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:

June 29, 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, "Pac-Recycle," reinvents the well-known game Pac-Man to emphasize the value of recycling and 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. Recycling 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 segregation and its greater impact on sustainability. As a result, Pac-Recycle goes beyond the area of being only a leisure activity and transforms into a tool for change, encouraging good waste management practices.

The project's goal is to develop a Java-based game that is both entertaining and educational. Pac-Man, the player, is tasked with navigating a maze to gather various types of garbage, which they must then separate into the proper containers. To make the game tough and interesting, the game will

have elements including a point system that rewards accurate segregation, power-ups that momentarily improve Pac-Man's powers, and progressive stages that get more difficult. The project's limits include maintaining the game's balance and enjoyment, regulating player growth without creating extreme difficulty spikes, and maintaining the emphasis on recycling as its core theme.

II. Methodology

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

We will utilize Object-Oriented Programming (OOP), a key component of Java, to organize the game. With the use of this paradigm, we will be able to represent game objects like Pac-Man, ghosts, garbage, and bins 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's many stages, which helps to optimize speed by properly managing memory resources.

Pac-Recycle'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, and classifying the recyclable items for each level.
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.

Pac-Recycle 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. Project Description

Project Architecture Overview

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

1. Game Objects (Pac-Man, Ghosts, Trash Items, Bins, Maze): Each of these elements will be modeled as an individual class in Java. For

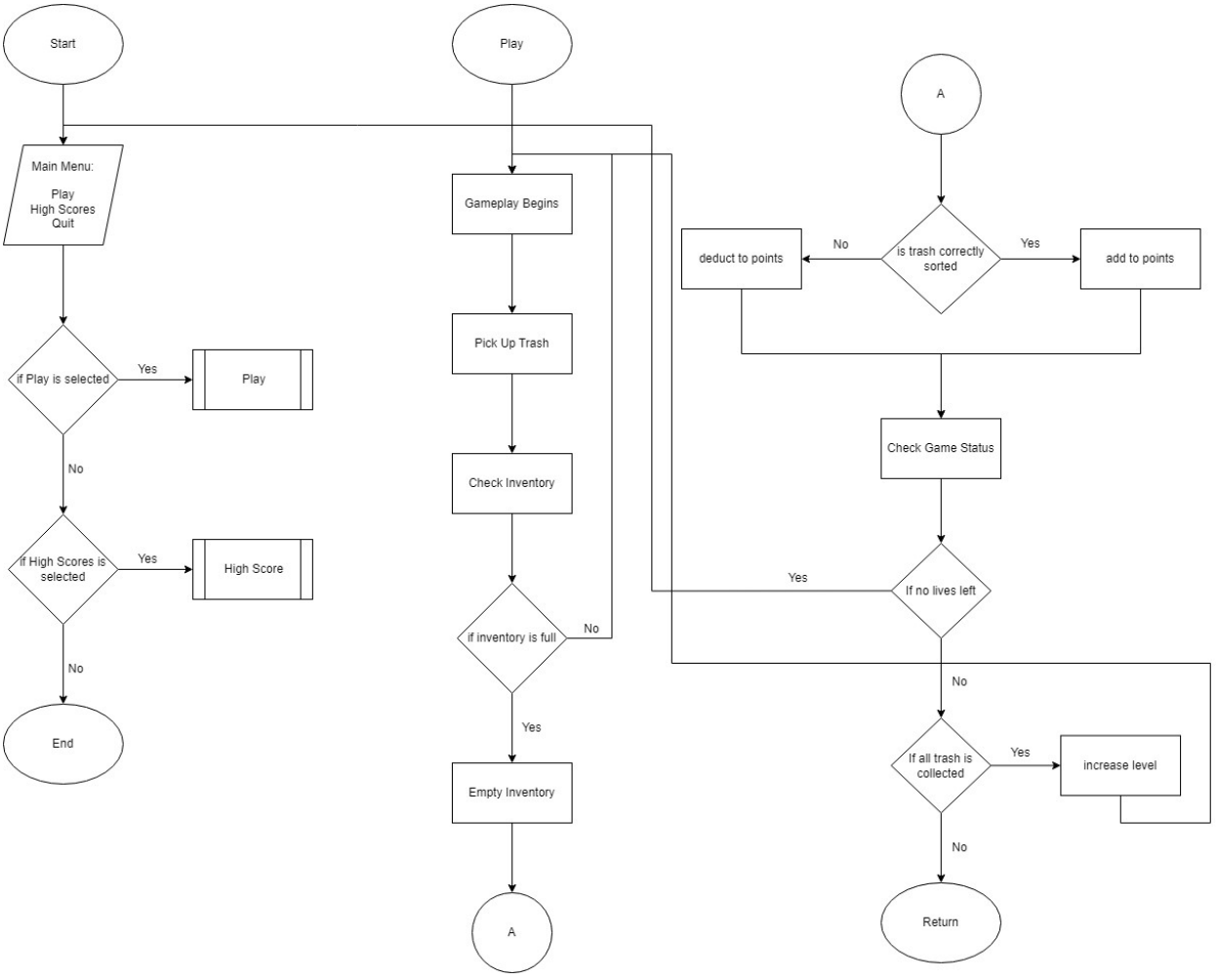
instance, the Pac-Man class will include attributes like position, direction, and speed, and methods to handle movements and interactions with other game objects.

