games. Today, with the raped development of technology we have, games that are rising up together with it. Nowadays with technology we have many games that are developed for computers specifically for windows. With the high technology equipped with these computer games become robust and attract many people to buy or have this gadget for them to experience what's inside it which makes it a trend for the new generation of gadget.

Snake game is a computer action game, whose goal is to control a stake to move and collect food in a map. It has been around since the earliest days of home computing and has re-emerged in recent years on mobile phones.

It isn't the world's greatest game, but it does give you an idea of what you can achieve with a simple python program, and perhaps the basis by which to extend the principles and create more interesting games on your own. To move the snake, use up arrow for up, down arrow for down, "left arrow for left and right arrow for right. Press "Q' to exit the game at any time, press "C" to continue the game.

The aim of the game is to collect the dots (food) and avoid the obstacles (walls, boundaries). As you collect the food, the stake gets longer. The score also increases. There is no concept of life. Once you hit an obstacle, that's it, game over.

# Chapter 2: LITERATURE REVIEW

#### LITERATURE SURVEY

The history of the Snake game goes back to the 1970's, the concept originated in the 1976 arcade game Blockade, and its simplicity has led to many implementations. However, it was the 1990's when the game took on the look that we will be using. It was sold under numerous names and many platforms but probably gained widespread recognition when it was shipped as standard on Nokia mobile phones in the late 1990' The first published Nokia, for monochrome phones. It was programmed in 1997 by Taneli Armanto of Nokia and introduced on the Nokia 6110,The game involves controlling a single block or snakehead by turning only left or right by ninety degrees until you manage to cut a block. When you get the block, the Snake grows an extra block or body segment.

#### MODULES

**2.2.1PyGame**

Py game is a cross-platform set of python modules designed for writing video games.

It includes computer graphics and sound libraries designed to be used with the Python Programming language.

* To install the library, you can use pip installer from the command line:

pip install pygame import pygame

**2.2.2Python time module**

Python has a module named time to handle time-related tasks. To use functions defined in the module, we need to import the module first. Here's how:

