CSCI3100 Project: Pac-Man

UML Specification and UI Design

Group A8

BAI, Yuan 1155157073
BAO, Wenrui 1155157220
LI, Jianqiang 1155157143
YUE, Haoyuan 1155157271
ZHANG, Juyuan 1155160257

Spring, 2023
CSCI3100 Software Engineering
The Chinese University of Hong Kong

Contents

1 UML Design

1.1 Account Component

1.1.1 Structural Diagram

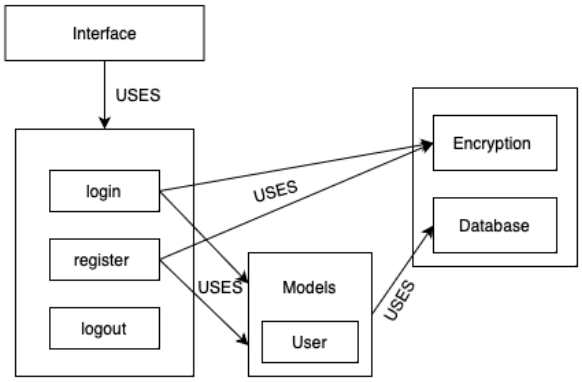

Figure 1: Structure of Account System

1.1.2 UMLs

1.1.2.1 UML Use-case Diagram

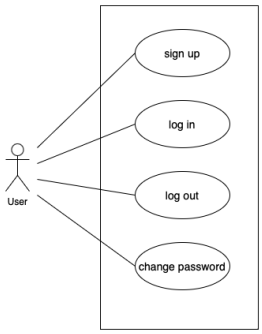

Figure 2: Use-case of Account System

1.1.2.2 UML Class Diagram

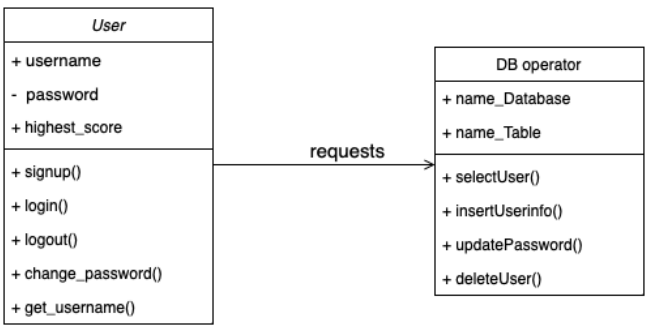

Figure 3: Class of Account System

1.1.2.3 UML Sequence Diagram

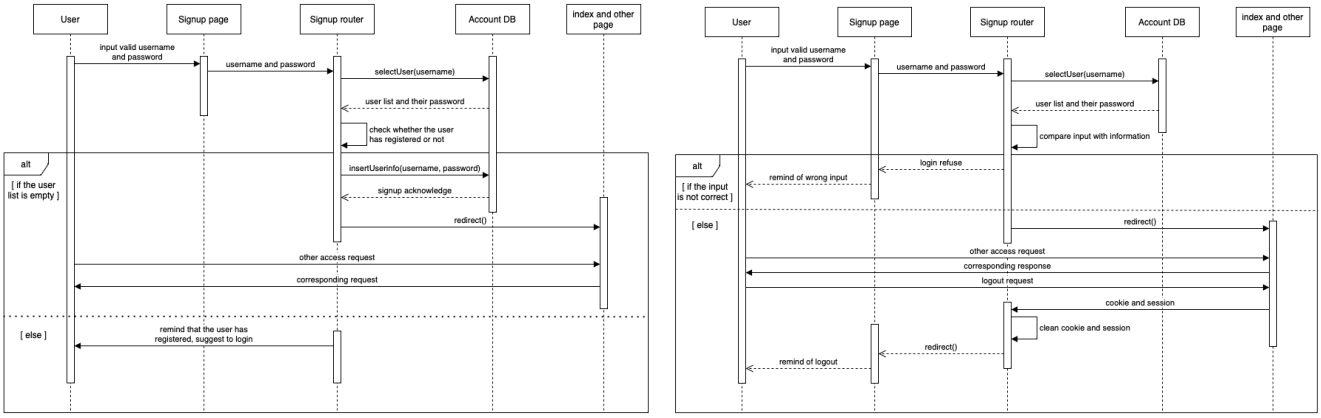

Figure 4: Sequence of Account System

1.1.2.4 UML Activity Diagram

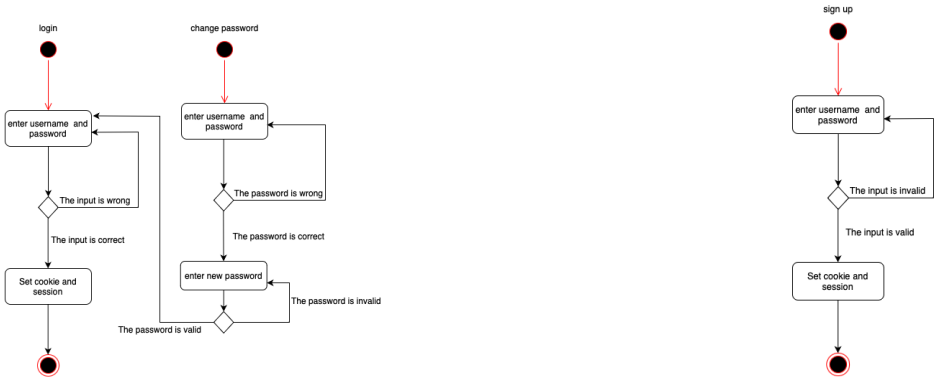

Figure 5: Activity of Account System

1.1.3 Functionality

This component deals with all the operations regarding users accounts. The model of the user is defined in this module. Once users logged in the account, all functions are available. Usernames are required in all operations in the account system. The users are required to input the password in the procedure of signup, password changing.

