A Report on GAME HACKATHON 2024
Titled
"GameGen: Conquer Algorithmic Challenges in Gaming using Java"

Report Made by

TeamSkit
Atanu Debnath 2023510014
Saurav Chavan 2023510012
Tony Thomas 2023510058
Samuel Waghmare 2023510063

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 Lates Photo

Group member Name	**Description of the person (Tell about yourself which best describes you as a person and as professional)**
Samuel Waghmare	Full Stack Developer
Tony Thomas	An aspiring web developer.
Saurav Chavan	An artificial intelligence enthusiasts.
Atanu Debnath	Tech nerd.

Group Photo with Name of the group.

TeamSkit

Table of Contents

Game Description:

The game is a classic 2D shoot them up arcade-style game where the player controls a spaceship to defend against waves of enemy ships. The player's objective is to survive as long as possible while destroying incoming enemies and avoiding enemy fire. As the game progresses, the enemies become more numerous and aggressive, posing an increasing challenge to the player.

The player's spaceship is equipped with a cannon that fires bullets to destroy enemies. Additionally, power-up items occasionally appear on the screen, providing temporary enhancements such as increased firepower, extra lives, or temporary invincibility.

The game features intuitive controls, smooth gameplay, and vibrant graphics to engage players in an immersive arcade experience. With each wave of enemies, the intensity ramps up, requiring quick reflexes and strategic maneuvering to succeed. Players can compete for high scores and aim to beat their previous records or challenge friends for bragging rights.

Motivation:

1. Nostalgia: The game taps into the nostalgia of classic arcade games, providing players with a familiar and enjoyable gaming experience reminiscent of the golden age of arcade gaming.

2. Accessible Gameplay: The simple and intuitive gameplay mechanics make the game accessible to players of all skill levels, allowing anyone to pick up and play without a steep learning curve.

3. Engaging Challenge: As the game progresses, the increasing difficulty level keeps players engaged and motivated to improve their skills, offering a satisfying challenge that encourages repeated play.

4. Competition: The inclusion of high score tracking encourages friendly competition among players, driving them to achieve higher scores and outperform their peers.

5. Escapism: In today's fast-paced world, games provide a means of escape and relaxation. This game offers a brief respite from daily stresses, allowing players to immerse themselves in a fun and engaging gaming experience.

Overall, the game combines elements of nostalgia, accessibility, challenge, competition, and escapism to provide players with an enjoyable and rewarding gaming experience that keeps them coming back for more.

Tools used (Frontend)

Based on the provided code snippets, it appears that the game is developed using Java with Swing for the graphical user interface (GUI). Swing is a GUI toolkit for Java that allows developers to create rich, platform-independent desktop applications.

Here's how Swing is utilized in the project:
- The `JFrame` class is used to create the main window of the game.
- Various Swing components, such as buttons, panels, and labels, are likely used to design the game's user interface.
- Graphics2D, a class in the Java 2D API, is used for rendering graphics elements on the screen, including player characters, enemies, bullets, and power-up items.

Swing provides a convenient and flexible way to create interactive GUI applications in Java, making it a suitable choice for developing desktop games like the one described in the provided code snippets.

Detailed Innovation description

Collision Detection:

Collision detection algorithms are likely used in the Player and Enemy classes to detect collisions between the player's character and enemies, as well as between bullets and enemies. These algorithms ensure that appropriate actions are taken when collisions occur, such as reducing player lives, damaging enemies, or destroying objects.

