Presented to the

De La Salle University - Manila

Term 3, AY 2022-2023

In partial fulfillment

of the course

LBYCPEI

Submitted by:

CRUZ, Jose Andres A. – BS-CpE

LIM, Curt Ethan – BS-CpE

TAN, Cielbert l. – BS-CpE

Section EQ1

Submitted to:

Ramon Stephen L. Ruiz

LBYCPEI Instructor

Submitted on:

August 3, 2023

I. Introduction

The merging of technology and environmental sustainability as we move deeper into the twenty-first century has generated a fresh way of looking at how to deal with the most urgent problems of our day. Our initiative, "PacBoy," reinvents the well-known game Pac-Man to emphasize the value of trash management. It links to the Sustainable Development Goal (SDG) #12 - Responsible Consumption and Production of the United Nations and is more than simply a game. (United Nations, n.d.) It is an engaging platform that promotes eco-conscious behaviors.

The significance of this initiative is located in its unique methods. Waste management is brought to life in an engaging and collaborative way by mixing environmental education into a gaming platform. As they play the game, participants unintentionally gain knowledge about the need for appropriate waste management and its greater impact on sustainability. As a result, PacBoy goes beyond the area of being only a leisure activity and transforms into a tool for change, encouraging good waste management practices.

II. Methodology

In creating this project, we have chosen to use Java's capabilities to the best extent possible. We have largely rely on its fundamental advantages, including automated memory management, multithreading, object-oriented programming, and handling of exceptions.

We utilized Object-Oriented Programming (OOP), a key component of Java, to organize the game. With the use of this paradigm, we have been able to represent game objects like PacBoy, Roaches and garbage as distinct classes that contain their characteristics. Our code is easier to maintain because of this modular design, which also makes it easier to make improvements or updates in the future.

We also want to make use of the exception handling functionality of Java. This will assist us in keeping the game stable and consistent even in the face of unanticipated situations. As an illustration, if Pac-Man is aimed at a wall or outside the playing area, an exception will be raised and gracefully managed to prevent a total application failure.

To control the simultaneous events in our game, Java's multithreading feature, which permits the execution of numerous processes at the same time, would be essential. The player's movements, score tracking, and the ghosts' actions will all be controlled simultaneously and without interference (Eckel, 2018).

Furthermore, effective memory utilization will depend on Java's automated memory management, also known as garbage collection. The garbage collector will remove unwanted items as players go through the game, which helps to optimize speed by properly managing memory resources.

PacBoy's development process will be divided into four main stages:

1. Planning and Design: This entails carefully articulating the game rules, developing blueprints for the game's architecture.
2. Development: In this stage, the real coding process will start. This will entail writing classes, objects, methods, and control structures in Java in order to convert our idea into a playable game.

3. Testing and debugging: Following the development phase, a thorough testing process will be carried out to make sure all game mechanics function as intended. Additionally, we'll carefully examine the game for any problems or mistakes and fix them right away to ensure a smooth gaming experience.
4. Deployment: The game will be made available for people to play, participate with, and learn after comprehensive testing and final polishing.

PacBoy is a compelling illustration of how technology can be used to further sustainability as well as a testimonial to the effectiveness of Java in game creation. We intend to contribute a tiny but meaningful amount to the global movement for sustainable consumption and production by developing an engaging gameplay experience that emphasizes recycling concepts.

III. OOP ASPECTS

Our game, PacBoy, adopts a modular architecture, harnessing the power of Object-Oriented Programming (OOP) in Java. The main components include:

1. Game Objects (PacBoy, Cockroaches, Trash Items, Maze): Each of these elements will be modeled as an individual class in Java. For instance, the Pac-Man class will include attributes like position, direction, and speed, and methods to handle movements and interactions with other game objects.
2. Game Engine: This serves as the backbone of the game, managing the game loop (a continuous cycle that keeps the game moving), collision detection (checking when PacBoy interacts with other objects), and game state (keeping track of scores and lives).
3. User Interface (UI): This includes the visual elements of the game such as the game board, score display, and start menu. It also manages player input to control PacBoy's movements.
4. Data Management: This component handles saving and loading of game data, including player scores and game progress. This will enable features such as high score tracking and game resumption.