1.1.4 Procedures and Functions

Function	Parameter	Description
signup	username, password	Insert the information of the new user into the database. Assign a passport for the user and redirect to the index.
login	username, password	Assign a passport for the user. Redirect to the index.
logout	password	Clean all session and cookie of the user and redirect to the login page.
change_password	username, new password	Update the password in the database with the new password. Redirect to the login page.
selectUser	username	Search the user's information with the username.
insertUserinfo	username, password	Insert the valid input of user into the database.
updatePassword	new password	Search the user with the sid and update the password in the database with the new password.
deleteUser	username, password	Search the user with sid and delete all information of the user in the database.

The procedure of key functions:
Signup:
In the signup procedure, the user is required to input username and password. If the user have recorded in the database, they will be reminded of changing their usernames. After completing the signup procedure, users will be redirected to the login page.
Login:
In the login procedure, the user is required to input username and password. If the users have not registered their accounts yet, they will be reminded of going to the signup page to sign up. If the input is matched with the database, the login procedure will be completed.
Change password:
In the procedure of changing passwords, if the users have not registered yet, they can't change them. If the input password matches the database, the user can input the new password. And the new password will be updated.

1.2 Homepage Component

1.2.1 Structural Diagram

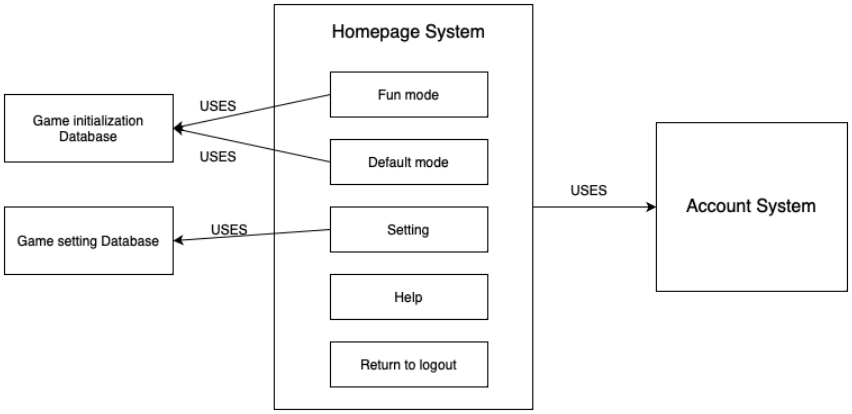

Figure 6: Structure of Homepage System

1.2.2 UMLs

1.2.2.1 UML Use-case Diagram

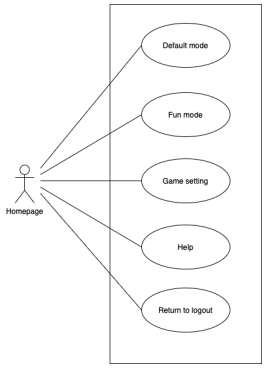

Figure 7: Use-case of Homepage System

1.2.2.2 UML Class Diagram

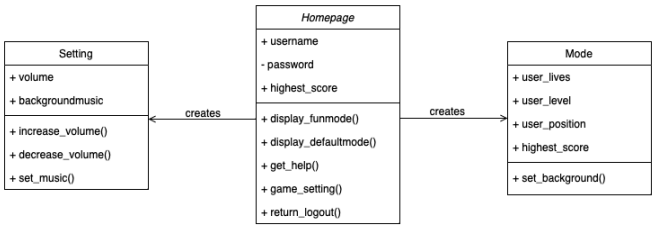

Figure 8: Class of Homepage System

1.2.2.3 UML Sequence Diagram

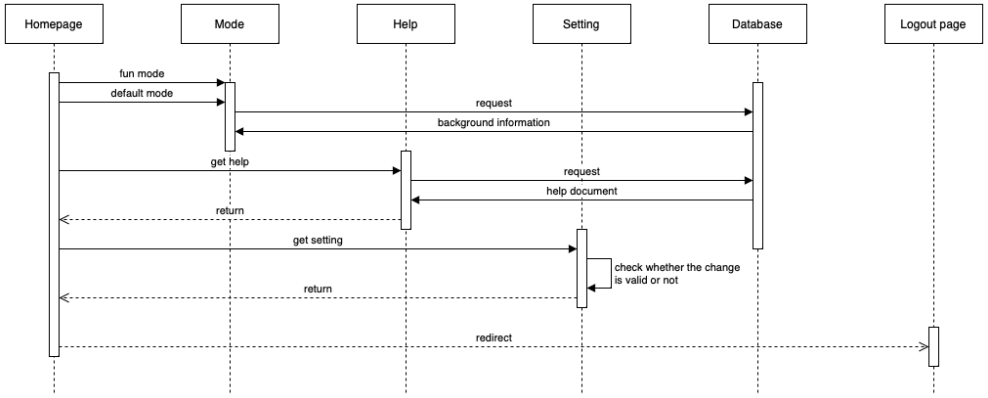

Figure 9: Sequence of Homepage System

1.2.2.4 UML Activity Diagram

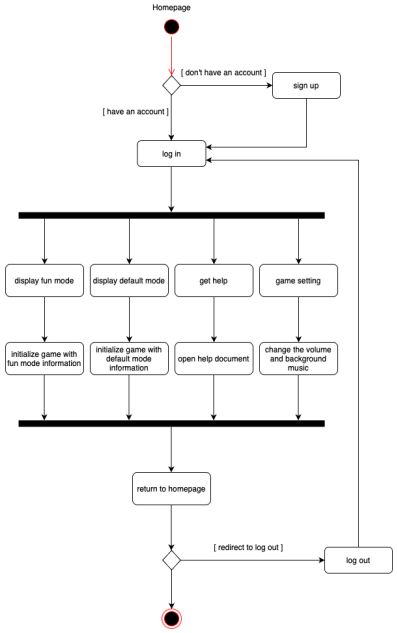

Figure 10: Activity Diagram of Homepage System

1.2.3 Functionality

The Homepage component contains all of the operations for users to start the game, ask for help, and set the game setting. The detailed functions will be stated below.

1.2.4 Procedures and Functions

Function	Parameter	Description
display_funmode	username	Redirect to Initialize the fun mode game with the fun mode background information.
display_defaultmode	username	Redirect to Initialize the default mode game with the default mode background information.
get_help	username	Open the help document.
game_setting	username	Redirect to game setting page to change the volume and music.
return_logout	username	Return to logout page.
set_background	background information	After getting information in the database, initialize background of the game.
increase_volume	volume_size	Increase the volume size of the game.
decrease_volume	volume_size	Decrease the volume size of the game.
set_music	music_name	Set the background music of the game.