import time

**2.2.3Python random module**

Python has a built-in module that you can use to make random numbers

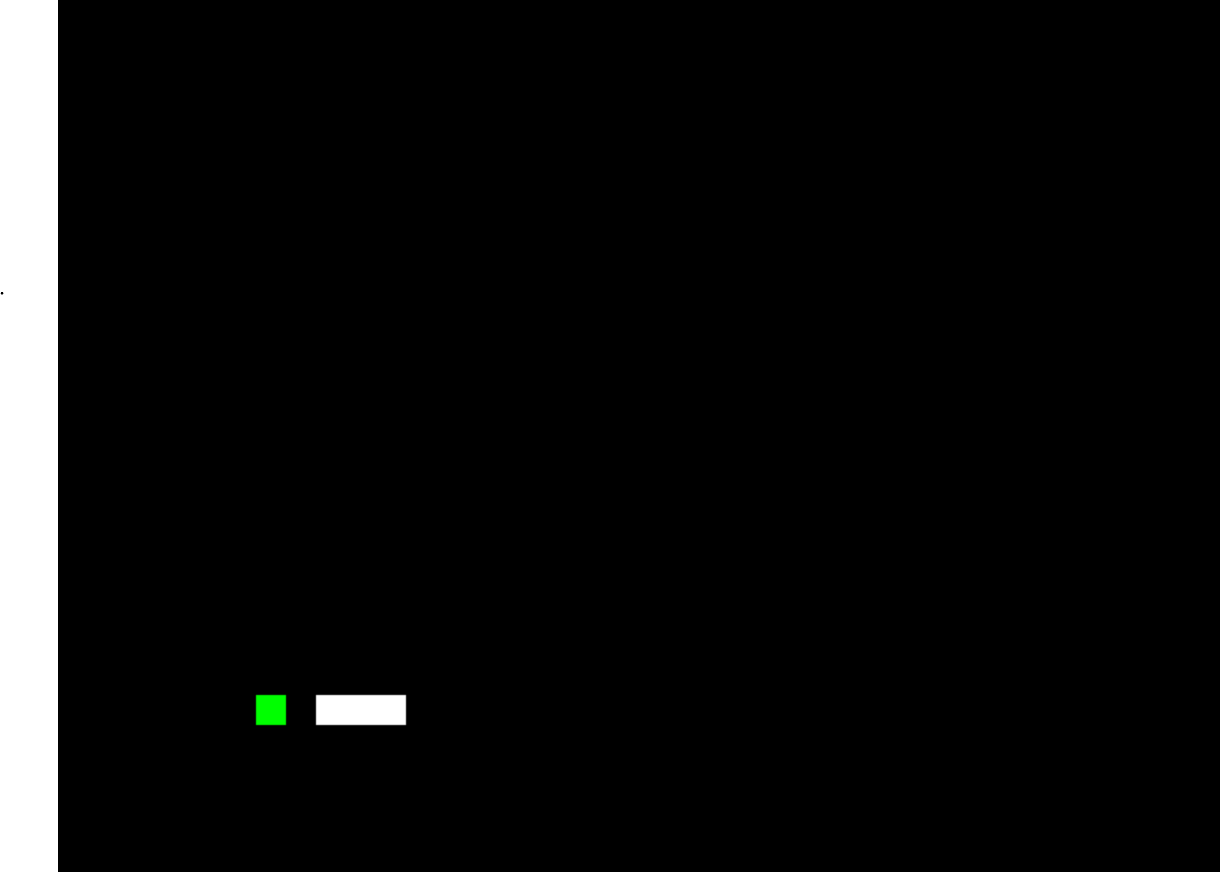
import random

# Chapter 3: PROBLEM FORMULA

#### DESCRIPTION OF PROBLEM DOMAIN

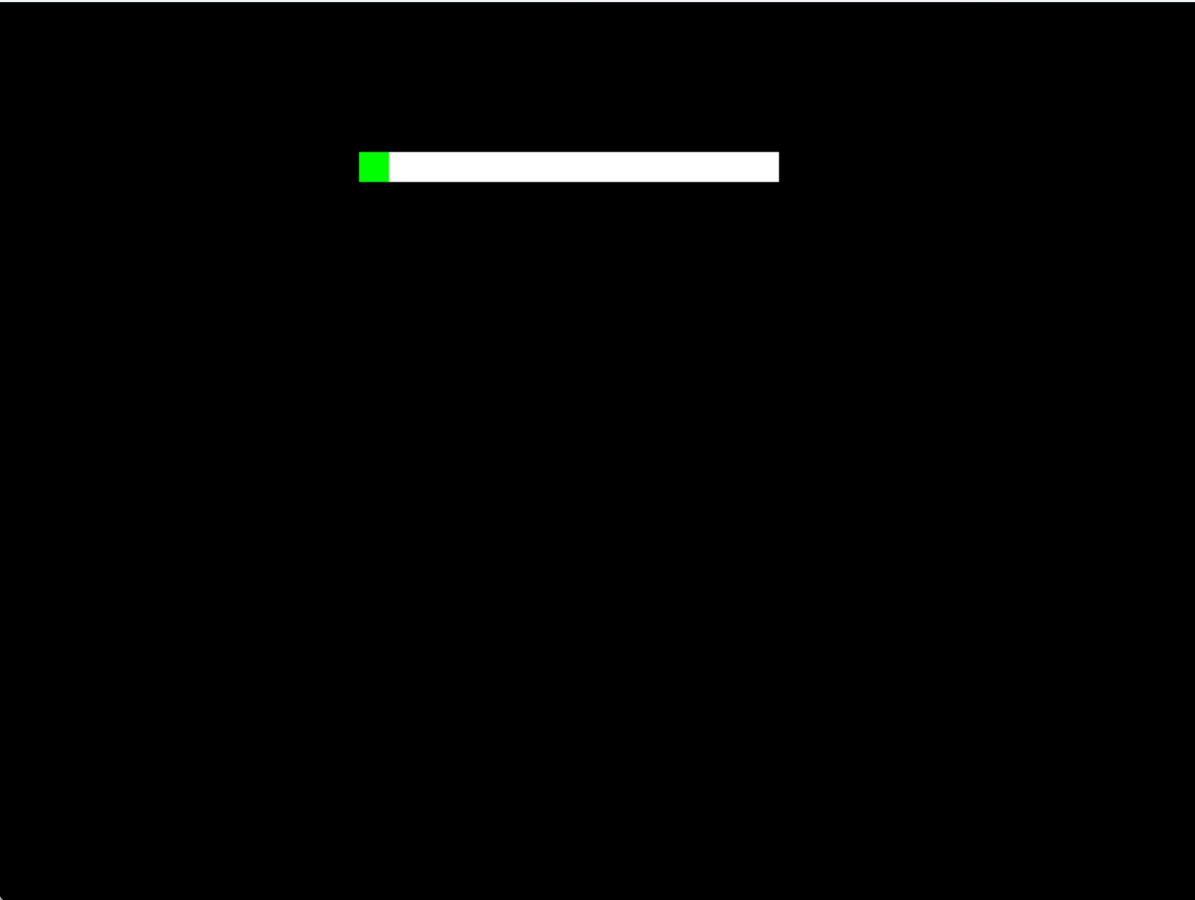
The core focus of our project was to determine which algorithms would be more effective in a hard real-time environment. The domain in this case is the Snake Game, which will, in turn, attempt to identify an, or even the, algorithm that can not only play the game but compete with human players. The Snake Game is a classic arcade style game where it is a single-player game but the focus is to achieve the highest score possible thus competing with yourself and others.