IV. UML Diagram

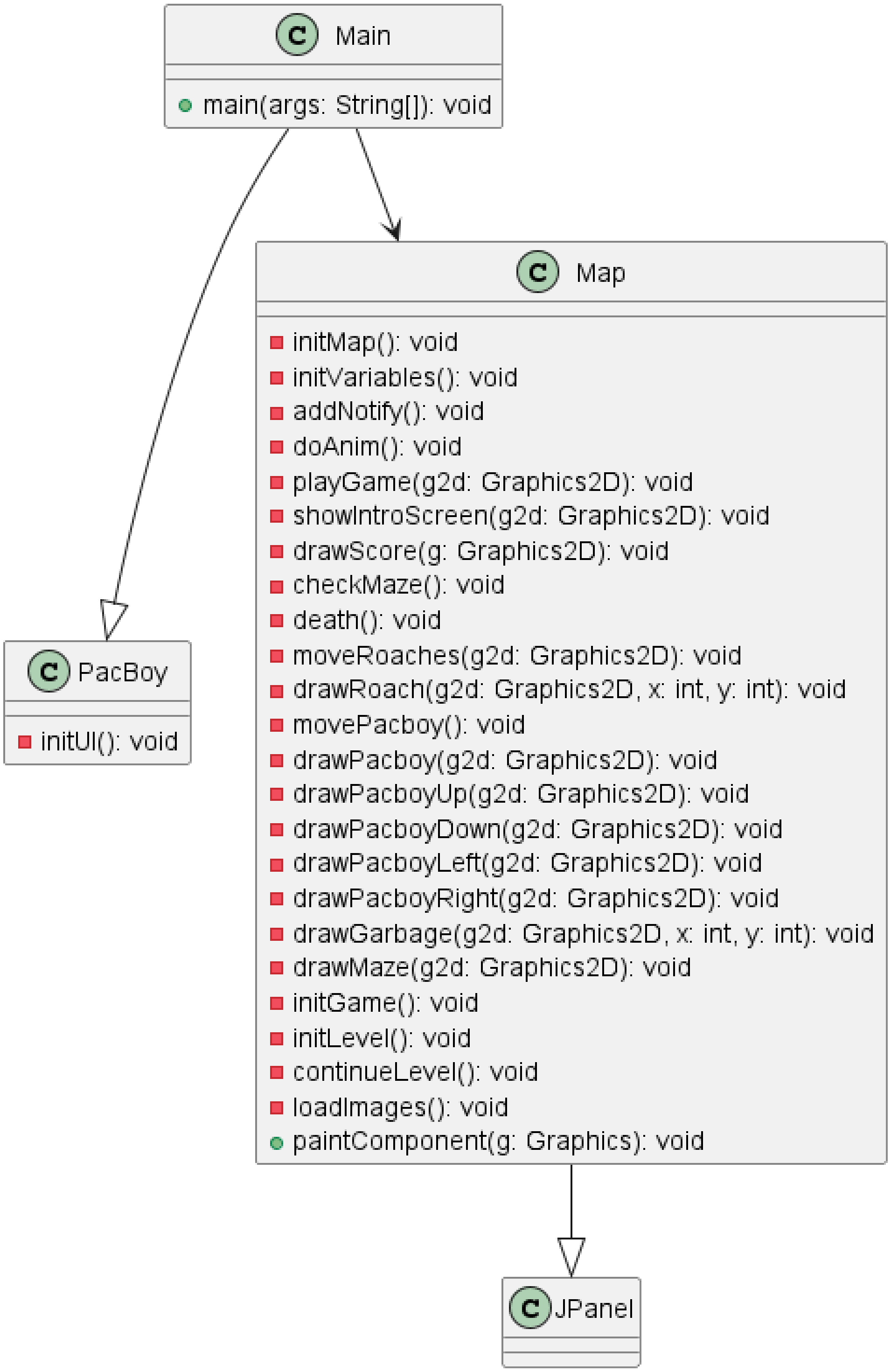
Main
+ main(args: String[]): void
Map
- initMap(): void
- initVariables(): void
- addNotify(): void
- doAnim(): void
- playGame(g2d: Graphics2D): void
- showIntroScreen(g2d: Graphics2D): void
- drawScore(g: Graphics2D): void
- checkMaze(): void
- death(): void
- moveRoaches(g2d: Graphics2D): void
- drawRoach(g2d: Graphics2D, x: int, y: int): void
- movePacboy(): void
- drawPacboy(g2d: Graphics2D): void
- drawPacboyUp(g2d: Graphics2D): void
- drawPacboyDown(g2d: Graphics2D): void
- drawPacboyLeft(g2d: Graphics2D): void
- drawPacboyRight(g2d: Graphics2D): void
- drawGarbage(g2d: Graphics2D, x: int, y: int): void
- drawMaze(g2d: Graphics2D): void
- initGame(): void
- initLevel(): void
- continueLevel(): void
- loadImages(): void
+ paintComponent(g: Graphics): void
PacBoy
- initUI(): void
JPanel