The procedure of key functions:
display_funmode:
In the procedure of displaying fun mode, the user is redirected to the fun mode game, which will be initialized with fun mode background information.
display_defaultmode:
In the procedure of displaying dafault mode, the user is redirected to the default mode game, which will be initialized with default mode background information.
get_help:
In the procedure of getting help, the user could read the help document to know more about the game information.
game_setting:
In the procedure of setting game, the user could change the size of the volume and the background music.

1.3 Whole Game Logic Component

1.3.1 Structural Diagram

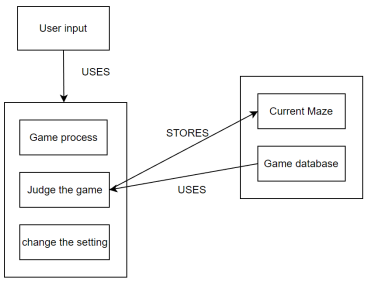

Figure 11: Structure of Whole Game Logic

1.3.2 UMLs

1.3.2.1 UML Use-case Diagram

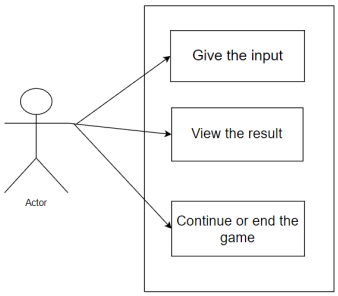

Figure 12: Use-case of Whole Game Logic

1.3.2.2 UML Class Diagram

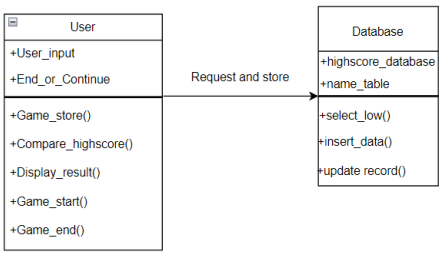

Figure 13: Class of Whole Game Logic

1.3.2.3 UML Sequence Diagram

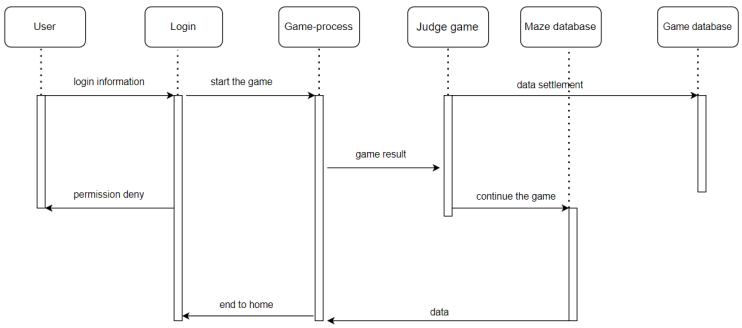

Figure 14: Sequence of Whole Game Logic

1.3.2.4 UML Activity Diagram

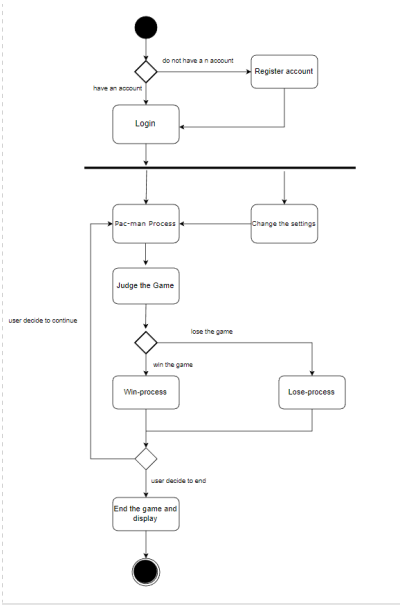

Figure 15: Activity Diagram of Whole Game Logic

1.3.3 Functionality

This component handles the process and architecture of Pac-man built-in system. The user gives the input and the whole system will give the result and display it to the user. Additionally, if the user doesn't give input, the Pac-man's default movement direction is to the right.

1.3.4 Procedures and Functions

Function	Parameter	Description
Ghost number	self	The number of Ghosts in the game board
Map number	self	Different map has different map number, the user can choose before the game start.
Token number	self	Each token was eaten will increase the user's score,user can set its number by themselves.
Ghost speed	self	The user can set the Ghost speed to adjust their game feelings.
Prop number	self	Different prop has different function, user can set the number and their categories
Set default	None	Set the background setting to the origin default

Figure 16: Functionality Diagram of Whole Game Logic

The procedure of whole process:
The game system calculates the next game result based on user input and presents it to the user. The user decides whether to continue or end the game based on the result. After each game, the score of the game will be compared with the highest scores in the database. If it is higher, the data will be updated.