Movement Algorithms:

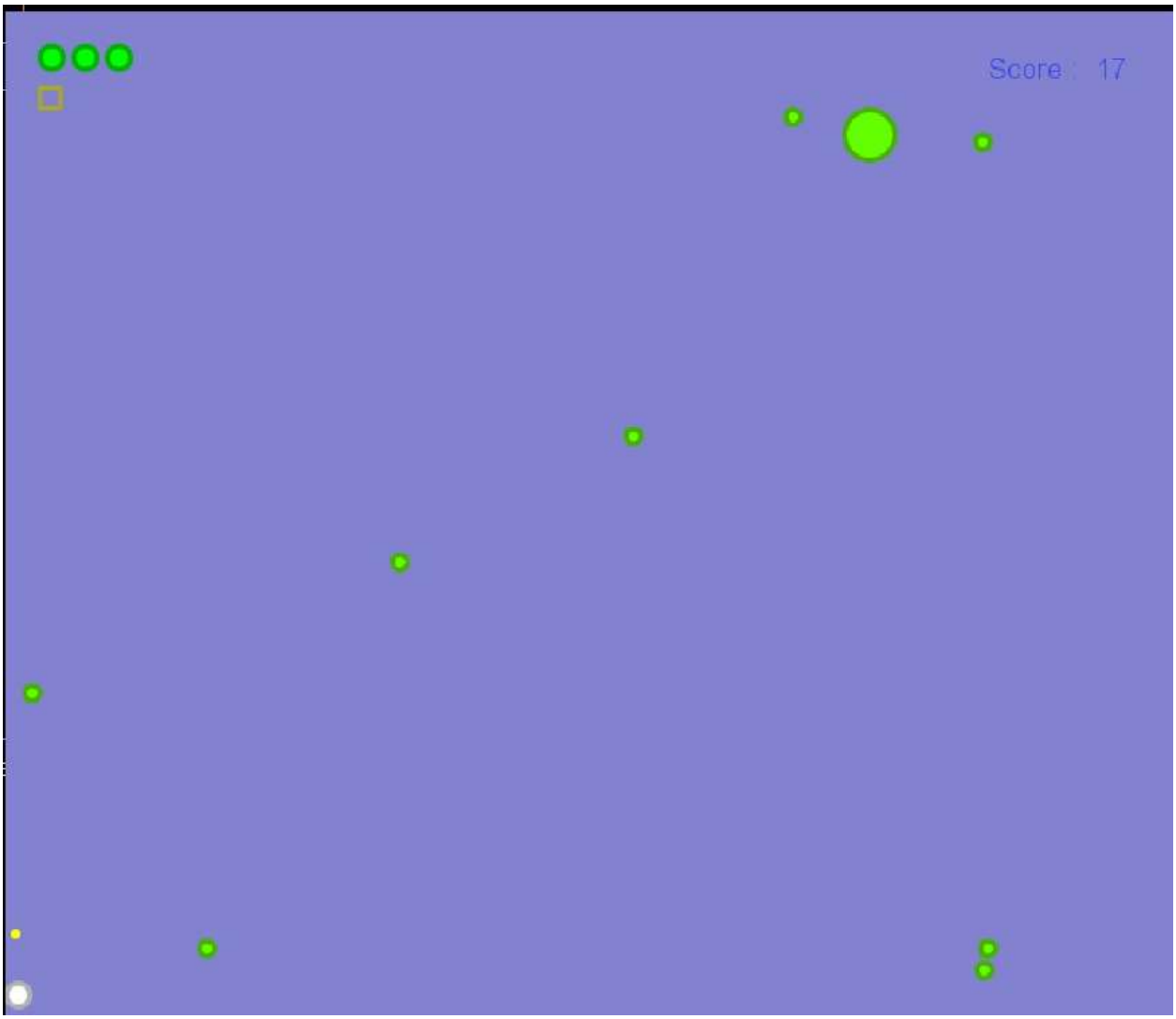
Movement algorithms are used in the Player, Enemy, Bullet, and PowerUp classes to handle the movement of game objects across the game screen. These algorithms determine the velocity and direction of movement for each object, ensuring smooth and responsive gameplay.