Fig 1. UML Diagram

The UML diagram represents the class structure of a Java program comprising three classes: Main, PacBoy, and Map. The Main class contains the main method, serving as the entry point of the application. The Main class has a relationship with both PacBoy and Map. The PacBoy class extends JFrame, representing the main game window, and contains the initUI method for initializing the user interface. The Map class extends JPanel, acting as the game board where the gameplay logic takes place. It includes various methods for handling game mechanics, such as moving characters (pacboy and roach), checking the maze layout, and drawing the graphics. The Map class also includes methods for game initialization, game continuation, and handling user input using the TAdapter inner class. The UML diagram provides an overview of the class hierarchy and relationships between the classes in the Java application.

V. Walkthrough/Results

A. Output of the Program

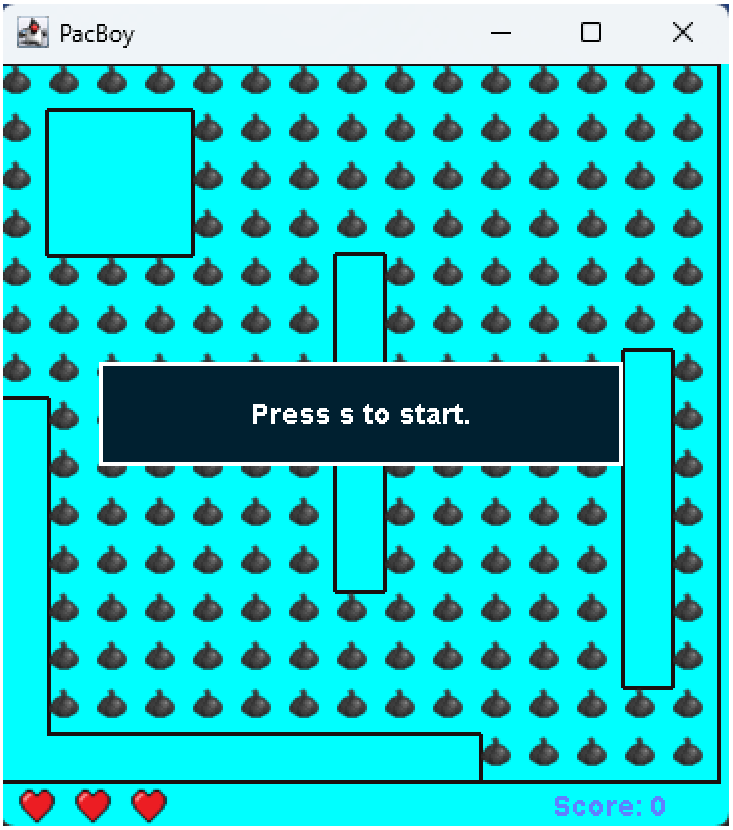

Fig 2. Starting Display of PacBoy

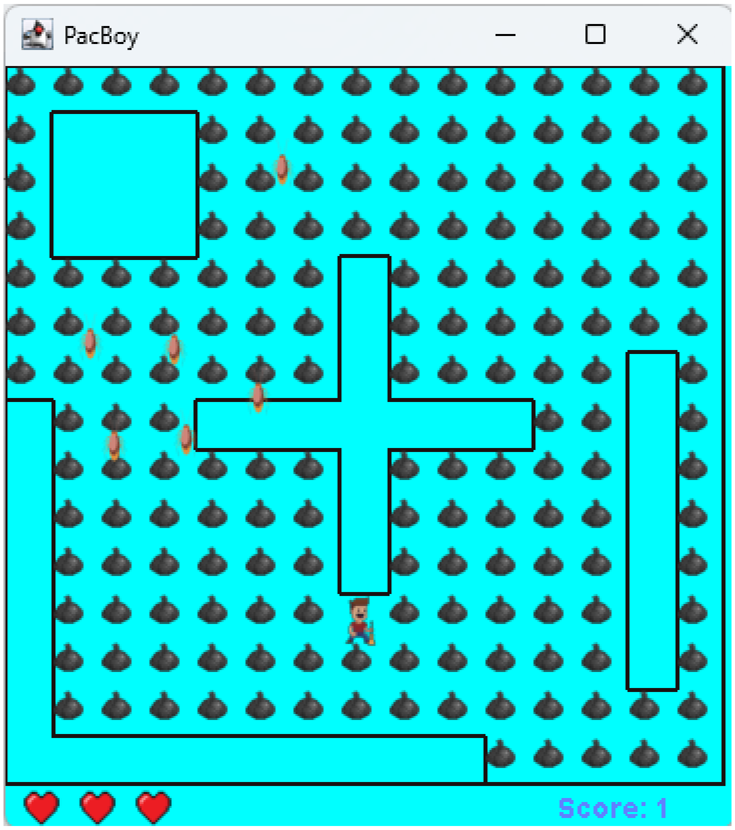

Fig 3. Display of PacBoy when started

B. Game Controls

You can control the movement of Pac-Man using the following keys:

- Up: Up arrow key
- Down: Down arrow key
- Left: Left arrow key
- Right: Right arrow key

C. Gameplay

Your task in PacBoy is to navigate the maze and collect trash. Avoid the roaches that roam around the maze. If they catch you, you'll lose a life. The game ends when you've lost all your lives or cleared all the trash items from the maze.

D. Scoring

Points are awarded per trash item you collected:

- One trash item is equal to one point

These points are continuously being added throughout the game which is also displayed in the bottom right corner

VI. Constraints

Our project also are experiencing some constraints such as it is only a one-level mini-game, the format of the map does not change, and there is no reward system built in the game code. We were unable to code it to our imagined initial plan on how our was going to look visually and be played. The initial plan was to make a three-level mini-game where PacBoy needed to collect different types of trash and be able to sort them in proper color-coordinated trash bins.

VII. Conclusion

The PacBoy project is significant for a number of reasons, the most important of which is its effective blend of gaming and environmental education. By incorporating the principles of waste management into a game format, we cater to the learning preferences of the digital generation, thereby providing a more engaging and effective educational experience (Fletcher et al., 2012).