1.4 Pac-Man Agent

1.4.1 Structural Diagram

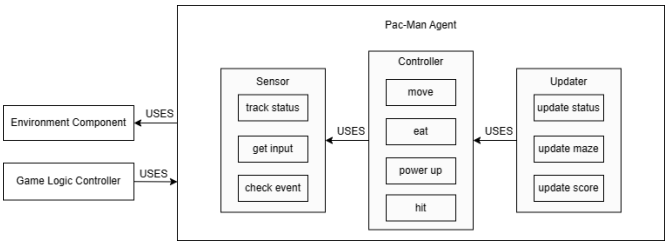

Figure 17: Structure of Pac-Man Agent

1.4.2 UMLs

1.4.2.1 UML Use-case Diagram

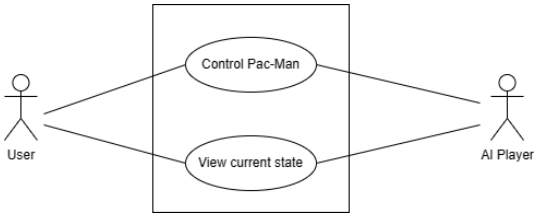

Figure 18: Use-case of Pac-Man Agent

1.4.2.2 UML Class Diagram

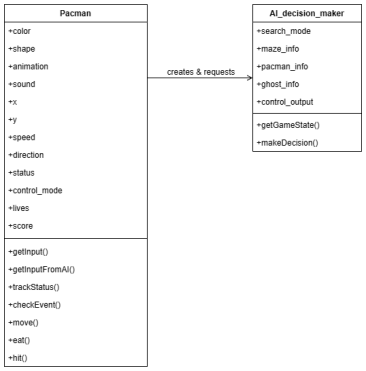

Figure 19: Class of Pac-Man Agent

1.4.2.3 UML Sequence Diagram

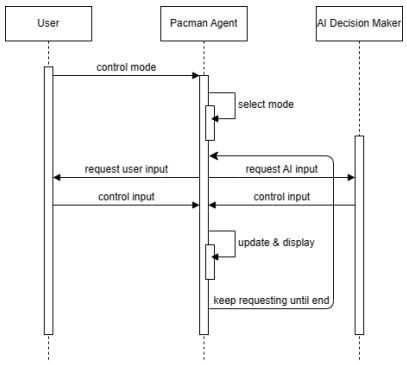

Figure 20: Sequence of Pac-Man Agent

1.4.2.4 UML Activity Diagram

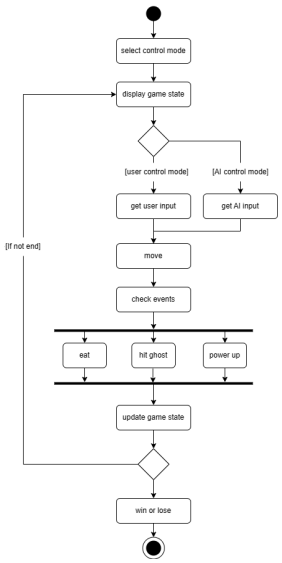

Figure 21: Activity Diagram of Pac-Man Agent

1.4.3 Functionality

This component handles the process and architecture of Pac-Man agent. During the game, user gives the input and controls Pac-Man to move, eat dots, get items, or hit by ghosts. The agent will first check whether the input is valid and then handle the movement and the following events. Additionally, before the game start the user can choose AI mode, where the control input will be provided by AI decision maker.

1.4.4 Procedures and Functions

Some key functions and procedures of Pac-Man agent are listed as follows:

getInput or getInputFromAI: The agent request control input from either user or AI decision maker. For user control, the function further checks whether the input is valid. For AI control, the function will additionally pass the current game state as parameters and the return input won't be checked.

trackStatus: This function tracks the status of Pac-Man and return a pre-defined status value like power-up or weak. Different status will affect the event handling.

checkEvent: The function checks the current game state and returns a pre-defined event value like eatingDot or hitGhost. Then according to the event type, different functions will be called.

AI decision maker: In the AI mode, AI decision maker will be called to play game. There are several search modes to choose. At each step, the Pac-Man agent fetches the current game state to AI decision maker and gets a valid control input from it via getInputFromAI() function.

1.5 Ghost Agent

1.5.1 Structural Diagram

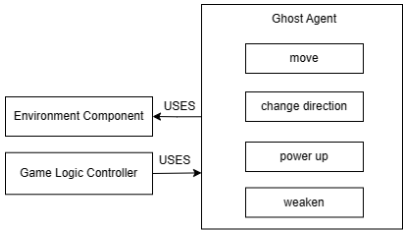

Figure 22: Structure of Ghost Agent

1.5.2 UMLs

1.5.2.1 UML Class Diagram

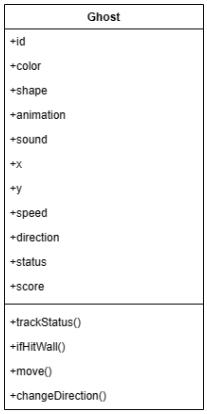

Figure 23: Class of Ghost Agent

1.5.2.2 UML Sequence Diagram

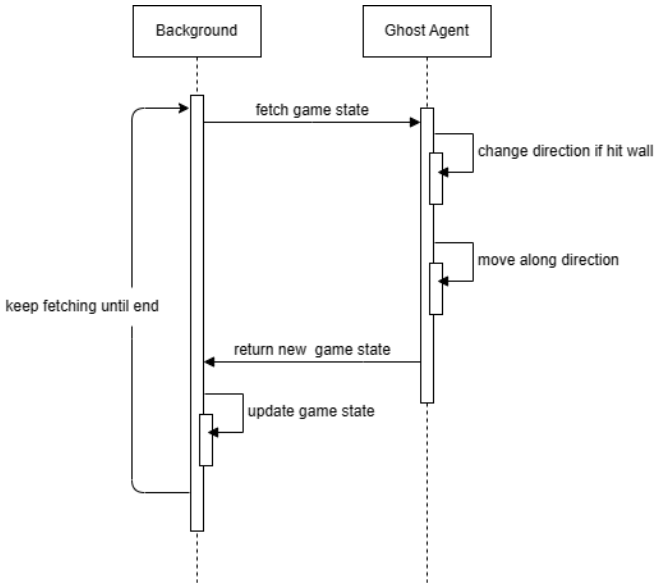

Figure 24: Sequence of Ghost Agent

1.5.2.3 UML Activity Diagram

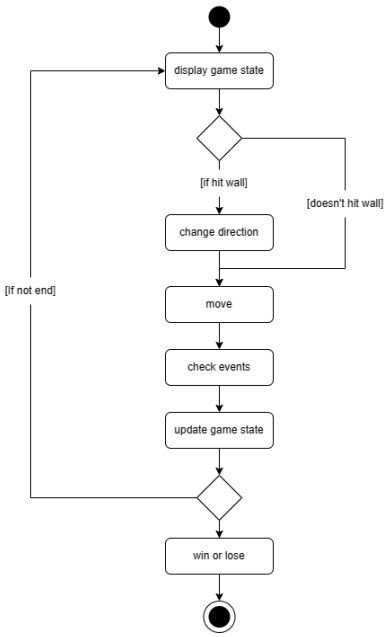

Figure 25: Activity Diagram of Ghost Agent

1.5.3 Functionality

This component handles the process and architecture of ghost agent. Every ghost carries a unique id and acts individually during the game. At each step, each ghost agent will first check whether it hits the wall. If so, it will change its direction to chase the Pac-Man. If not, it will just move along its current direction.

