**VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY**

**THE INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**



|  |  |  |
| --- | --- | --- |
| Name | ID | Contribution |
| Hoàng Văn Mạnh | ITDSIU21099 | 25% |
| Nguyễn Xuân Vinh | ITDSIU21069 | 25% |
| Bùi Gia Phú | ITDSIU21107 | 25% |
| Đỗ Dương Thịnh | ITDSIU21121 | 25% |
| **PACMAN GAME REPORT** | | |
|

Ho Chi Minh City, Vietnam

2022-2023

**1. Table of Contents**

1. **Table of Contents**
2. **Topic Motivation**
3. **Game Rule and Game Loop**  
   3.1 Game Rule  
   3.2 Special Win Game Rule
4. **System Design**  
   4.1 List of Classes and Their Functions  
   4.2 UML Diagram  
   4.3 Demo
5. **Conclusion and References**

# **2. Topic Motivation**

# As a team embarking on our inaugural game development project, we've enthusiastically chosen to delve into the iconic world of Pac-Man. Pac-Man stands as a timeless symbol in the realm of video games, achieving global recognition since its debut. With adaptations and variations spanning across diverse cultures, Pac-Man has secured its place in gaming history.

# Originating from the creative mind of Toru Iwatani and released by Namco in 1980, Pac-Man swiftly became a cultural phenomenon. It's simple yet addictive gameplay has captivated players for decades, contributing to its widespread appeal. Pac-Man has maintained its status as a beloved classic, inspiring numerous spin-offs, merchandise, and even a dedicated fanbase.

# Pac-Man's gameplay, centered around navigating a maze, consuming dots, and avoiding ghosts, presents a rich foundation for exploration and enhancement. Our team sees this project as an exciting opportunity to not only fulfill the requirements of our university course but also to create a compelling and entertaining product. By reimagining the Pac-Man experience, we aim to infuse new dimensions of challenge and enjoyment for players while paying homage to the enduring legacy of this iconic game.

**3. Game Rule and Game Loop**

## **3.1 Game Rule**

***Pac-Man Game Rules - Brief Overview***

Pac-Man is a classic arcade game for player set in dynamic mazes. Players guide Pac-Man to eat dots, strategically avoid ghosts, and aim for a high score.

Certainly! Here's the translation of the key points into English:

**1. Game Objective:**

- Describe the main goal of the game: Guide Pac-Man to eat dots, avoid ghosts, and achieve a high score.

**2. Maze Dynamics:**

- Introduce the main features of the maze: Navigate through dots, with ghosts acting as opponents to add challenge.

**3. Starting Points:**

- Describe the initial starting points for players (e.g., 0 points).

**4. Pac-Man's Moves:**

- Describe how players control Pac-Man: Make decisions to navigate Pac-Man through the maze and collect points by eating dots.

**5. Game End Conditions:**

- Information on how the game concludes: Continue guiding Pac-Man until achieving the highest score or running out of points.

**6. Victory:**

- Describe how the winner is determined: The player with the highest score becomes the Pac-Man champion. Victory showcases adept maze navigation and strategic gameplay.

**3.2 Special Win Game Rule**

In the Pac-Man game, players can achieve victory through unique conditions beyond the standard gameplay. A player will win the game if:

1. Maze Mastery: The player strategically places Pac-Man to consume all dots on one of the four sides of the maze, showcasing dominance in that section.

2. Top Scorer: In a game with a predetermined limit of turns, victory is awarded to the player with the highest score when the specified number of turns is reached. This encourages players to maximize their points within the allotted timeframe.

These special win conditions add an extra layer of strategy and excitement to the Pac-Man game, allowing players to pursue diverse paths to victory beyond the traditional gameplay dynamics.

**4. System Design**

**4.1 List of Classes and its Functions:**

1. **Board.java :** Board is responsible for managing information about the game board.  
   Functionality:

Creates a game board with a specified size.

Manages lists of cells and players.

Provides methods to add new players.

1. **GameSound.java :** GameSound is responsible for handling sound in the game.  
   Functionality:

Plays sound from the specified sound file.

Stops the sound if necessary.

1. **Login.java :** Login is the login interface allowing players to access the game.  
   Functionality:

Displays the login interface with input fields.

Handles login events, validates login information.

1. **Pacman.java :** Pacman represents the Pacman game.  
   Functionality:

Initializes the interface and necessary components for the Pacman game.

Handles keyboard events to move Pacman on the screen.

Manages the score and game state.