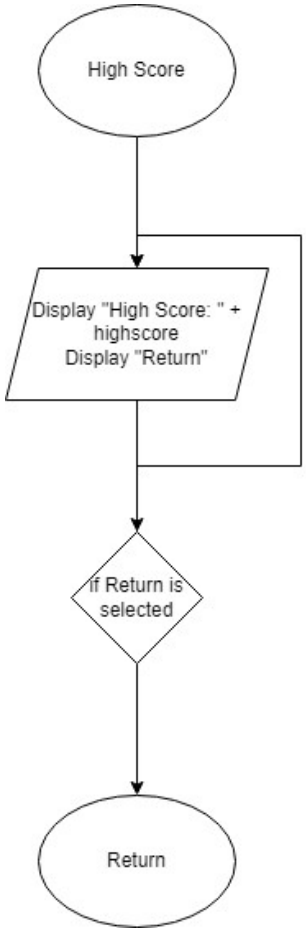
2. Game Engine: This serves as the backbone of the game, managing the game loop (a continuous cycle that keeps the game moving), collision detection (checking when Pac-Man interacts with other objects), and game state (keeping track of scores, lives, and level progression).
3. User Interface (UI): This includes the visual elements of the game such as the game board, score display, level indicator, and menus. It also manages player input to control Pac-Man'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.

Relevant Diagrams

1. IPO (Input-Process-Output) Diagram: This diagram depicts the flow of data in the game.
    - Input: Player commands (up, down, left, right), Game start signal.
    - Process: Game Engine processes player inputs and game mechanics (collisions, scoring, game progression).
    - Output: Visual representation of the game state on the UI (Pac-Man and Ghosts positions, scores, game levels).

2. Flowchart: This diagram would illustrate the flow of the game. It starts with the game launch, followed by presenting a start menu to the user. After the user starts the game, the game loop begins, continuously checking for user inputs, updating game state, and rendering the updated state on the screen. When the game ends (either by the player losing all lives or quitting), the high score, if any, is saved, and the player is given an option to restart or exit.

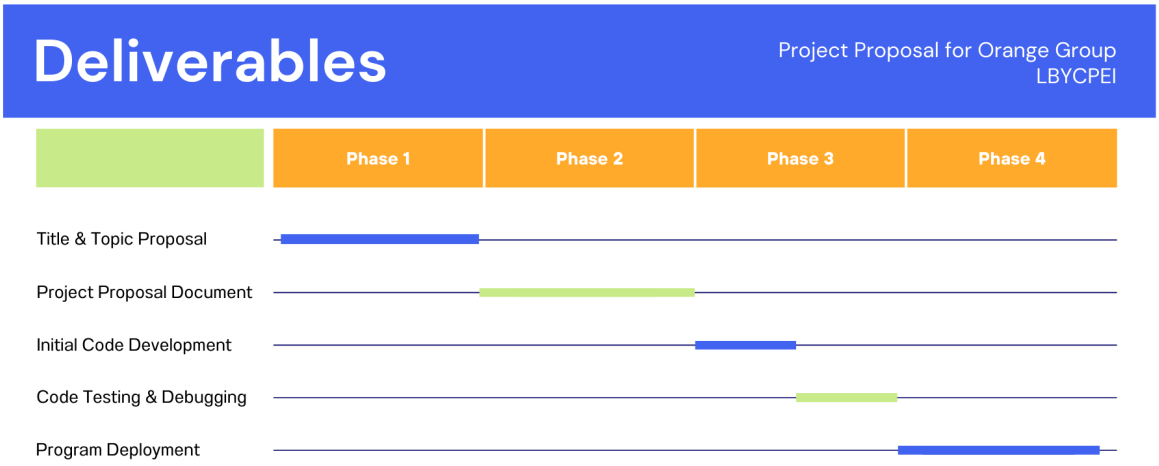




3. UML (Unified Modeling Language) Diagram: This would show the relationships between different classes in the game. For example:
    - Pac-Man class would have relationships with the Ghost, Trash, and Bin classes, as Pac-Man interacts with these objects during the game.
    - Game Engine class would be connected to all the game object classes as it manages their behaviors.
    - The UI class would also interact with the Game Engine class to display the game's state and receive player input.