1.5.4 Procedures and Functions

Some key functions and procedures of ghost agent are listed as follows:

trackStatus: This function tracks the current status of ghost and returns a pre-defined status value like dead or normal. In the fun mode, the ghost may have more BUFFs or DEBUFFs which will alse be represented by status.

ifHitWall: This function checks whether the ghost hits a wall and returns a bool value. If true, changeDirection() will be called.

move: At each step, the ghost should move along its current direction. The function is called after ifHitWall() check and potential changeDirection() call.

1.6 Background Setting Component

1.6.1 Structural Diagram

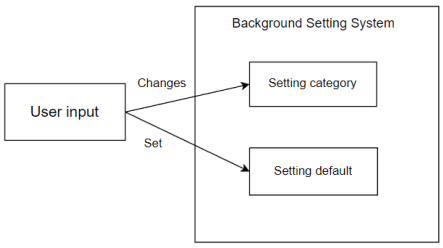

Figure 26: Structure of Background Setting

1.6.2 UMLs

1.6.2.1 UML Use-case Diagram

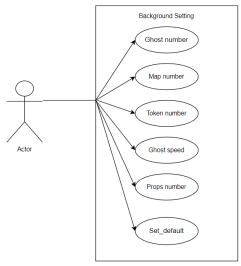

Figure 27: Use-case of Background Setting

1.6.2.2 UML Class Diagram

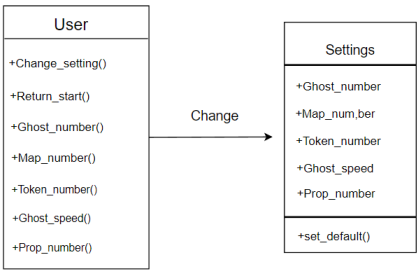

Figure 28: Class of Background Setting

1.6.2.3 UML Sequence Diagram

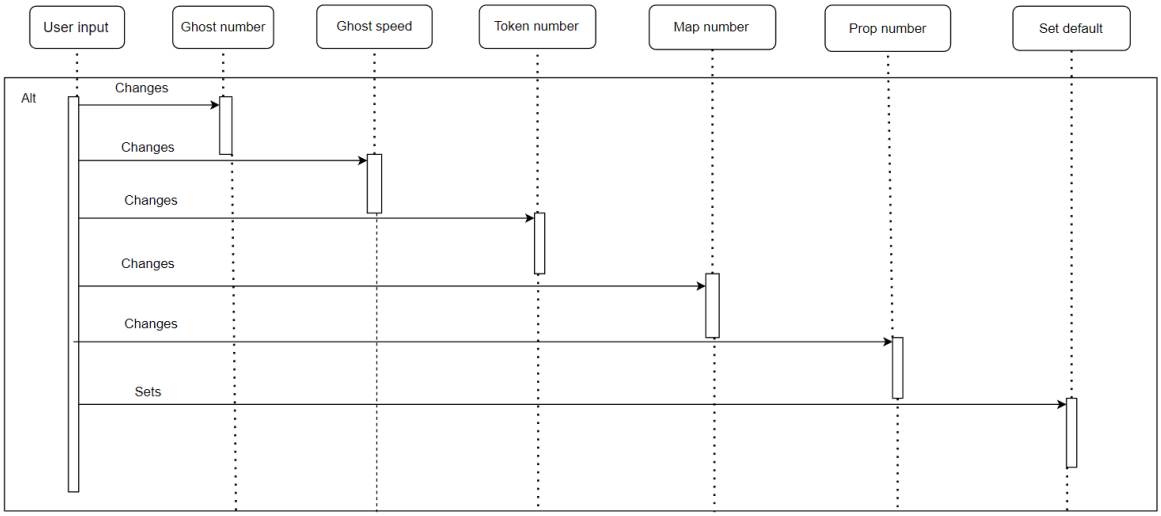

Figure 29: Sequence of Background Setting

1.6.2.4 UML Activity Diagram

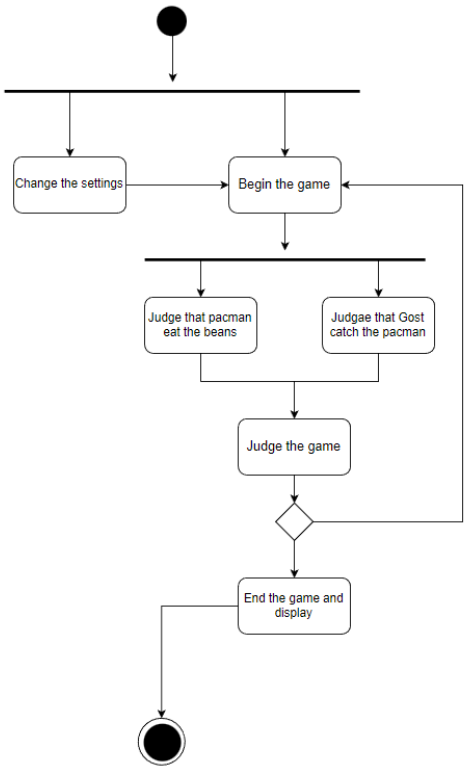

Figure 30: Activity Diagram of Background Setting

1.6.3 Functionality

This component handles the process and architecture of Background Setting system. The user gives the input and decide change which attribute. Additionally, the user can erase the previous settings to restore them to the default settings.

1.6.4 Procedures and Functions

Function	Parameter	Description
Game_start	User_input	Game will start according to the user input
Game_end	End_or_Continue	Game will end because of crash or user input
Compare_highscore	Game_result	Compare the game score with the historical high scores
Game_store	Game_result	This game is a high score and system will record it into the highest database
Display_result	Game_result	The game result will be display to the user
Select_low	None	Select a record which is low for updating record newly created
Insert_data	Game_result	Insert the data to the high score database
Update_record	Low_result, Game_result	Change the newly insert high record with the low record selected

Figure 31: Functionality Diagram of Background Setting

2 User Interface Design

2.1 Home Page

2.1.1 Description of the view

This page is both the Landing page and Home Page. The user can click on the buttons and view any information they want to find.