To play the game one controls a snake by selecting one of the cardinal directions that the snake will move in. In order to score points you must direct the snake to an apple, there is only one apple



**Fig 3.1.1: Green** Square Is Apple and White Rectangle Is Snake

square or unit. The game ends when the snake runs into either the boundaries of the play area or itself, the trailing snake body. The domain provides a very interesting problem given that the snake always moves after a given timing delay and the snake continually grows in size. The delay is the feature that really makes the game difficult because if you do not react fast enough the snake will continue moving in the last direction given. This causes the player to try to act as quickly as possible before the snake runs into an obstacle. Also because the snake is constantly trailed by its tail(being the main obstacle) any move taken cannot be undone or immediately back tracked. So if you were to make a wrong turn into a dead end there is no way to reverse that move to back out of the loop.



**Fig 3.1.2 :** In This Figure, Snake Was Moving Toward Apple But Got Stuck Against The Wall Instead

Although with every move your tail moves thus creating a situation where a dead end or closed loop will no longer be a dead end after x moves. The apple also can be very tricky since you cannot know where the next apple position will be and this makes it difficult to go straight for the current apple without thinking of how to get into a good position for the next apple.

#### PROBLEM STATEMENT

The problem is to design a Snake Game which provides the following functionalities :

* + 1. Snakes can move in a given direction and when they eat the food, the length of the snake increases.
    2. When the snake crosses itself, the game will be over.
    3. Food will be generated at a given interval.

The main classes that can be used are :

1. Snake
2. Cell
3. Board
4. Game

**3.3 OBJECTIVE**

Snake game is one of the most popular arcade games of all time.

In this game, the main objective of the player is to catch the maximum number of fruits without hitting the wall or itself.

Creating a snake game can be taken as a challenge while learning

Python or Pygame. It is one of the best beginner – friendly projects that every novice programmer should take as a challenge.

Learning to build a video game is kinda interesting and fun learning.

# Chapter 4: SYSTEM DESIGN

To create a Snake game that allows users to control the movement of a snake on a screen, to get points for eating food and avoiding running into the walls or the growing tail of the snake itself. In this problem, we want to write a game where a graphical representation of a snake moves across the screen. When it encounters a piece of food, the snake grows longer and we gain a point. If it hits the wall we die.