Rendering Algorithms:

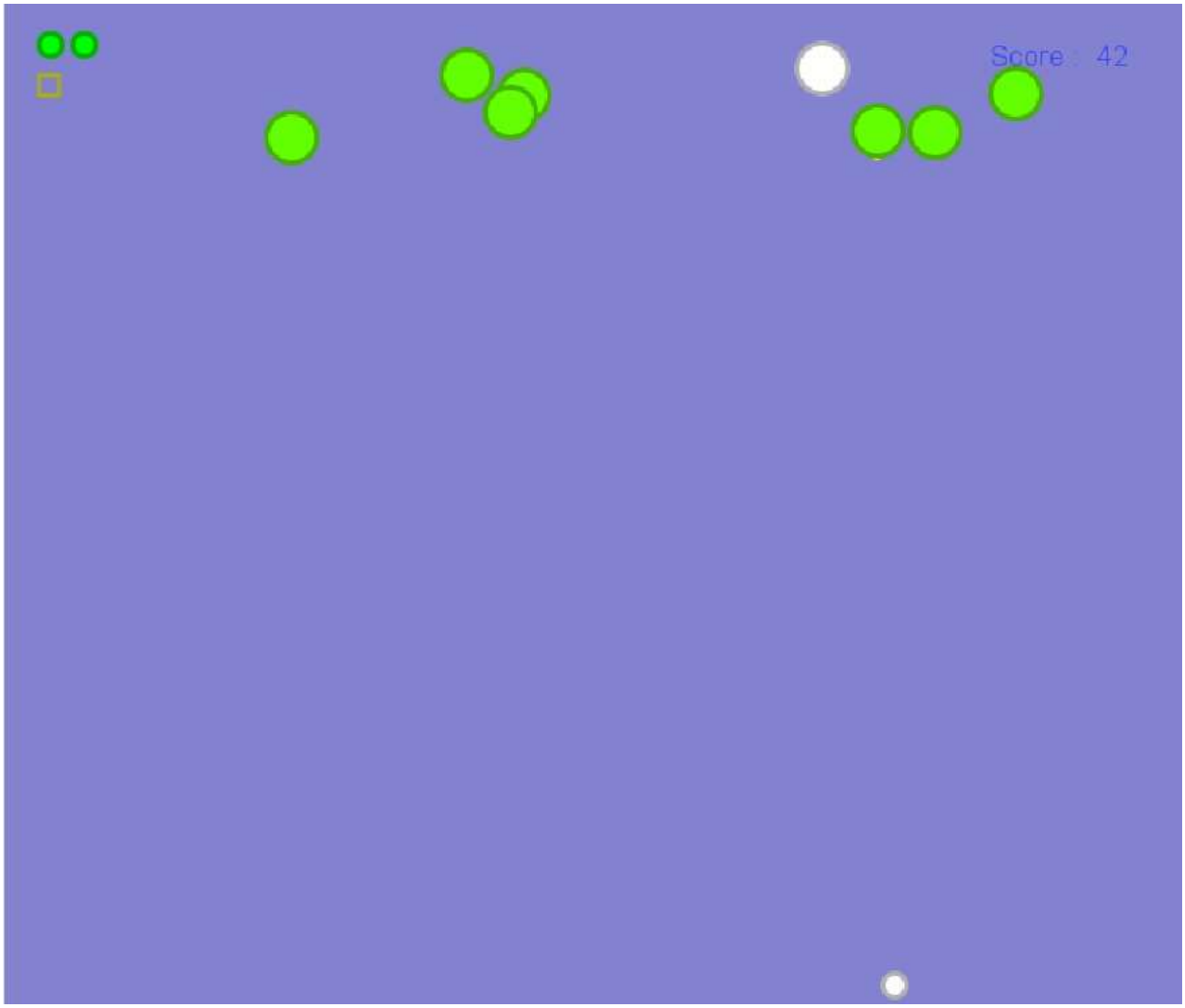
Rendering algorithms are primarily used in the Player, Enemy, Bullet, PowerUp, Explosion, Text, and possibly the GamePanel classes to render graphics elements on the screen. These algorithms utilize Java's graphics capabilities to draw sprites, shapes, text, and other visual elements, updating them as needed to reflect changes in the game state.

Screenshots

Level 1


Score: 17

Level 2



Level 3

Game Over

References

https://www.bubbleshooter.net/
https://gamedev.net/
https://www.youtube.com/watch?v=mPJO0lcnL4I