2.1.2 Screen of images

Figure 32: Homepage

2.1.3 Objects and Actions

There are three main functions: Click the "play" button, and you will be redirected to the registration interface, where you need to enter your nickname and then enter the first game level.; Click the "setting" button, and you will be redirected to the setting interface; Click the "high score" button to view the list of highest scores.

2.2 Enter Name

2.2.1 Description of the view

This page consists of a blank text box and a "play" button for the user to enter their nickname.

2.2.2 Screen of images

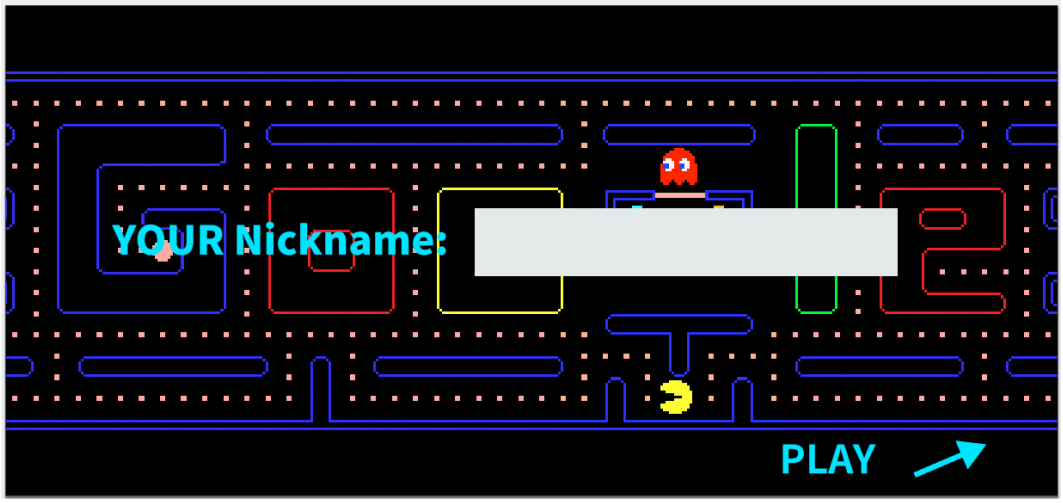

Figure 33: Enter Name

2.2.3 Objects and Actions

There is a blank text box on this page where you can enter the name you want to be on the scoreboard. After the input is complete, you can click the "play" button to enter the game.

2.3 Scoreboard

2.3.1 Description of the view

This page is used to display the player's highest scores in the game.

2.3.2 Screen of images

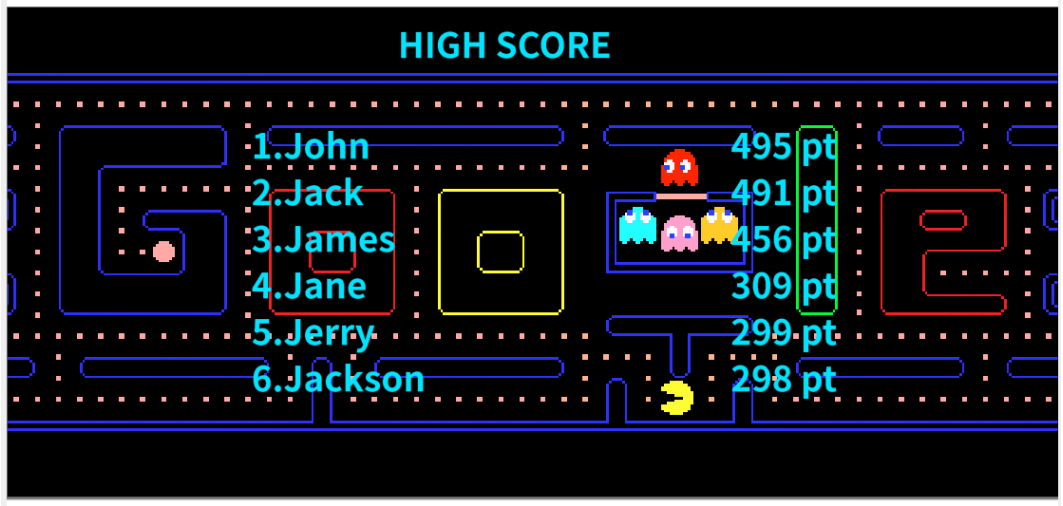

Figure 34: Scoreboard

2.3.3 Objects and Actions

There is a blank text box on this page where you can enter the name you want to be on the scoreboard. After completing the input, you can click the "play" button to enter the game.

2.4 Setting

2.4.1 Description of the view

This page is used to display the settings options that players can make.

2.4.2 Screen of images

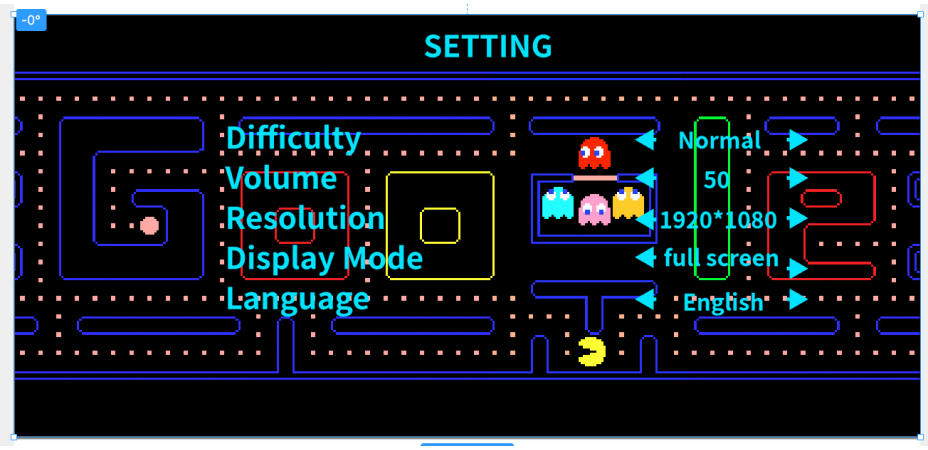

Figure 35: Settings

2.4.3 Objects and Actions

We provide users with five options that can be set. Each option is adjusted with the left or right button.