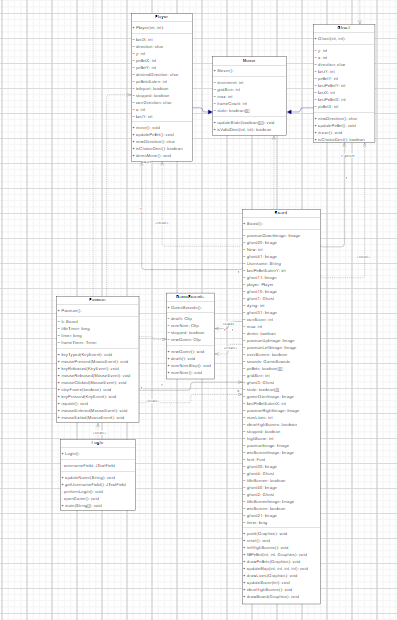
1. **highScores.txt** : highScores.txt is a file designed to store and manage high scores achieved by players in the game.  
   **Functionality:**

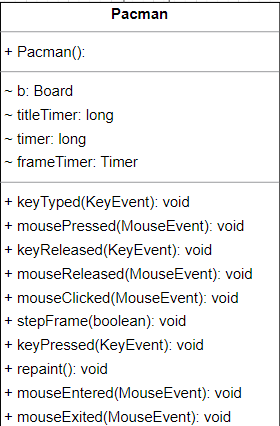
File Format: Utilizes a structured format to store player scores,  
Data Storage: Records player names and their corresponding high scores.

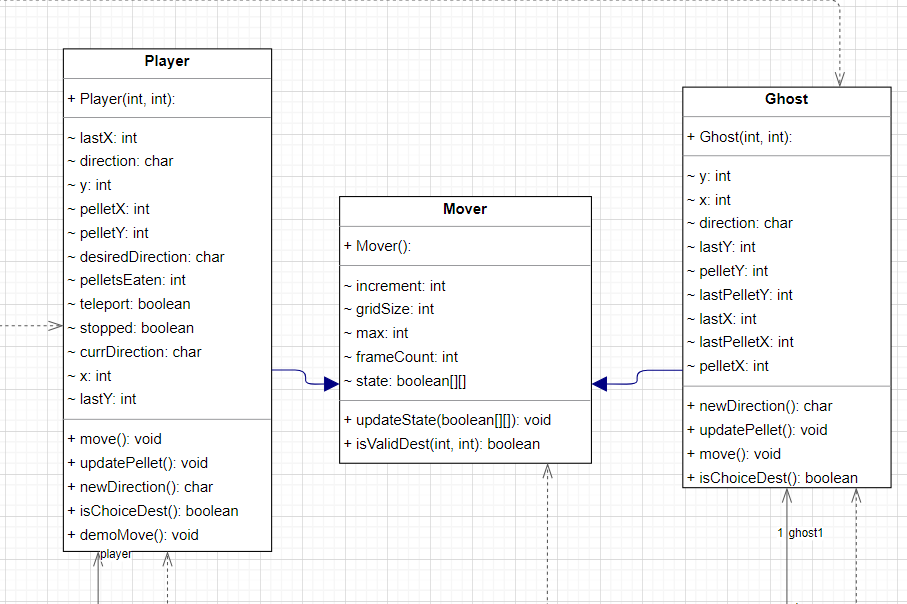
Accessibility: Ensures ease of access for the game to update, retrieve, and display high scores during gameplay or in a dedicated high scores section.

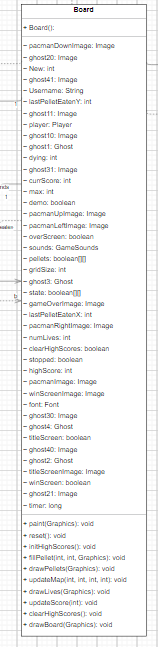
**4.2 UML Diagram**

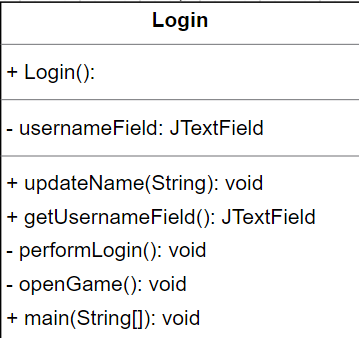
To make understanding simpler, we incorporated UML diagrams for the entire project and each mentioned group, enhancing visual clarity of the structure and algorithms.

