"Gaming Portal"

PROJECT REPORT

Submitted By-

Sukhjinder Singh (RD1710A15)

Ankush Kumar (RD1710A28)

Guided By-

Mr. Balraj Kumar

Head of School(SCA)

Table of Contents

1. INTRODUCTION

1.1 Brief Introduction About the Project

This project aims to bring the fun and simplicity of snake and Brick Breaker games. It will include computer controlled opponents whose aim will be to challenge the strategy. A single player can play these games and get scores. This project explores dimension in the traditional snake game to make it more interesting and fun. The simplicity of this gaming portal makes it an ideal candidate for a minor project as we can focus on topics of java programing like graphics and abstract window tools.

1.2 Objective

This gaming portal aims to change the way people think of traditional Snake and Brick game. It will offer the experience of strategy games to the player retaining the simplicity of traditional game. The major objectives of this project are:

- Create a snake game that will have all the functionality of classic snake games.
- The Ball Game is concerned with developing a video game, where the player attempts to break through a wall by knocking out bricks with a ball, which he or she bounces on his paddle.

2. Feasibility Studies

A feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to give full comfort to the decisions makers. Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

2.1. **Technical Feasibility**

The project is technically feasible as the technology involved in the project is easily available. To deploy the application, the only technical aspects needed are mentioned below:

Operating Environment: any OS

Platform: Java

2.2. **Economic Feasibility:**

This project “Gaming Portal” is economical feasible. In this project, we use the Java language. It is economical feasible because java is free language.

2.3. **Operational Feasibility:**

This project is also operational feasible because in this project we provide the graphical user interface (GUI) which is easy to understand & operate.

3. SYSTEM REQUIREMENTS

3.1. Hardware Requirements

The following are **recommended** hardware specifications:

- Computer: Intel Pentium IV or compatible
- Memory: 512 MB
- Hard Disk: 10 GB
- Monitor
- Standard Keyboard and Mouse

3.2. Software Requirements

The following are **recommended** software specification:

- NetBeans IDE
- O/S: Window-7/8/10 32bit,64bit

3.3. Study of Tools and Technology

- Front End

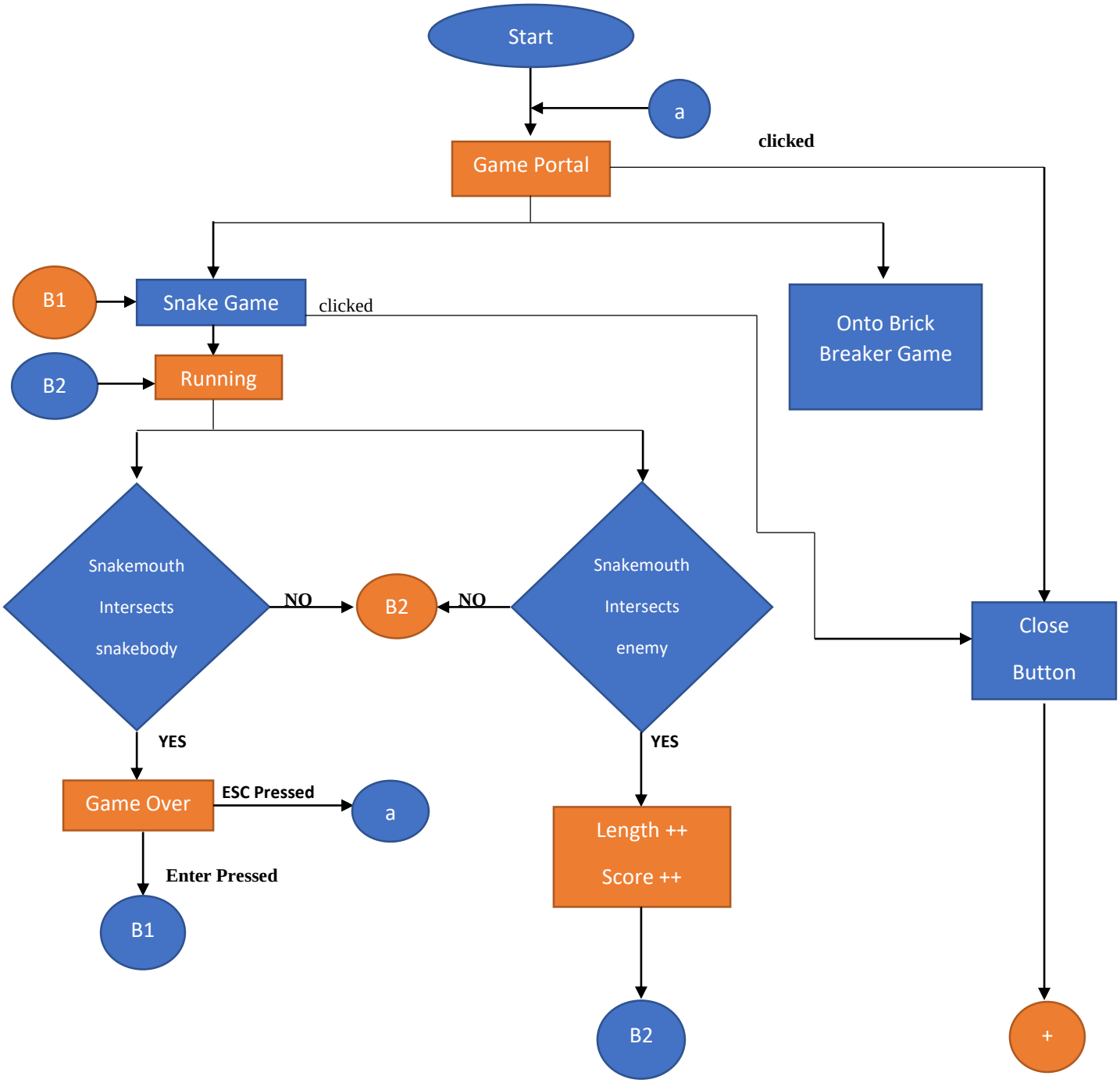
NetBeans IDE: NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, NetBeans IDE sets the standard for developing with cutting edge technologies out of the box.

