A Report on GAME HACKATHON 2024

Titled

"GameGen: Conquer Algorithmic Challenges in Gaming using Java"

Report Made by

Grey_Knights

Anand Harnekar	2023510026
Chirag Moolya	2023510036
Dharmesh Mishra	2023510035
Durgesh Mandge	2023510032

FYMCA Sem-II
Academic Year: 2023-24
ISE-2 (DAA and JAVA)

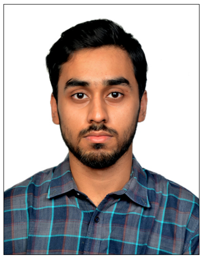
Under the Guidance of

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

Group member Name	**Description of the person (Tell about yourself which best describes you as a person and as professional)**
Anand Harnekar 	Flexible and charming artists, always ready to explore and experience something new.
Chirag Moolya 	Imaginative and strategic thinkers, with a plan for everything.
Dharmesh Mishra 	A quiet and mystical, yet very inspiring and tireless idealist.
Durgesh Mandge 	Innovative inventors with an unquenchable thirst for knowledge.

Group Photo with Name of the group.

(Grey_Knights)

Table of Contents

Description of Game

The Snake Game is a classic arcade-style game where the player controls a snake that moves around a grid-like board. The objective is to eat food pellets scattered throughout the board to grow longer. However, as the snake grows, it becomes increasingly challenging to navigate without running into obstacles, such as the snake's own body or the board's boundaries. The game ends when the snake collides with itself or the boundaries. The player's score increases with each food pellet eaten, and the game typically keeps track of the highest score achieved.Additionally, in this version of the Snake Game, players receive bonus points if they efficiently navigate to and consume food pellets using the fewest possible keystrokes, **adding an extra layer of strategy and reward to the gameplay experience.**

Motivation

The motivation behind creating and playing the Snake Game lies in its simplicity and addictiveness. Despite its straightforward mechanics, the game offers a compelling challenge that engages players of all ages. Its fast-paced action and easy-to-understand rules make it accessible to both casual gamers and those seeking a quick gaming fix. Additionally, the competitive aspect of trying to beat one's high score adds an element of replayability and encourages players to continually improve their skills. Overall, the Snake Game provides a nostalgic gaming experience while offering a fun and entertaining way to pass the time.

Tools Used

Frontend:

- Java Swing: Java's built-in GUI toolkit for creating graphical user interfaces (GUIs). It provides components like JFrame, JPanel, JLabel, etc., used to design and display the game window, score labels, and bonus messages.
- Java: The core programming language used for implementing the game logic, including handling user input, managing the game state, and controlling the snake's movement.
- Java AWT (Abstract Window Toolkit): Although not explicitly mentioned, Java AWT is often used alongside Java Swing for event handling, graphics, and user interface components.
- Timer: Java's Timer class from the javax.swing package is utilized for creating the game loop, which updates the game state and redraws the screen at regular intervals to create the illusion of motion.
- Random: The Random class from java.util package is employed to generate random coordinates for placing food pellets on the game board.
- Java AWT Event Handling: KeyListener interface from java.awt.event package is implemented for capturing keyboard input events to control the snake's movement.

Backend:

For the backend logic, it relies solely on Java without any additional frameworks or libraries.

Detailed Innovation description

The Snake Game implementation provided showcases several innovative features and design choices:

1. Bonus Points for Efficient Movement:

- One innovative aspect of this Snake Game is the introduction of bonus points for players who navigate the snake efficiently, using the minimum number of keystrokes to reach and consume food pellets.
- This feature encourages players to strategize their movements, plan ahead, and execute precise maneuvers, adding a layer of skill-based challenge to the gameplay.

2. Dynamic Difficulty Adjustment:

- The game dynamically adjusts its difficulty level as the snake grows longer. As the snake's length increases, the challenge of navigating without colliding with obstacles escalates.
- This dynamic difficulty ensures that players remain engaged and challenged throughout the gameplay session, as the game progressively becomes more demanding as they progress.

3. Real-time Feedback with Bonus Message:

- When players earn bonus points by efficiently consuming food pellets, the game provides real-time feedback by displaying a bonus message on the screen.
- This immediate feedback mechanism acknowledges the player's skillful gameplay and reinforces positive behavior, enhancing the overall gaming experience.

4. Visual Clarity and User Interface Design:

- The game's user interface design prioritizes visual clarity and simplicity, with clear labels for the score, high score, stroke count, and bonus messages.
- By presenting essential information prominently and unobtrusively, the game ensures that players can focus on gameplay without distractions, contributing to a seamless and immersive experience.