Gaming, or the use of game design features outside of gaming situations, has been demonstrated to increase motivation and engagement (Deterding et al., 2011). PacBoy raises awareness and comprehension of sustainable waste management methods in a fun and participatory way by applying this principle to recycling.

Furthermore, research has demonstrated that video games can improve players' environmental knowledge and attitudes (Bogost, 2007). PacBoy takes advantage of this potential to build a greater understanding and appreciation for recycling, potentially pushing players to adopt more environmentally friendly activities in their daily lives.

This project is consistent with studies that suggest video games might be an effective tool for fostering sustainable behavior (Frasca, 2007). PacBoy's game elements are intended to inspire players to correctly Pick up trash as a habits that may translate into real-world activities.

The PacBoy initiative demonstrates the convergence of technology and sustainability, demonstrating how popular internet platforms may be used to address important environmental challenges (McGonigal, 2010).

Finally, PacBoy addresses the need for new, engaging ways to environmental education and the promotion of sustainable behavior. It provides an impactful platform for promoting responsible waste management by combining the power of gamification and the popularity of video games, contributing to the broader aims of SDG #12.

VIII. Contributions

Lim, Ethan Curt - Coding, Wrote project paper, Demo presentation, UML Diagram

Andres Cruz - Coding, Wrote project paper

Cielbert Tan - Editing the picture elements, Wrote project paper

IX. References

Bogost, I. (2007). Persuasive Games: The Expressive Power of Videogames. MIT Press.

Deitel, P., & Deitel, H. (2020). Java: How to Program. Pearson.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification". MindTrek '11: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 9-15.

Eckel, B. (2018). Thinking in Java. Prentice Hall.

Fletcher, R. B., & Tobias, S. (2012). The multimedia principle. The Cambridge Handbook of Multimedia Learning (2nd ed.), Cambridge University Press.

Frasca, G. (2007). Play the Message: Play, Game and Videogame Rhetoric. IT University of Copenhagen.

Larson, E. W., & Gray, C. F. (2021). Project Management: The Managerial Process (8th ed.). McGraw-Hill Education.

McGonigal, J. (2010). Gaming can make a better world. TED Talks. https://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world

Rubin, J., & Chisnell, D. (2008). Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley.

United Nations. (n.d.). Goal 12: Ensure sustainable consumption and production patterns. Sustainable Development Goals. https://www.un.org/sustainabledevelopment/sustainable-consumption-production/