**For the complete project**  
  
  


**For the scope : Pacman**  
  


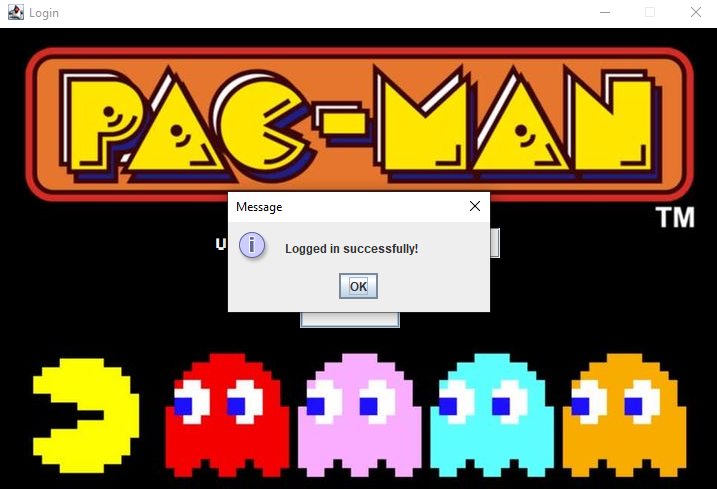
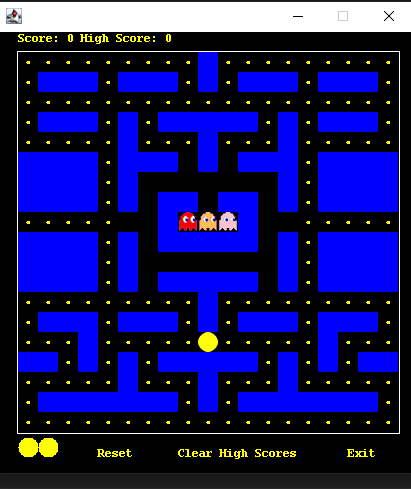
**For the scope: Mover**  
  


**For the scope: Board**  
  


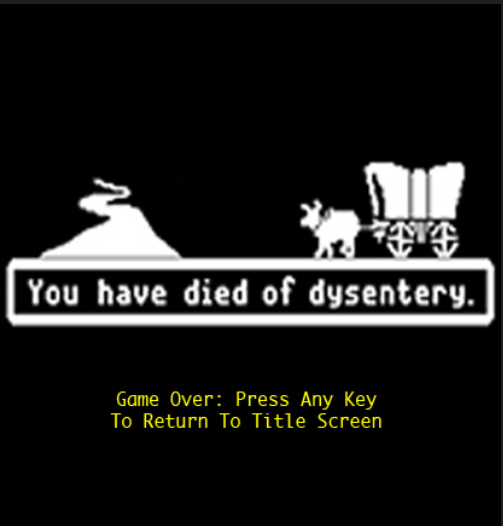
**For the scope: Login**  
  


**4.3 Demo**

To evaluate our Pacman game on a machine equipped with an IDE and Java Development Kit, clone our Git repository and compile the 'Main' class to launch the game. Presented below is a snapshot of the current game build, showcasing various game states.





**5. Conclusion**

* In wrapping up our Pac-Man game project, we're thrilled with the blend of classic arcade vibes and modern development principles. The maze's dynamic generation and interactive dice system provide an ever-changing, strategic experience. Our Chance Cards system adds spice, keeping each round fresh.
* Immersive sound effects, captivating visuals, and animated characters breathe life into the Pac-Man world. Following an Object-Oriented Approach, our game is well-organized, scalable, and embraces SOLID principles, offering a valuable learning experience.
* While our game isn't flawless, it's a labor of love aimed at marrying gaming nostalgia with contemporary coding practices. We invite players to dive in, navigate dynamic mazes, and experience the timeless joy of Pac-Man in a new light.

**Acknowledgement**  
We would like to express our sincere gratitude to our instructor and the individuals who contributed to the achievement of our project's objectives.

* Dr. Tran Thanh Tung
* MSc. Nguyen Quang Phu
* Original code from foreignguymike (ForeignGuyMike, 2018)
* The sites Geeksforgeeks, Javapoints, and so on
* The README.md template from othneildrew (Drew, 2018/2022)

**REFERENCES**



## 5.1 What can be improved next time?

* Do more research before choosing a game project, make a list of advantages and disadvantages.
* Find a high-quality source of images for the game visual style for avoiding low, pixel image lead to eyes irritation
* Add chance types: Adding more variety to the game can make it more interesting and challenging for players.
* Improve game interface: Redesign the game interface such as animation, colors, sound effects, themes to catch the player’s eyes.
* More real player(s): This project only allows one player playing with bot(s) so it must allow more than one player can play at the same time.
* Improve the mini game: Add more kind of enemy and bonus reward when player estimate enemies.