2.5 Log-in

2.5.1 Description of the view

This page is used to confirm user identity for login.

2.5.2 Screen of images

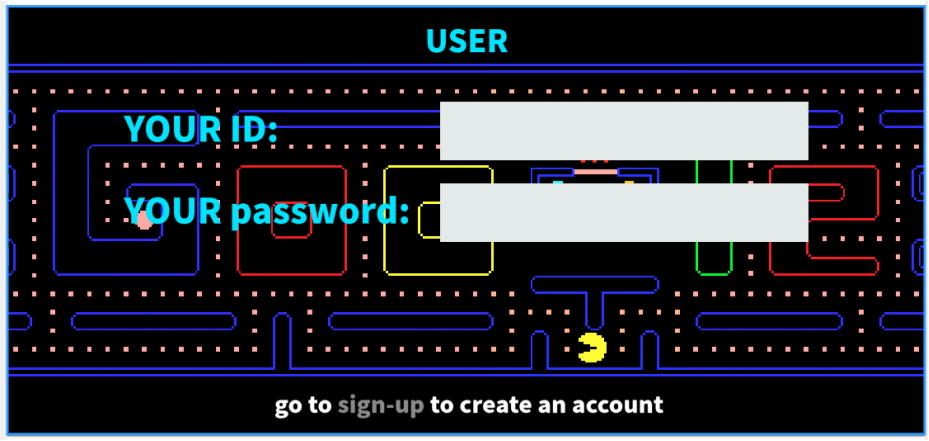

Figure 36: Log-in

2.5.3 Objects and Actions

Users need to enter their username and password in two text boxes. Our system will compare it to the database to complete the login process. If the user does not have an account, he can click the jump button below to sign up for an account.

2.6 Sign-up

2.6.1 Description of the view

This page is used for user registration accounts.

2.6.2 Screen of images

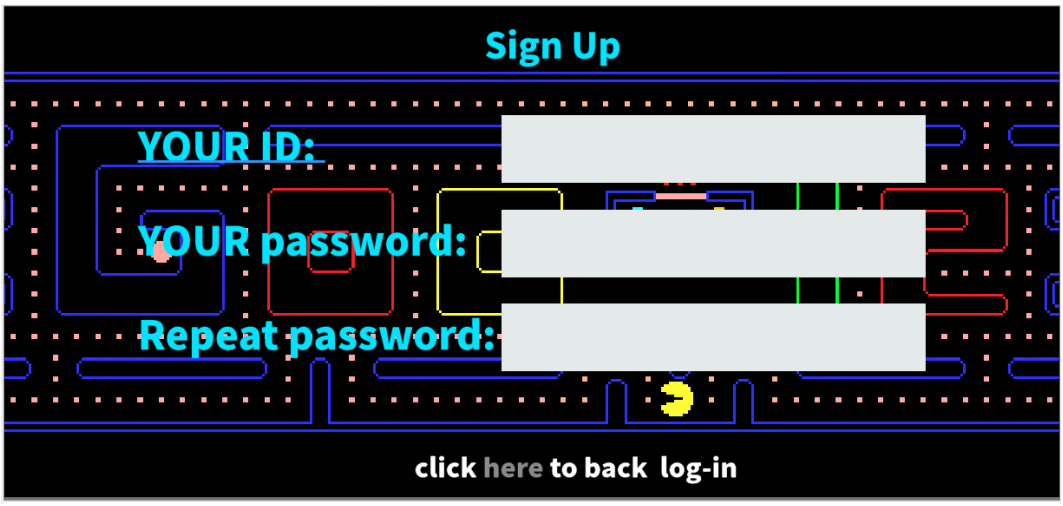

Figure 37: Sign-up

2.6.3 Objects and Actions

Users can create an account by entering information in three text boxes. We will store account information in a database. Users can then click the gray button to jump back to the login interface.

2.7 In game play

2.7.1 Initialize Maze

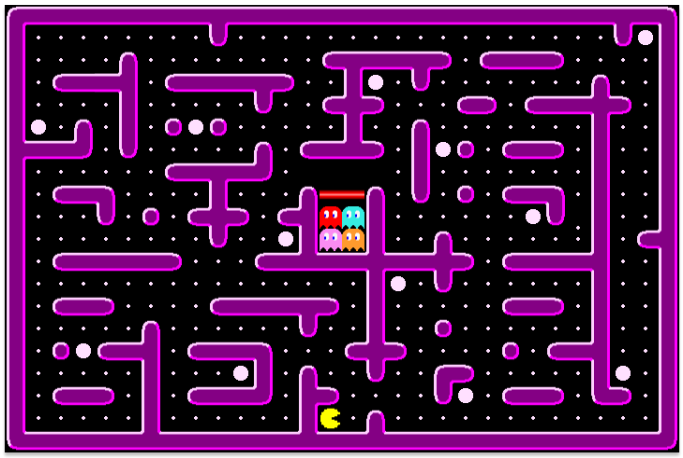

Figure 38: Initialized maze

After the player starts the game, the game will initialize a maze. Here is an example of a maze.The player will control Pac-Man, eating beans while avoiding ghosts through the maze. The Player can use the arrow keys to control the direction of Pac-Man's movement, up, down, left, or right.The goal of the game is to get all the beans and accumulate more scores in the process!