5. Optimized Resource Utilization:

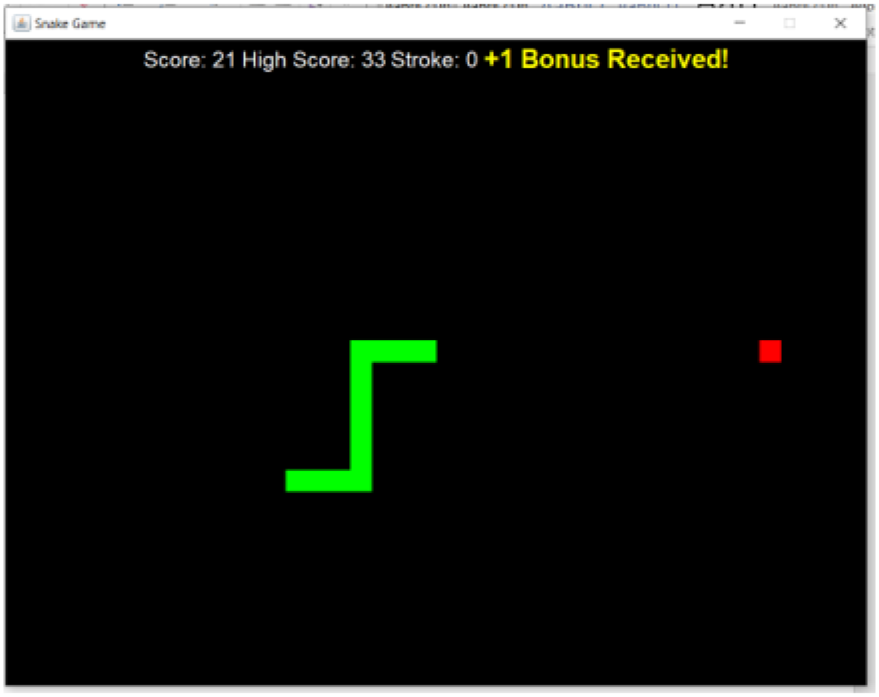
- The code demonstrates efficient resource utilization by leveraging Java's built-in libraries for graphical rendering (Java Swing) and event handling (Java AWT).
- By utilizing these standard Java libraries, the game achieves cross-platform compatibility and maintains optimal performance without relying on external dependencies or specialized frameworks.

6. Encapsulation and Modularity:

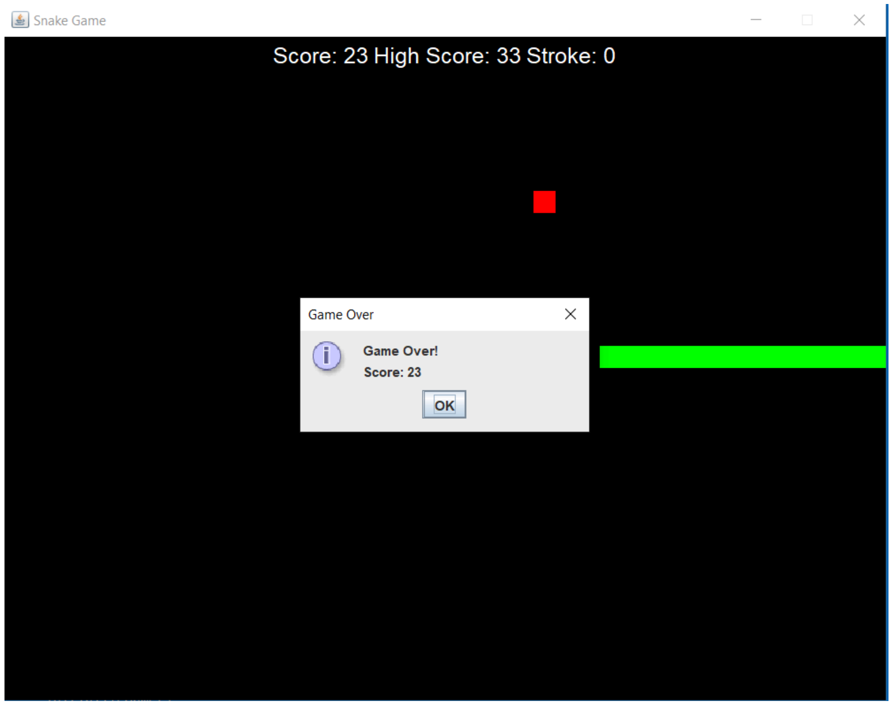
- The code exhibits principles of encapsulation and modularity, with distinct methods and classes responsible for specific functionalities such as initializing the game, handling user input, updating the game state, and rendering graphics.
- This modular design promotes code maintainability, scalability, and readability, making it easier to understand, debug, and extend the game's functionality in the future.

Overall, the Snake Game implementation incorporates innovative features, thoughtful design choices, and efficient coding practices to deliver an engaging and enjoyable gaming experience for players of all skill levels.

Screenshots



Score: 21 High Score: 33 Stroke: 0 **+1 Bonus Received!**



Score: 23 High Score: 33 Stroke: 0

Game Over!

Score: 23

References

- Tutorialspoint. (n.d.). Snake Game using Java Swing. Retrieved from https://www.tutorialspoint.com/snake-game-using-java-swing
- GeeksforGeeks. (n.d.). Snake Game in Java. Retrieved from https://www.geeksforgeeks.org/design-snake-game/ Baeldung. (n.d.).
- Numerous discussions and code snippets related to implementing the Snake Game in Java. Retrieved from https://stackoverflow.com/