These diagrams, along with the described architecture, serve as the fundamental blueprint for developing our Pac-Recycle game. This architecture encapsulates the complexity of the game mechanics, provides a solid base for managing game objects and user interactions, and allows for future expansion or modifications.

IV. Deliverables



Other Deliverables:

A. User Manual:

1. Introduction

Pac-Recycle is an exciting game that combines the thrill of the classic Pac-Man with an educational twist on recycling and waste management. In this game, you navigate through a maze,

collecting trash items and segregating them into different types of bins. This user manual will guide you on how to play and make the most of your Pac-Recycle experience.

2. Getting Started

   To start the game, simply open the application on your device. You'll be greeted by the main menu with options such as 'Play,' 'High Scores,' and 'Quit.' Select 'Play' to begin your game.

3. Game Controls

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

   - Up: 'W' or Up arrow key
   - Down: 'S' or Down arrow key
   - Left: 'A' or Left arrow key
   - Right: 'D' or Right arrow key

   You can pause the game at any time by pressing the 'P' key and resume by pressing 'R.'

4. Gameplay

   Your task in Pac-Recycle is to navigate the maze, collect trash items, and deposit them in the right bins located in the maze corners. Different types of trash items are scattered around the maze. Each trash item is color-coded, corresponding to a specific bin:

   - Green: Compostable waste

- Yellow: Recyclable waste
- Red: Hazardous waste
- Blue: General waste

Collecting a trash item will add it to Pac-Man's inventory. Once your inventory is full or you approach a bin, you can deposit the collected items.

Avoid the ghost characters 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.

5. Scoring

   Points are awarded based on the trash items you collect and deposit:

   - Picking up a trash item: 10 points
   - Depositing trash in the correct bin: 50 points

   You'll lose points if you deposit items in the wrong bin (-25 points).

6. Levels and Progression

   The game gets progressively more challenging with each level, featuring more ghosts and faster gameplay. The level increases every time you clear all the trash items from the maze.

7. Saving and Loading Game

Your game progress is automatically saved when you quit the game. To continue a previous game, select 'Load Game' from the main menu.

8. Troubleshooting

   If the game crashes or freezes, try restarting the application. If issues persist, please contact our support team.

   Have fun playing Pac-Recycle and remember: every small step towards recycling can make a big difference for our planet!

B. Technical Document

C. API Documentation

V. Evaluation

Evaluating the success of a project such as Pac-Recycle entails examining both technical performance and the game's efficacy in encouraging recycling and sustainability. Here are some criteria to consider:

Criteria for Evaluating Technical Performance:

1. Functionality: The game should function normally without crashing or freezing. All features and game mechanics should work properly.
2. Usability: The game should be simple to use, easy to browse, and understand. This can be measured through user input.
3. Performance: The game should have low latency and a high frame rate to give a smooth playing experience.

4. Reliability: The game should save progress, scores, and other data consistently and without loss or corruption.
5. Compatibility: The game must be compatible with the platforms of operating systems intended.

Criteria for Evaluating Effectiveness:

1. Player Understanding of Recycling: Surveys or quizzes could be given to players before and after the game to assess how much they have learned about recycling and trash segregation.
2. Behavioral Change: While it is difficult to directly quantify, surveys can be used to assess players' self-reported changes in recycling practices after playing the game.
3. Metrics like average session length, daily active users, and player retention rates can all be used to determine how interesting the game is for users.
4. Player Satisfaction: Game evaluations and ratings can be used to gauge satisfaction and perceived efficacy.
5. Reach: The number of downloads or users can indicate the game's popularity and, in turn, its success in spreading the recycling and sustainability message.

Using these criteria, we can ensure a thorough evaluation of the project that considers both its technical and instructional effectiveness.

VI. Conclusion

The Pac-Recycle 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 recycling and 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). Pac-Recycle 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). Pac-Recycle 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.

Furthermore, this project is consistent with studies that suggest video games might be an effective tool for fostering sustainable behavior (Frasca,

2007). Pac-Recycle's game elements are intended to inspire players to correctly classify waste, creating recycling habits that may translate into real-world activities.

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

Finally, Pac-Recycle 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.

VII. 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/