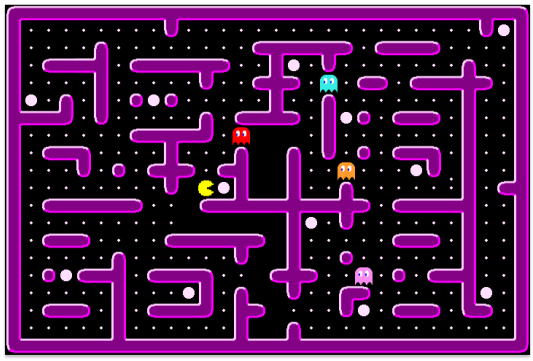

Figure 39: Before eating power bean

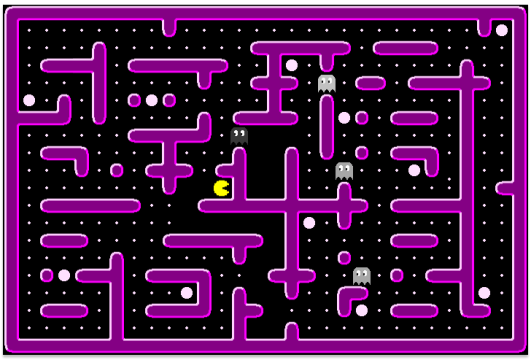

Figure 40: After eating power bean

In the process of playing, Pac-Man sometimes encounters some special beans, the most common of which is the "huge and powerful beans.". If Pac-Man eats it, the ghosts will temporarily become weak (appearing gray and trying to escape from Pac-Man), and weak ghosts can be eaten by Pac-Man and contribute some points.

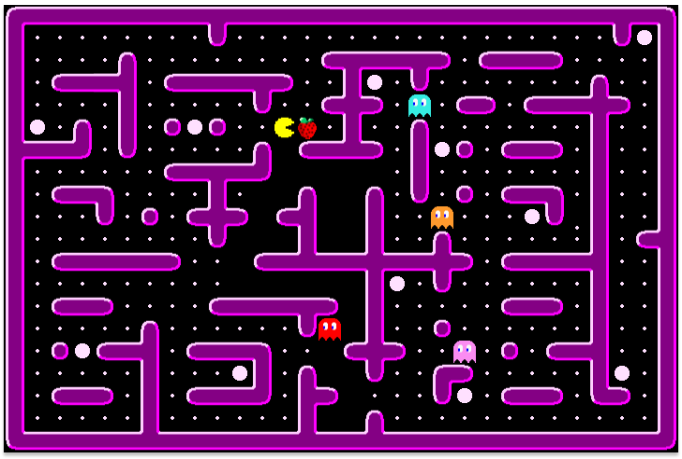

Figure 41: Eat other food

However, power beans are not the only special food in the game. There are many other foods there. If Pac-Man eats them, he will gain other effects. Such as obtaining additional scores, and so on.

2.7.2 Game ending processing

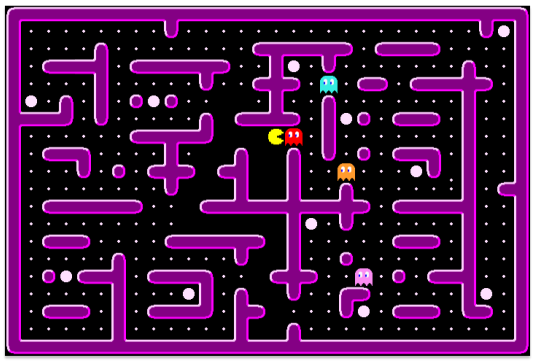

Figure 42: Before death

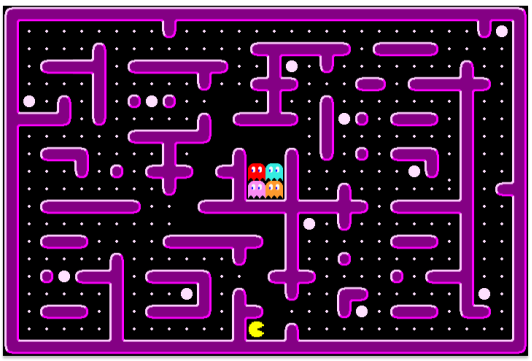

Figure 43: After death

If Pac-Man is caught by a normal ghost, he will lose a life! If it has any remaining opportunities, the game will restart from its current state and only initialize the positions of the Pac-Man and ghosts.

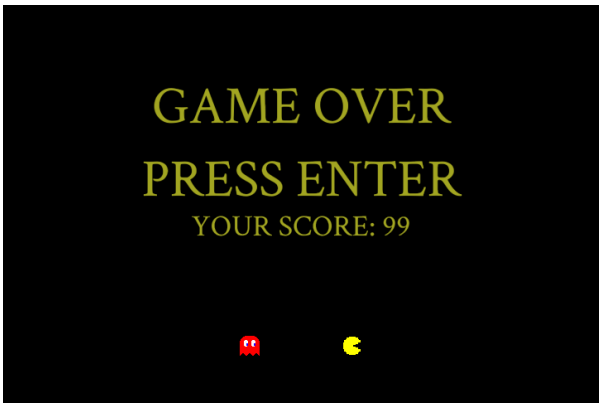

Figure 44: Gameover

However, if Pac-Man doesn't have a chance, it will really die, which means the game is over. At this point, you will receive an interface such as the following, which will print out the total score you have obtained for this game.

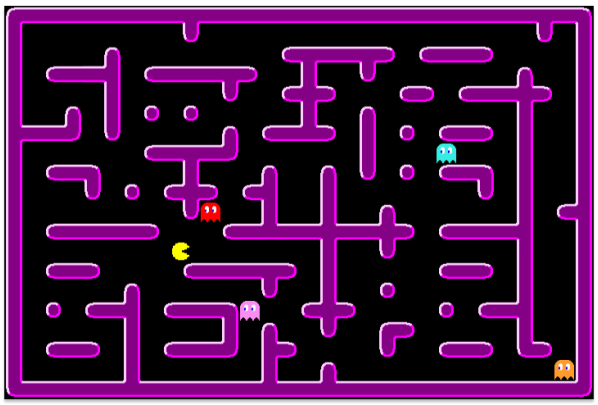

Figure 45: Win a maze

On the other hand, if Pac-Man eats all the beans in the maze. Congratulations, player has won this maze. According to the game process, if the player can enter the next level, it will be sent to the next maze and initialized, retaining the score of this game. In that case, the next step is to repeat the previous process.

If the player wins all levels, it means that the player has won the entire game process, and the game will also end, but this is done in a winner's way. The player will receive the same interface as shown below, and the same interface will print out the scores obtained by the player for the entire game.

Figure 46: Win the whole game