A Mini-Project Submitted to

MVP's
Institute of Management, Research & Technology [IMRT]
MVP Campus, Gangapur Road, Nashik

For the Degree of Master in Business Administration MBA [Information Technology]

In the Subject of
Software Development

Submitted by
Namrata Patil
Roll No. 15

Under the Guidance of
Dr. Varsha Bhabad
Dept. Of Computer Management
Institute of Management, Research & Technology
[IMRT], Nashik-2

A C K N O W L E D G E M E N T

It gives me immense pleasure to submit my Mini-project entitled as Snake Game

I am blessed with invaluable guidance of **Dr. Varsha Bhabad** for this mini-project work.Without his constant encouragement, mentoring, step by step guidance, this mini-projectwould not have seen the light of the day. He, beside a source of motivation, was always helpful in exploring the various dimensions of the subject and enriched my knowledge andlearning curve with his rich experience and professional competence. I am truly indebted for the insights I gained through focused discussion and deliberations I had with him on this mini project work.

I am highly grateful to Respected Director **Dr. D. K. Mukhedkar** and other Staff members ofthis institute for encouraging me throughout this mini-project work. His support was truly instrumental in completion of my mini-project work. I take this opportunity to thank my classmates for his consistent encouragement and best wishes for this endeavor.

I acknowledge my sincere thanks to all Friends who participated in this mini-project, without their active responses this mini-project work could not have been possible. I would also like tothank all the people who provided me with the facilities being required and conductiveconditions for my mini project.

INDEX

INTRODUCTION

The snake game was created back in 1970. In 1980, a new version of the snake game evolved. In the 1990s, Nokia had this version of the snake game on its mobile phone. The game is related to moving a snakehead by using the arrows left, right, up and down to eat an apple. When you eat the apple, the snake becomes one block bigger. If the snake hits the walls of the screen, the game is over. The score is dependent on the number of apples eaten. Below is the quote from the creator of the game on Nokia: Taneli Armanto

"I NEVER IMAGINED IT'D BECOME SO POPULAR. Maybe it was the massive distribution of the phone, the simplicity of the game or the ultimate combination of the two — but Snake became a phenomenon. People were starting to stare at their cell phones for extended amounts of time."
Armanto – Creator of Snake game on Nokia.

SCOPE AND OBJECTIVE OF THE GAME

Objectives

Particular expected archived goals:

- To learn to code 2D Snake Game using Java Swing
- Improve knowledge, experience and ability concerning with Java programming.
- Improve project management and work organizing skill regarding time management, work division, etc.

Project Scope

The main work is to write a snake game using Java programming language. This game will generate various problems related to mathematics. Player has to answer the questions by controlling a snake with keyboard keys [up, down, left, right] to eat the answer ball. Each correct answer ball eaten prolongs the snake's life.

RERUIREMENT ANALYSIS

- **FEASIBILITY STUDY**

During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that proposed system is not a burden to the company. This study can be categorized into three types. They are:

- **ECONOMIC FEASIBILITY STUDY**

  This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of system is limited, the expenditures must be justified.

  Thus, the developed system was well within the budget and this was achieved because most of the technologies used are freely available. Only customized products had to be purchased.

- **TECHNICAL FEASIBILITY STUDY**

  This study is carried out to check the technical facility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This developed system has modest technical requirements, as only minimal or null changes are required for implementing this system.

SOFTWARE AND HARDWARE REQUIREMENT

Software Requirements:

For the effective execution of this computerized system, we will require the software like:

- **Front End:** Java
- **Back End:**
- **Software Requirement**: jdk-14.0.2_windows-x64_bin