To write this program we are going to need:

1. A way of representing the sake of representing the food.
2. A way to display the score,
3. A way for our instructions to reach the snake
4. A way to know when we've run into something and died

Our system is going to involve working with both hardware and software, and so we will need to understand what we have available in hardware that can assist us.

If we build our software so that the snake is controlled by directional arrows on the keyboard. Now that understand how our hardware will work in the design of our system, let's move on to starting the design of our software system

#### SOFTWARE DESIGN

* We are going to use an object-oriented approach and provide some detail here. We have to think about the Classes that we want to build, with the associated variables and functions that will make sense for the development.

Let's start by looking at the snake itself, the hero of the game. The stake has a location on the screen, and contains multiple visual elements, as it can grow, and the snake's head is connected to the rest

of the snake and the stake's body follows it around the screen. If the snake "eats" food, it grows. The snake moves in a very precise way. Based on what the user types, the snake will move in a given direction. Every time the snake moves, the head will go in the new direction, and every piece of the snake will move up, by occupying the space that was formerly occupied by the piece in front of it.

#### FUNCTIONAL REQUIREMENTS

Here are the requirements (functional requirements for how the snake moves.

1. The snake must appear to move around the screen
2. The snake must turn in response to user input
3. The snake will increase in length if it eats food
4. The snake will die if it runs to the walls
5. The snake never stops moving

#### NON-FUNCTIONAL REQUIREMENTS

The primary features of IT projects involve implementing like an application, a piece of infrastructure, a patch

In this specific context functional requirements tells us about what project does when interacted, whereas non-functional requirements describe about the bounds of performance should be:

#### Robustness

#### Reliability

#### 3.Availability

#### 4.Reusability

#### Effectivenes

#### SOFTWARE REQUIREMENT SPECIFICATION

* + 1. SRS

Software Requirement Specification (SRS) is the starting point of the software developing activity. As the system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need forthe requirement phase arose. The software project is initiated by the client. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase) The SRS phase consists of the activities.

* + 1. Problem/Requirement Analysis.

The process is order and more nebulous of the two, deals with understanding the problem, the constraints.

* + 1. Requirement Specification

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this

The Requirement phase terminates with the production of the validate SS document. Producing the SRS document is the basic goal of this phase.

* + 1. Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and developers. Software Requirement Specification is the medium which makes sure which the client and user needs are accurately specified. It forms the basis of software development.A good SRS should satisfy all the parties involved in the system.

The SRS document is the basic goal of this phase. The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers.

The developer is responsible for asking for clarifications, where necessary, and will not make alterations without the permission of the client.

#### HARDWARE AND SOFTWARE REQUIREMENTS

* + 1. Software Requirements

The development and deployment of the application requires the following general and specific minimum requirements for software:

* Programming Language Translaties-Python 18

IDE-Jupyter Notebook Operating System used Windows 10.

* + 1. Hardware Requirements

The development and deployment of the application requires the following general and specific minimum requirements for hardware:

Processor-(32-bit or 64-bit) RAM (8 GB).

# CHAPTER 5: IMPLEMENTATION

#### PYTHON

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language.

Python is easy to leam yet powerful and versatile scripting language which makes it attractive for Application Development

Python's syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas

Python supports multiple programming patterns, including object

#### PY-GAME

Python is the most popular programming language or nothing wrong to say that it is the next generation programming language.

In every emerging field in computer science, Python makes its presence actively Python has vast libraries for various fields such as Machine Learning (Numpy, Pandas, Matplotih), Artificial intelligence

(Pytorch, TensorFlow), and Game development

Game programming is very rewarding nowadays and it can also be used in advertising and as a teaching tool too. Game development includes mathematics, logic, physics, AL, and much more and it can be amazingly fine. In python, game programming is done in pygame and it is one of the best modules for doing so

Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python

programming language. Pygame was officially written by Pete Shinners to replace PySDL. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

#### Installing pygame:

The first thing you will need to do in order to create games using Pygame is to install it on your systems. To do that, you can simply use the following command:

*pip install pygame*

#### Python Version Release Date

Python 1.0 January 1994

Python 1.5 December 31, 1997

Python 1.6 September 5, 2000

Python 2.0 October 16, 2000

Python 2.1 April 17, 2001

Python 2.2 December 21, 2001

Python 2.3 July 29, 2005

Python 2.4 November 30, 2014

Python 2.5 September 19, 2006

Python 2.6 October 1, 2008

Python 2.7 July 3, 2010

Python 3.0 December 3, 2008

Python 3.1 June 27, 2009

Python 3.2 February 20, 2011

Python 3.3 September 29, 2012

#### FUNCTION DESCRIPTION

Once that is done, just import Pygame and start our game development. Before moving on, take a look at the Pygame functions that have been used in this Snake Game along with their descriptions.

#### Function Description

init() Initializes all of the imported Pygame modules ( returns a tuple indicating success and failure of initializations)

display.set\_model() Takes a tuple or a list us ats parameter to create

a surface (tuple preferred)

update() Updates the screen

quin() Used to un-initialize everything

set caption() Will set the caption test on the top of the display

screen.

event.get() Returns list of all events

Surface fill() Will fill the surface with a solid color

time Clock() Helps truck time

font SysFont() Will create a Pygame font from the System font

resources.

#### IMPLEMENTATION STEPS

1. Installing pygame
2. Create the screen
3. Create the Snake
4. Moving the Snake
5. Game over when snake hits the boundaries
6. Adding the Food
7. Increasing the length of the snake & Displaying the score

#### WORKING ALGORITHM

Let's look at how a program to run the whole game might look.

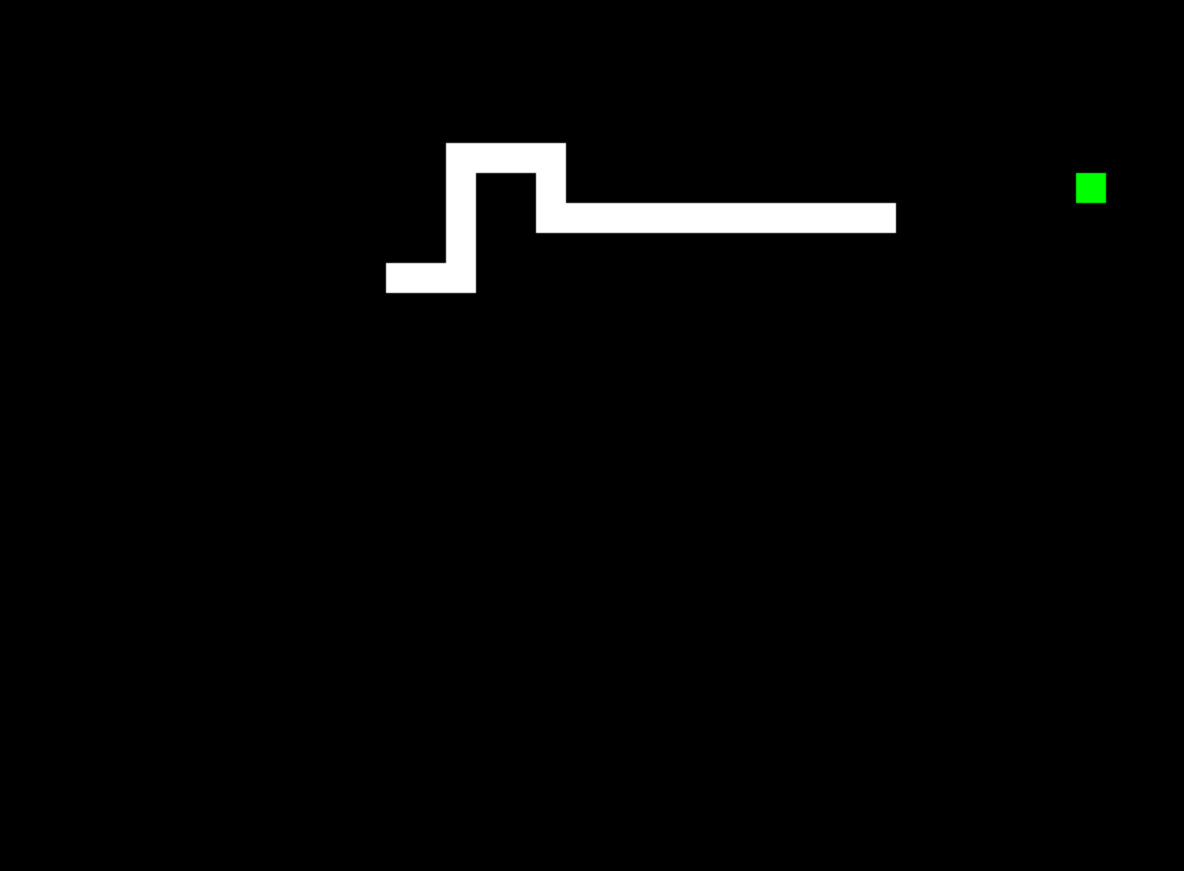
1. Draw the playing area with bounding rectangle, set the counter to zero and display I
2. Draw the snake in a starting position,
3. Draw the food in a starting location.
4. On user input, change snake direction.
5. Move the snake one move
6. If the snake is over food, eat it, increase the score, grow, move the food.
7. else if the snake is over in a wall, die.
8. Go back to step 4
9. Until the snake die

