A Report on GAME HACKATHON 2024

Titled

"GameGen: Conquer Algorithmic Challenges in Gaming using Java"

Report Made by

Hack Ninjas

Vaibhav Agarwal 2023510001

Deepak Gahalot 2023510021

Vivek Tiwari 2023510059

Harshil Vasani 2023510062

FYMCA Sem-II

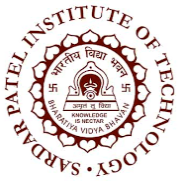
Academic Year: 2023-24

ISE-2 (DAA and JAVA)

Under the Guidance of

Dr. Aarti Karande

Prof. Sakina Salmani



Master of Computer Applications Program

Computer Science and Engineering Department

Bharatiya Vidya Bhavan's

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Munshi Nagar, Andheri (W), Mumbai – 400 058.

(Autonomous Institute Affiliated to University of Mumbai)

Date: 16th April 2024

Group Member Description along with Latest Photo

Vaibhav Agarwal 	I am a person who loves coding a lot. I like to solve question on leetcode and have keen interest in handling databases.
Deepak Gahalot 	I like to learn different technologies like java python mysql Springboot Angular Devops

Vivek Tiwari 	I am a person who loves playing games and developing games. Also intrigued by books and philosophies.
Harshil Vasani 	I am a person who loves doing mathematics.

Group Photo with Name of the group. HACK NINJAS

Table of Contents

1.Description of game and motivation

The provided code is for a Snake Game implemented in Java using Swing for the GUI. Let's break down the game and its motivation:

Description of the Game:

- **Game Objective**: The objective of the Snake Game is to control the snake, guiding it to eat fruits while avoiding collisions with walls, obstacles, and its own body. As the snake eats fruits, it grows longer, making the game more challenging.
- **Game Mechanics**:
  - **Snake Movement**: The player controls the snake's movement using arrow keys. The snake moves continuously in the direction it is facing.
  - **Fruit Placement**: Fruits appear randomly on the game board. When the snake eats a fruit, the player's score increases, and the snake grows longer.
  - **Collision Detection**: The game checks for collisions with walls, obstacles (defined in the maze layout), and the snake's body. Collisions result in the game ending.
  - **Score Tracking**: The player's score is displayed on the game screen. Additionally, the game keeps track of the player's maximum score achieved.
- **Database Integration**:
  - **Score Storage**: The game stores players' scores in a MySQL database (`snake_game_db`). Each score is associated with the player's username.
  - **Max Score Retrieval**: The game retrieves the maximum score from the database and displays it alongside the player's current score.

Motivation for Game Development:

1. **Entertainment**: The Snake Game provides a classic and enjoyable gaming experience, appealing to players of all ages.
2. **Skill Development**: Developing a game involves various programming skills, including GUI development, event handling, and game logic implementation.
3. **Learning Database Integration**: Integrating a game with a database enhances the learning experience, as it involves working with data storage and retrieval.
4. **Personal Challenge**: Creating a game like Snake presents a satisfying challenge for developers, allowing them to showcase their creativity and problem-solving skills.

By developing the Snake Game, programmers can enhance their Java skills, learn about game development principles, and gain experience in database integration, contributing to their overall growth as developers.

2.Tools used(Both frontend and backend)

Frontend (GUI Development):

1. **Java Swing**:
    - Java Swing is a set of GUI components for Java programs.
    - It provides a platform-independent toolkit for building graphical user interfaces in Java applications.
    - In the Snake Game, Java Swing is used to create and manage the game window, buttons, labels, and other GUI elements.
2. **JFrame**:
    - `JFrame` is a class in Java Swing used to create a top-level window for a Java application.
    - It serves as the main container for the Snake Game, providing the window where the game is displayed.
3. **JPanel**:
    - `JPanel` is a container that can hold other GUI components.
    - It is used to organize and manage the layout of GUI elements within the game window.
4. **Graphics**:
    - The `Graphics` class in Java provides methods for drawing graphics primitives (such as lines, rectangles, and text) onto components.
    - It is used to render the game board, snake, fruit, and other visual elements of the game.

Backend (Database Integration):

1. **MySQL**:
    - MySQL is an open-source relational database management system (RDBMS).

    - It is used to store and manage data for the Snake Game, such as player usernames and scores.
2. **JDBC (Java Database Connectivity)**:
    - JDBC is a Java API for connecting and executing SQL queries against a database.
    - It provides a standard interface for Java applications to interact with databases.
    - In the Snake Game, JDBC is used to establish a connection to the MySQL database (`snake_game_db`), execute SQL queries (such as inserting scores and retrieving maximum scores), and handle database operations.
3. **PreparedStatement**:
    - `PreparedStatement` is a class in JDBC used to execute parameterized SQL queries.
    - It helps prevent SQL injection attacks and improves performance by precompiling SQL statements.
    - In the Snake Game, `PreparedStatement` is used to insert player scores into the database.

Overall:

- Java Swing provides the frontend framework for creating the graphical user interface of the Snake Game.
- MySQL and JDBC are used for backend database integration, enabling the storage and retrieval of player scores in a MySQL database.