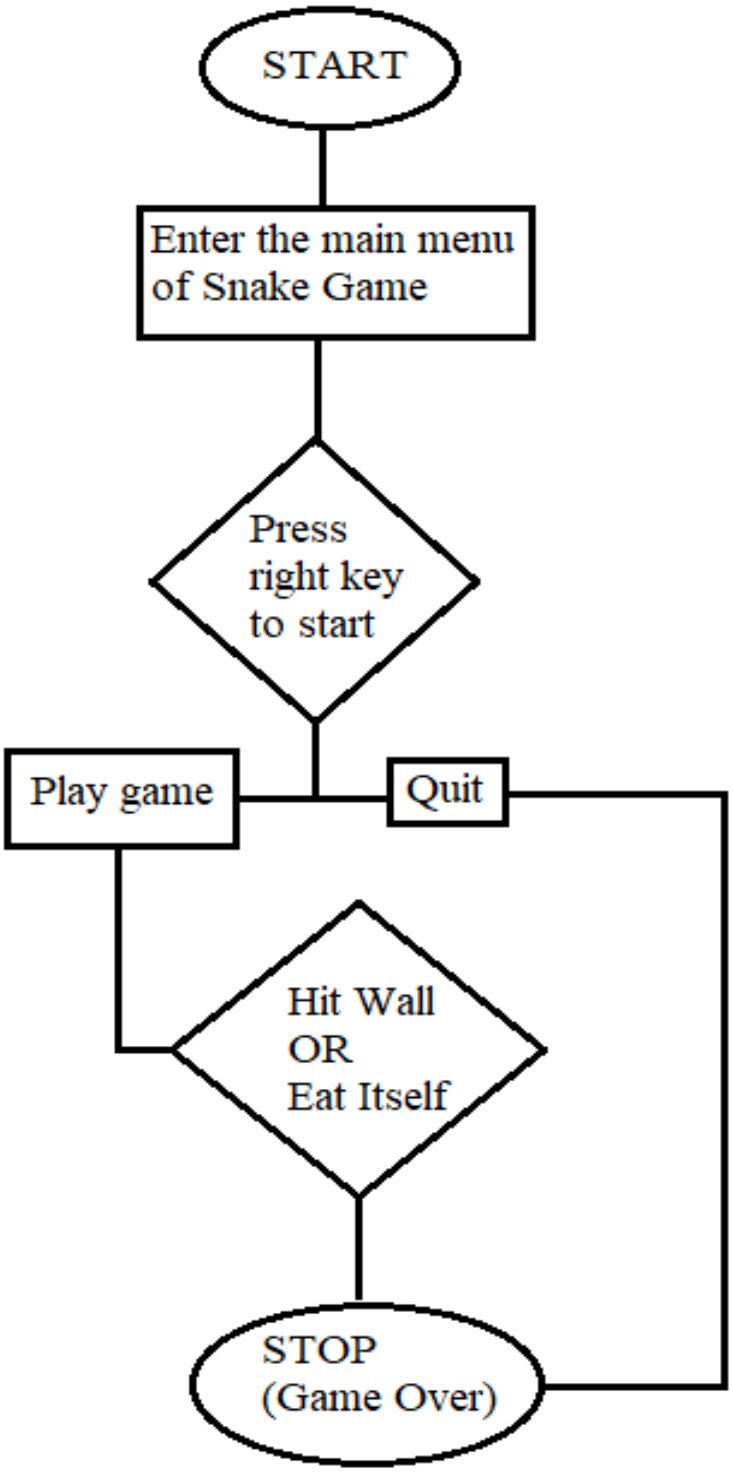
Hardware Requirements:

The hardware requirement to run this software system efficiently are as follows:

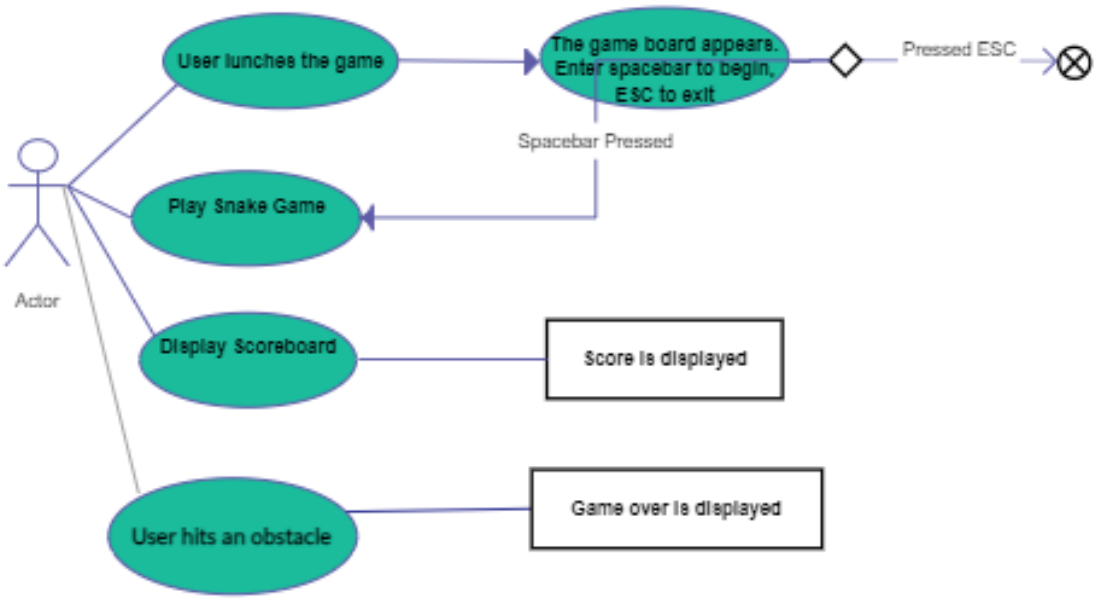
- **Processor**: Preferably 3.0 GHz or Grater
- **RAM**: 1GB or Greater
- **Web Browser**: Google Chrome, Mozilla etc.
- A print