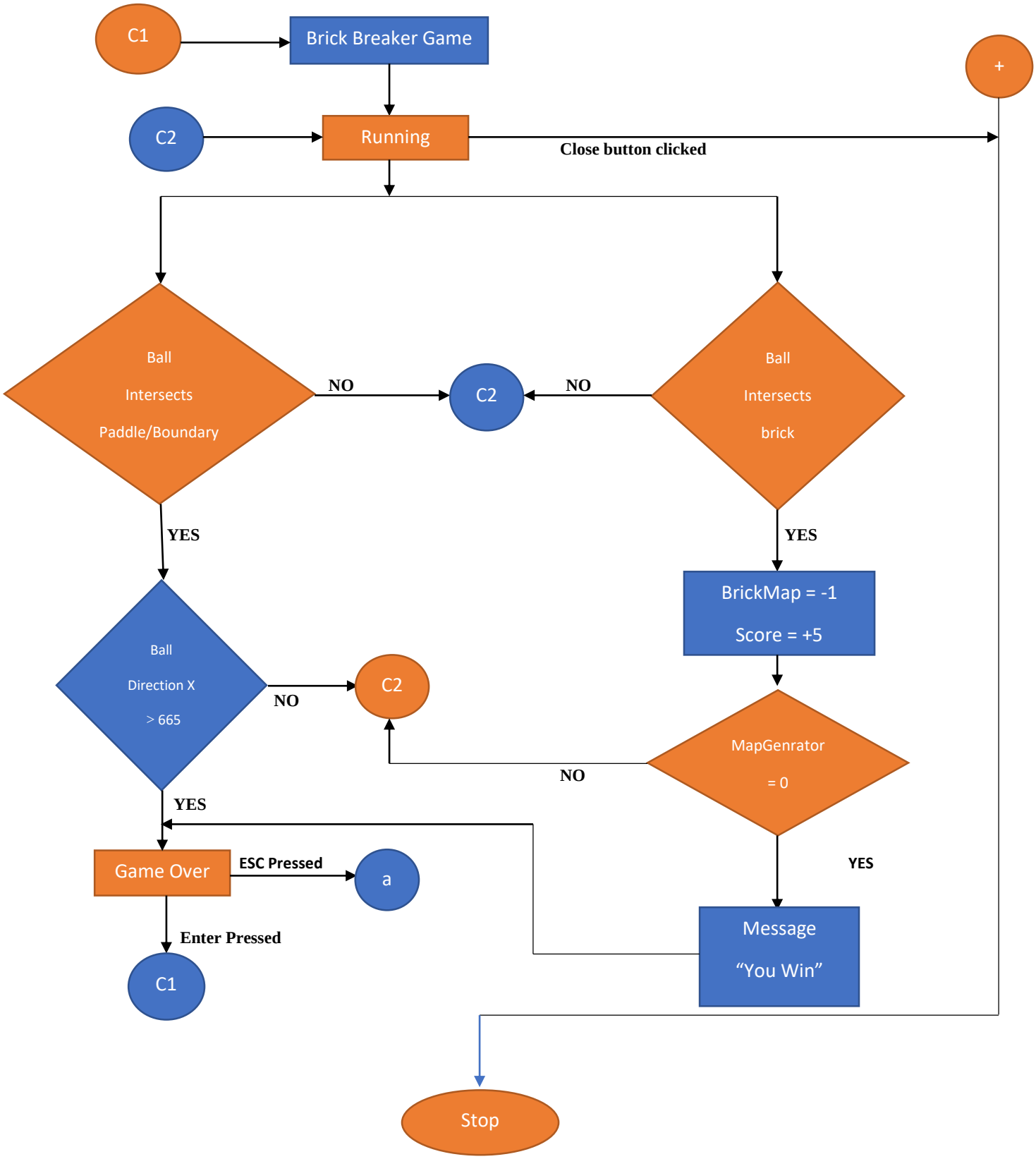
- Back End

  Java: Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

4. Design

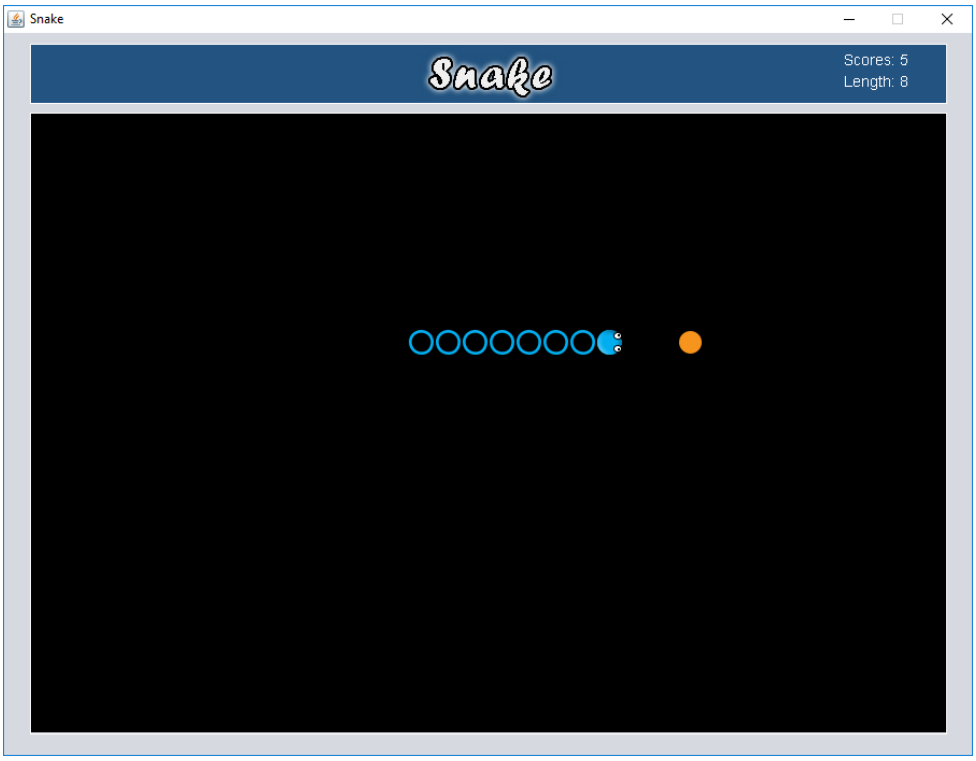
4.1. Flowchart




C1
Brick Breaker Game
+
C2
Running
Close button clicked
Ball
Intersects
Paddle/Boundary
NO
C2
NO
Ball
Intersects
brick
YES
YES
Ball
Direction X
> 665
NO
C2
BrickMap = -1
Score = +5
MapGenrator
= 0
NO
YES
YES
Game Over
ESC Pressed
a
Enter Pressed
C1
Message
"You Win"
Stop