3.Detailed innovation description

1. Database Integration:

- **Innovation**: Storing Player Scores
- **Description**: Unlike traditional Snake Game implementations, this version integrates a MySQL database (`snake_game_db`) to store player scores. Each time a player completes a game, their username and score are stored in the database.
- **Impact**: This feature allows for persistent storage of player scores, enabling users to track their progress over time. It adds a competitive element to the game, motivating players to achieve higher scores and compete with others.

2. Max Score Display:

- **Innovation**: Retrieving Maximum Score
- **Description**: The game retrieves the maximum score from the database and displays it alongside the player's current score. This feature provides players with additional context about their performance compared to the best score achieved in the game.
- **Impact**: Players can gauge their progress relative to the highest score recorded in the game. It adds a competitive edge and encourages players to aim for surpassing the maximum score.

3. Username Input:

- **Innovation**: Personalized Gaming Experience
- **Description**: Before starting the game, players are prompted to enter their username. This feature personalizes the gaming experience by allowing players to identify themselves and track their scores in the database.

- **Impact**: Players feel more engaged with the game as they can associate their achievements with their username. It enhances the sense of ownership and encourages repeated play to improve their scores.

4. Dynamic Maze Layout:

- **Innovation**: Customizable Game Environment
- **Description**: The game features a maze layout with walls and empty spaces defined in a 2D array. This layout can be easily customized to create different game environments, offering players varied challenges and gameplay experiences.
- **Impact**: By adjusting the maze layout, developers can create diverse levels with unique obstacles, adding replay value and complexity to the game. Players encounter new challenges with each playthrough, keeping the game engaging and entertaining.

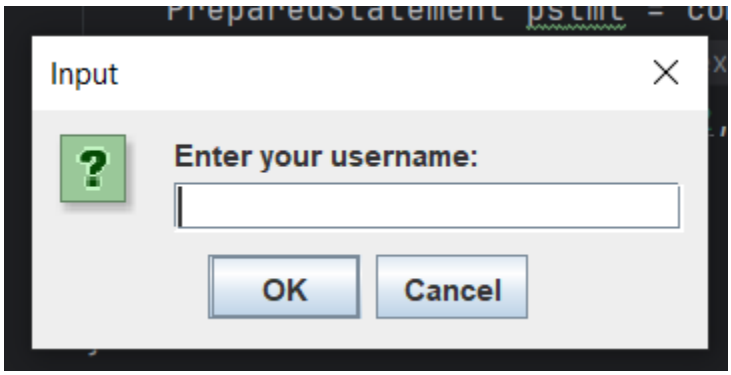
5. Interactive Game Over Dialog:

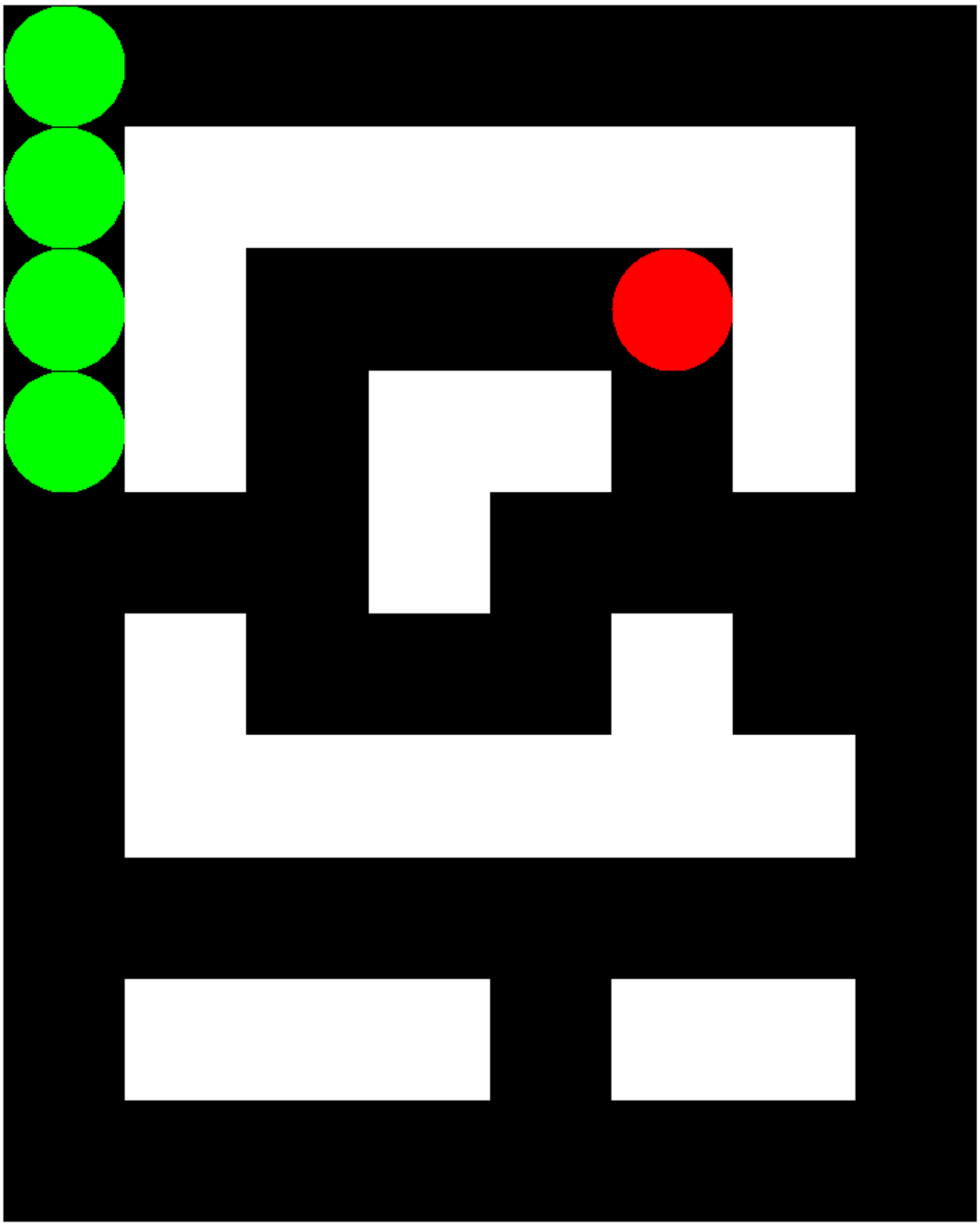
- **Innovation**: Player Interaction and Choice
- **Description**: When the game ends, players are presented with an interactive dialog box. They have the option to restart the game or exit the application based on their preference.
- **Impact**: This feature empowers players to decide whether they want to continue playing or take a break. It enhances the user experience by providing flexibility and control over the gaming session.