#### CHAPTER 6: RESULT AND DEMONSTRATION

#### 6.TEST CASES

* + 1. How can we test a full Snake game and, assuming it passes that stage,
    2. how can we playtest that? The functional requirements that we developed turn, almost immediately.
    3. To display the snake, the first thing we want to do is to make sure that we can draw the snake and move it around on the screen. So our testing for correct function will be

1. Can I display the snake's head on the screen?
2. Will it move around as I want it to using keyboard control?
3. Is it displaying correctly?
4. Is the body moving correctly?
5. If we identify an error in the snake, because it's a Class, we will go into the Snake class and fix it there. However, because we've written the Food and Scoreboard as separate classes, whatever we do in the Snake class shouldn't break anything in there, unless we accidentally change the code without noticing. The next step for the snake will be checking what happens when the head is detected as colliding with something. Does it grow when it eats cat food? Does it die when it hits a wall or itself! We'd then continue to test the program until we've tested all of the individual elements and their interactions together.
6. One useful test case is to see if everything is being drawn where you expect. Because we aren't.
7. Using all the screen, it's possible to draw the food or the snake so that it overlaps the black rectangle that's the boundary. Has the programmer put the correct limits on the ranges where the snake and the food can appear?
8. Testing the non-functional requirements often falls into the realm of playability.
   1. Many snake games increase the speed of the snake as it gets longer, increasing the difficulty even further. This increased movement speed gives a sense of urgency and can be a way to engage players with the snake. But make it too fast and it becomes unplayable.