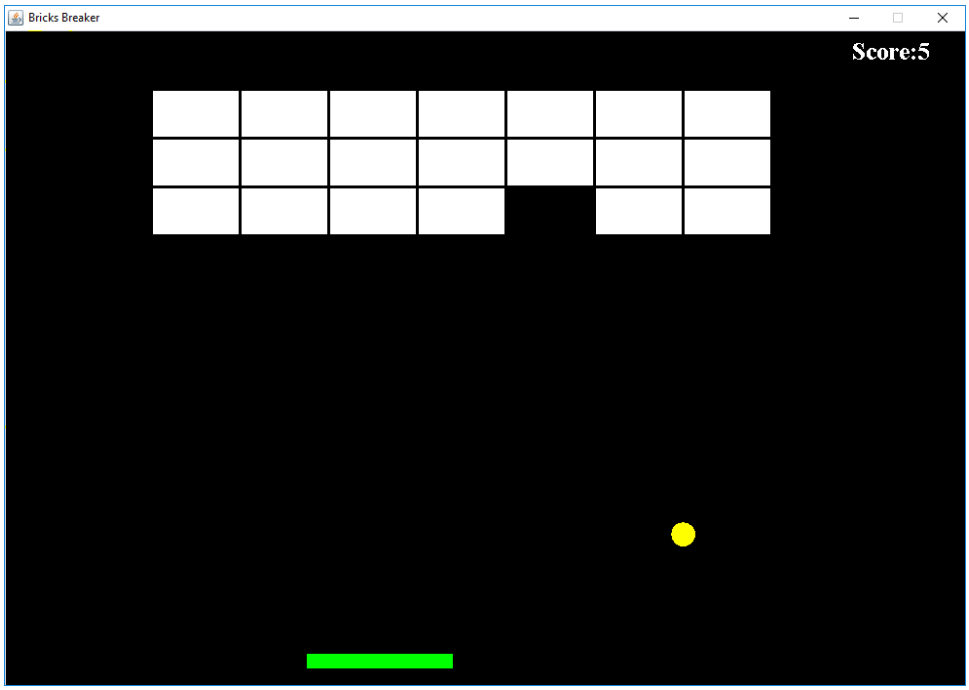
5. Screenshots

Game-selection



Snake-gameplay

Snake-game-over



Brick-breaker-gameplay

Break-beaker-game-over

6. Testing

Software testing is a process of running with intent of finding errors in software. Software testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc. Using following test cases project should be tested.

6.1. Test Cases for Frame1 class (Opening Interface)

- User can select one of the games
- On Selection of game, game should run
- On click of close button portal should close

6.2. Test Cases for Gameplay class (Brick Breaker Game Interface)

- On press of left arrow key game should start
- On press of left arrow key paddle should move 20px left
- On press of right arrow key paddle should move 20px right
- If ball intersects with paddle/boundary ball direction should change
- If ball intersects with brick then brick should be removed, and score should be increased
- If ball position X is greater than 665 than game should over
- If all bricks removed Message “You Win” should displayed and game should over
- On press ESC key it goes back to Opening Interface
- On press Enter key game should restart

6.3. Test Cases for MapGenreator class

- It should display bricks area on screen

6.4. Test Cases for Gameplay2 (Snake Game Interface)

- Game should start on any arrow key press
- Snake direction should change according to pressed arrow key

- If snakemouth intersects with enemy length of snake and scored should increase
- If snakemouth intersects with his own body game should end
- On press ESC key it goes back to Opening Interface
- On press Enter key game should restart

7. References

7.1. Book References

- JAVA: THE COMPLETE REFERENCE by HERBERT SCHILDT, MCGRAW HILL EDUCATION 7th edition

7.2. Internet References

- https://docs.oracle.com/javase/tutorial/uiswing/painting/closer.html
- https://www.youtube.com/watch?v=uZFgiqM0udA
- https://stackoverflow.com/questions/23509500/snake-game-in-java
- https://stackoverflow.com/questions/44849136/ball-in-brick-breaker-game-destroys-whole-lines-without-touching-themjava