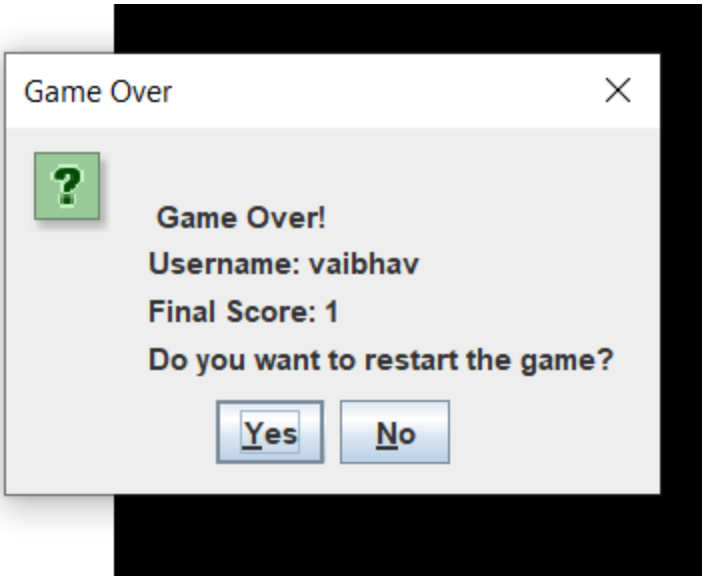
Overall Impact:

- The innovative features enhance the Snake Game experience by adding depth, personalization, and interactivity. By integrating database functionality, personalized usernames, dynamic maze layouts, and interactive dialogs, the game becomes more engaging, competitive, and enjoyable for players.

4.Screenshots






Game Over
Game Over!
Username: vaibhav
Final Score: 1
Do you want to restart the game?
Yes
No

5.References

Classic Maze Games:

- Pac-Man series (1980-present): Namco's iconic arcade game featuring maze navigation and pellet collection.
- The Legend of Zelda series (1986-present): Known for its dungeons, which often include maze-like layouts with puzzles and enemies.
- Myst (1993): A puzzle-adventure game with intricate, visually appealing mazes as part of its gameplay.

Modern Maze Games:

- The Witness (2016): A first-person puzzle game with a vast island filled with maze-like puzzles that progressively become more complex.
- Monument Valley series (2014-present): A mobile puzzle game featuring stunning visuals and Escher-like architectural mazes.
- The Talos Principle (2014): A philosophical puzzle game that includes maze elements as part of its challenges.

Online Maze Games:

- Mouse Maze (https://www.mousecity.com/games/mouse-maze): A simple browser-based maze game where you guide a mouse through a labyrinth.
- Scary Maze (https://www.scarymazegame.org/): An online prank game where players navigate a maze but encounter a scary surprise.
- Exit the Maze (https://exitthemaze.com/): A puzzle-solving game where players must find their way out of a series of increasingly difficult mazes.

Maze Game Creation Tools:

- Unity (https://unity.com/): A popular game engine that allows developers to create complex 3D maze games with custom mechanics.

- Unreal Engine (https://www.unrealengine.com/): Another powerful game development platform used for creating visually stunning maze games with advanced features.
- Phaser (https://phaser.io/): A JavaScript framework for creating 2D maze games that can be deployed on web browsers.

These references cover a range of maze game styles, from classic arcade experiences to modern puzzle adventures and online/browser-based games.