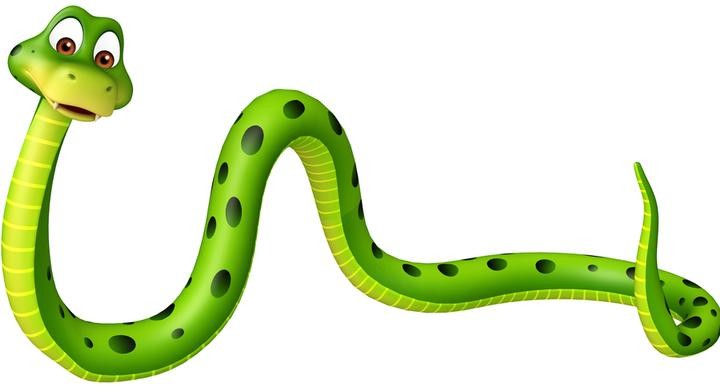


**Fig: 6.1.1 IN GAME SCREENSHOTS**

****

**Fig: 6.2.2 Q-Quit or C – Play Again**

# CHAPTER 7: CONTRIBUTION OF THE PROJECT TO THE SOCIETY



* + *Snake* is a classic game that requires players to assess their surroundings and find the quickest or safest route to a point. This is an excellent opportunity to learn about spatial awareness and plan ahead to your next move.
  + The classic game is infamous for using your own success against you when you become so long that you get in your own way. Whilst many games and activities can teach your child about vital life skills, there are not many that would educate on long term strategic planning.
  + As many parents will know, it can be extremely frustrating to reach such a high level and then lose as you crash into your own tail. The game requires patience in order to grow and a cool head once you inevitably lose. These are all valuable skills to learn early on in a child’s life that will benefit them in later years.
  + *Snake* is a tool that can be used as an educational helping hand. One of the important parts of learning is that you will never get something right the first time. *Snake* teaches children that practice makes perfect when it comes to learning new skills.

# CHAPTER 8 : CONCLUSIONS

The project in python programming of Snake Game is a simple console application with very simple graphics. In this project, you can play the popular "Snake Game" just like you played it elsewhere. You have to use the up, down, right, or left arrows to move the snake.

Foods are provided at the 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 along with the score. • It isn't the world's greatest game, but it does give you an idea of what you can achieve with relatively simple python programming, and perhaps the basis by which to extend the principles and create more interesting games on your own.

#### LIMITATIONS:

* The existing system only provides a text-based interface, which is not as user-friendly as Graphical user Interface.
* Since the system is implemented in Manual, the response is very slow.
* The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.

#### FUTURE SCOPE:

In this project, I have used a simple application. This project will be able to be implemented in future after making some changes and modifications as I made this project at a low level. The modifications that can be done in this project are:

1. It can be made with good graphics,
2. We can add more options like Top scores and Player Profile,
3. We can add multiplayer option

#### REFERENCES

#### 1. "Creating a Snake Game in Python" by Learn Python (PDF Link: [learnpython.org/snake\_game\_tutorial.pdf](learnpython.org/snake\_game\_tutorial.pdf))

#### 

#### 2. "Python Snake Game Tutorial" by Tech with Tim (PDF Link: [techwithtim.net/snake\_game\_tutorial.pdf](techwithtim.net/snake\_game\_tutorial.pdf))

#### 

#### 3. "Build a Snake Game in Python" by Programiz (PDF Link: [programiz.com/snake\_game\_python.pdf](programiz.com/snake\_game\_python.pdf))

#### 

#### 4. "Python Snake Game Tutorial" by Python for Beginners (PDF Link: [pythonforbeginners.com/snake\_game\_tutorial.pdf](pythonforbeginners.com/snake\_game\_tutorial.pdf))

#### 

#### 5. "Snake Game in Python Using Pygame" by GeeksforGeeks (PDF Link: [geeksforgeeks.org/snake\_game\_python\_pygame.pdf](geeksforgeeks.org/snake\_game\_python\_pygame.pdf))

#### 

#### 6. "Simple Snake Game Tutorial in Python" by CodeLoop (PDF Link: [codeloop.org/simple\_snake\_game\_python.pdf](codeloop.org/simple\_snake\_game\_python.pdf))

#### 

#### 7. "Snake Game using Python" by Guru99 (PDF Link: [guru99.com/snake\_game\_python.pdf](guru99.com/snake\_game\_python.pdf))

#### 

#### 8. "How to Code a Snake Game in Python" by The Renegade Coder (PDF Link: [therenegadecoder.com/snake\_game\_python.pdf](therenegadecoder.com/snake\_game\_python.pdf))

#### 

#### 9. "Creating Snake Game Using Python" by Educational Materials for Python (PDF Link: [edu-materials.com/snake\_game\_python.pdf](edu-materials.com/snake\_game\_python.pdf))

#### 

#### 10. "Snake Game Tutorial using Pygame" by Real Python (PDF Link: [realpython.com/snake\_game\_pygame.pdf](realpython.com/snake\_game\_pygame.pdf))

#### These tutorials cover various aspects of creating a snake game in Python using different libraries and approaches.