SYSTEM DESIGN

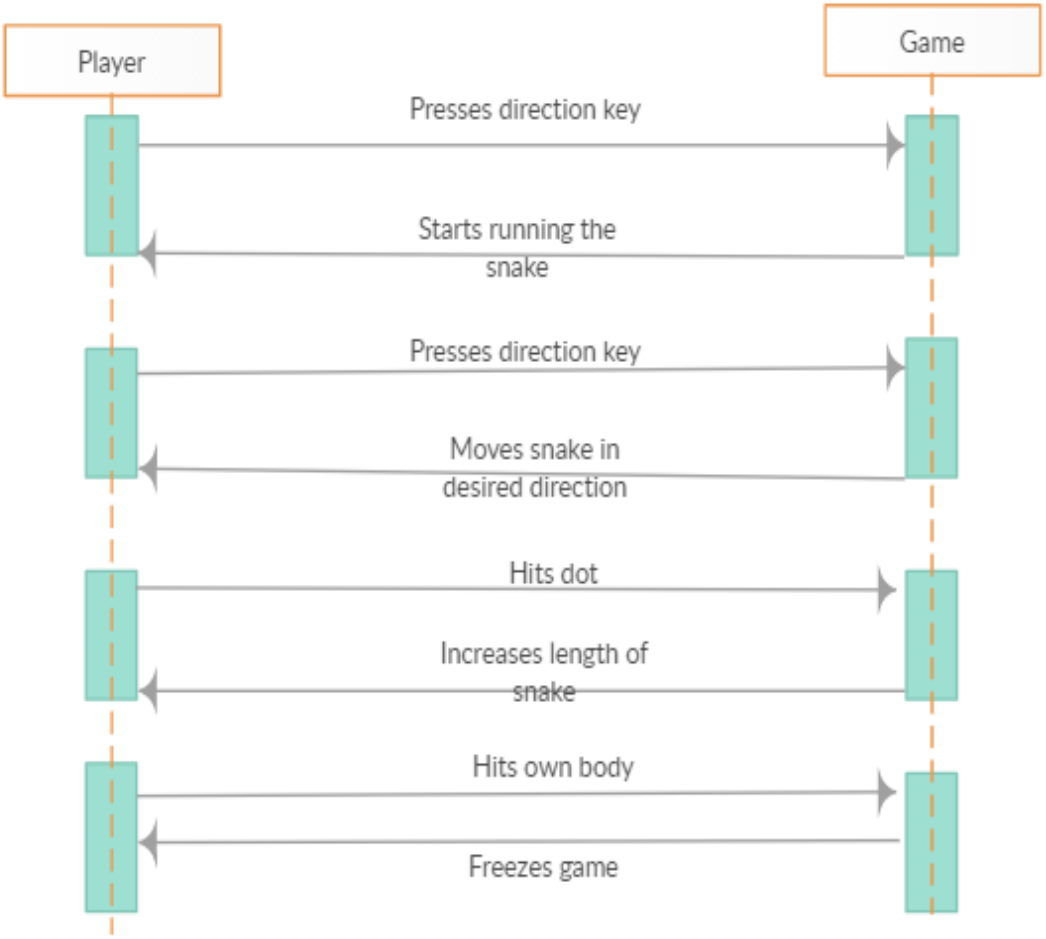
Entity Relationship Diagram:



Use Case Diagram:



UML DIGRAM:


Player
Game
Presses direction key
Starts running the snake
Presses direction key
Moves snake in desired direction
Hits dot
Increases length of snake
Hits own body
Freezes game

PLATFORM USED & LOGIC

What is JAVA?

Java is a multi-platform, object-oriented, and network-centric language. It is among the most used programming language. Java is also used as a computing platform.

It is considered as one of the fast, secure, and reliable programming languages preferred by most organizations to build their projects.

What does Java Swing mean?

Java Swing is a lightweight Java graphical user interface (GUI) widget toolkit that includes a rich set of widgets. It is part of the Java Foundation Classes (JFC) and includes several packages for developing rich desktop applications in Java. Swing includes built-in controls such as trees, image buttons, tabbed panes, sliders, toolbars, colour choosers, tables, and text areas to display HTTP or rich text format (RTF). Swing components are written entirely in Java and thus are platform-independent.

DEVELOPMENT OF JAVA SNAME GAME

Board.java

The size of each of the joints of a snake is 10 px. The snake is controlled with the cursor keys. Initially, the snake has three joints. If the game is finished, the "Game Over" message is displayed in the middle of the board.

First, we will define the constants used in our game.

The B_WIDTH and B_HEIGHT constants determine the size of the board. The DOT_SIZE is the size of the apple and the dot of the snake. The ALL_DOTS constant defines the maximum number of possible dots on the board (2500 = (500*500) / (10*10)). The RAND_POS constant is used to calculate a random position for an apple. The DELAY constant determines the speed of the game.

In the loadImages() method we get the images for the game. The ImageIcon class is used for displaying PNG images.

In the initGame() method we create the snake, randomly locate an apple on the board, and start the timer.

If the apple collides with the head, we increase the number of joints of the snake. We call the locateApple() method which randomly positions a new apple object.

In the move() method we have the key algorithm of the game. To understand it, look at how the snake is moving. We control the head of the snake. We can change its direction with the cursor keys. The rest of the joints move one position up the chain. The second joint moves where the first was, the third joint where the second was etc.

This code moves the joints up the chain.

This line moves the head to the left.

In the checkCollision() method, we determine if the snake has hit itself or one of the walls.

If the snake hits one of its joints with its head the game is over.

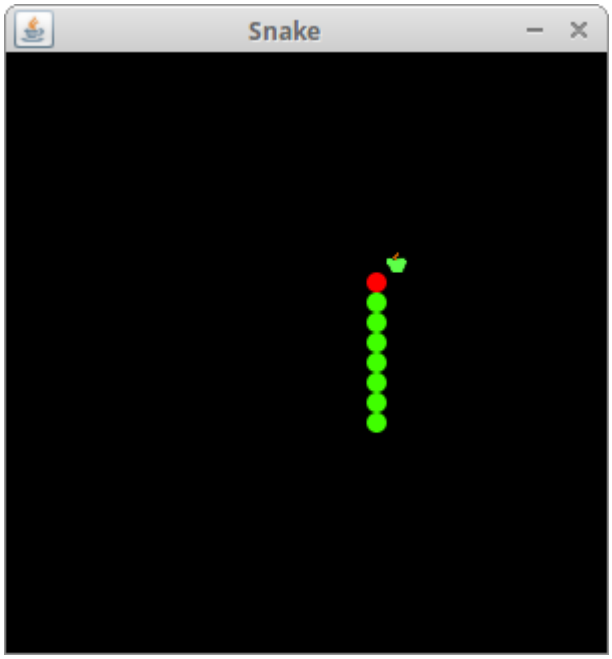
The game is finished if the snake hits the bottom of the board.

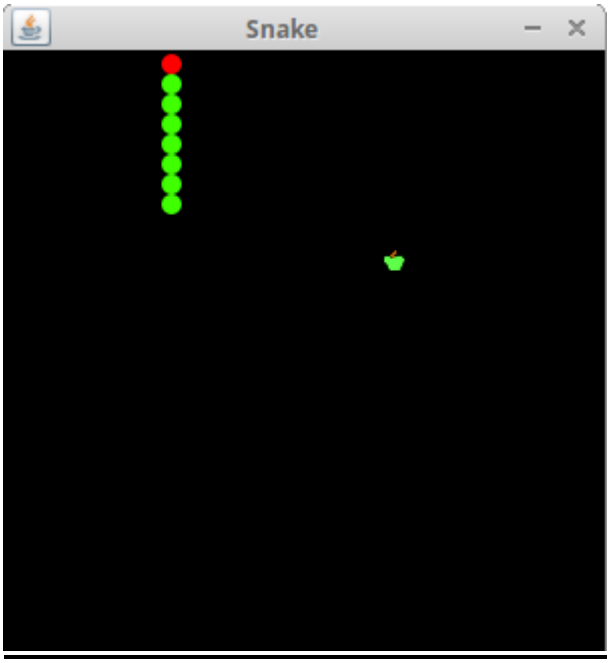
Snake.java

This is the main class.

The setResizable() method affects the insets of the JFrame container on some platforms.

Therefore, it is important to call it before the pack() method. Otherwise, the collision of the snake's head with the right and bottom borders might not work correctly.




Snake


Snake
Game Over

ADVANTAGE AND DISADVANTAGE

Advantages:

- Provides both additional functionalities and added components to AWT-replacement components
- Swing components are platform-independent.
- Swing components can use a different look and feel.
- Swing components use the Model-View-Controller paradigm (MVC) and thus can provide a much more flexible UI.
- Swing components are lightweight (are less resource-intensive than AWT).
- Swing provides built-in double buffering.
- Swing provides paint debugging support for when you build your own components.

Disadvantages:

- It can be slower than AWT (all components are drawn) as if we're not careful in programming.
- It requires Java 1.2 or a separate JAR file.
- Swing components that look like native components might not act exactly like native components.

CONCLUSION

This project gives us more thrilling, frustrating and also gives us more pleasure. It helps us in many sectors like- planning, designing, developing, managing, programming skill, and so on.

BIBLIOGRAPHY

- ✓ Class Lecture
- ✓ http://www.google.com
- ✓ http://www.wikipedia.com
- ✓ http://www.youtube.com
- ✓ http://www.w3school.com
- ✓ http://www.javapiont.com