Table of Contents

Contents

1.0 Project Planning and Documentation

1.1 Time Schedule

This table should reflect who did what, how long you expected sections to take and the actual hours it took to perform the tasks.

Task		Plan				Actual		
#	Task Name	Student	Planned Time	Cumulative Time	Finished Date	Time	Cumulative Time	Finished Date
1	Project plan & Review Meetings	Leah Denny Kathryn Hitchener Andrew Hamenko Koby Lestrelle	2 hrs 2 hrs 2hrs 2hrs	8hrs	1/08/2021	4hrs 4 hrs 4 hrs 4hrs	16hrs	8/08/2021
2	Identify Functional Requirement	Leah Denny	4 hrs	4 hrs	8/08/2021	5hrs	5hrs	27/08/2021
3	Identify Non-Functional Requirements	Leah Denny	2 hrs	2 hrs	8/08/2021	2hrs	2hrs	27/08/2021
4	Create Use Case Diagram	Andrew Hamenko Leah Denny	2 hrs 1hr	3hrs	23/08/2021	1hrs 1hr	2hrs	31/08/2021
5	1 Full Use Case Description	Leah Denny	2 hrs	2 hrs	30/08/2021	2.5hrs	2.5hrs	30/08/2021
6	Requirement Use Case Traceability Matrix	Leah Denny	1 hr	1 hr	30/08/2021	1.5hr	1.5hr	31/08/2021
7	Class Diagram	Kathryn Hitchener	4 hrs	4 hrs	23/08/2021	6hrs	6hrs	1/09/2021
8	Sequence Diagram	Kathryn Hitchener	2 hrs	2 hrs	23/08/2021	4 hrs	4 hrs	4/09/2021

9	Activity Diagram	Kathryn Hitchener	2 hrs	2 hrs	23/08/2021	2 hrs	2hrs	23/08/2021
10	Component and Controller View	Kathryn Hitchener	2 hrs	2 hrs	30/08/2021	3hrs	3hrs	1/09/2021
11	Implementation style view	Koby Lestrelle	2 hrs	2 hrs	30/08/2021	2hrs	2hrs	1/09/2021
12	Deployment style view	Koby Lestrelle	2 hrs	2 hrs	30/08/2021	2hrs	2hrs	1/09/2021
13	Coding: Maze & Environment	Andrew Hamenko	4 hrs	4 hrs	23/08/2021	7hrs	7...+hrs	Ongoing
14	Coding: Main Menu	Koby Lestrelle	3 hrs	3hrs	23/08/2021	3hrs	3hrs	26/8/2021
15	Coding: PacMan (Part 1)	Andrew Hamenko	2 hrs	2 hrs	30/08/2021	2hrs	2...+hrs	Ongoing
16	Coding Ghosts (Part 1)	Andrew Hamenko	2 hrs	2 hrs	30/08/2021	2hrs	2...+hrs	Ongoing
17	Github Version Management	Koby Lestrelle	1 hr	1 hr	08/08/2021	2hrs	2...+ hrs	Ongoing

1.2 Total working hours

Student Name (#ID)	**Plan (hours)**	**Actual (hours)**
Andrew Hamenko (s5220455)	12	16
Kathryn Hitchener (s5197403)	12	19
Koby Lestrelle (s5226483)	10	13
Leah Denny (s5131799)	12	16
Total working hours	46	64
Average working hours per person	11.5	16

1.3 Effort and contribution table

Student	Effort Level* (Rating from 0 – 5, the information is filled by the group)	Contribution Level* (Rating from 0 – 5, the information is filled by the group)	Justification If a student received level rating of 3 or less, your group need to give explanation for the low level rating
Andrew Hamenko (s5220455)	5	5	
Kathryn Hitchener (s5197403)	5	5	
Koby Lestrelle (s5226483)	5	5	
Leah Denny (s5131799)	5	5	
Total	20	20	

*Level ratings, 5 = excellent, 4 = good, 3 = reasonable, 2 = poor, 1 = unacceptable, 0 = none

1.4 Version Control System

Please see below for a screenshot of the version control system, github that is being used to control the code versions throughout this project.

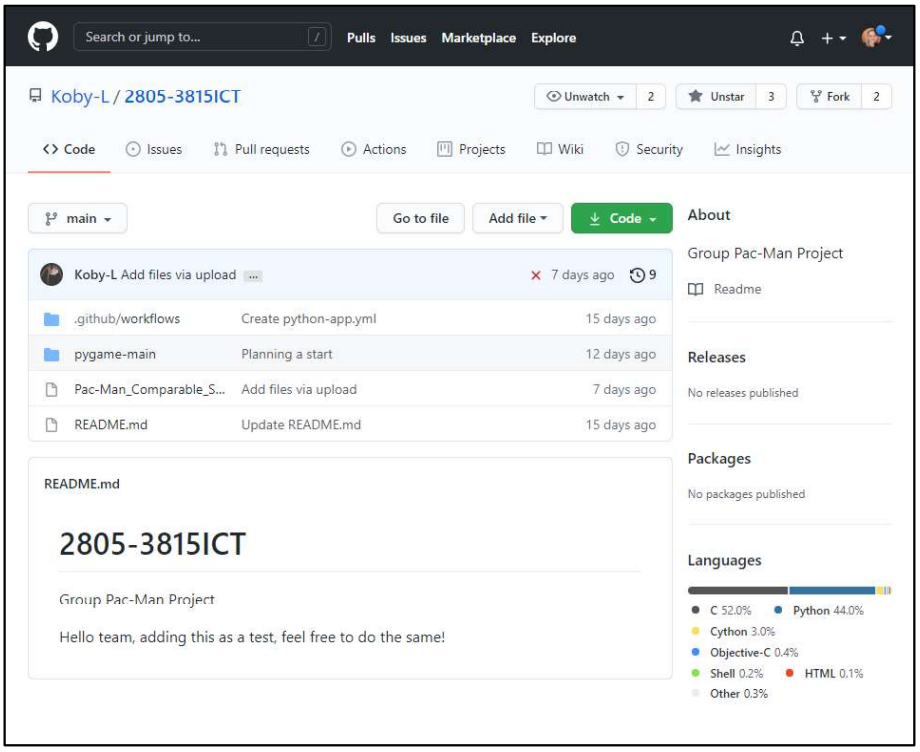


Figure 1 - Demonstration of Github Version Control System

2.0 Requirements Analysis

2.1 Functional requirements

Identifier	Priority	Requirement
F-REQ1	1	The system should allow for the game to be executed on at least two different platforms.
F-REQ2	1	The system should display a start-up page when the game is started.
F-REQ3	1	The start-up page should display the title and logo of Pac-Man.
F-REQ4	1	The start-up page should display the current year and course code for the course this assessment is completed for.
F-REQ5	1	The start-up page should display the list of all students in the group, in alphabetical order.
F-REQ6	1	The start-up page should display an exit button.
F-REQ7	2	The exit button displayed on the start-up page should be fully functional, that when pressed/clicked on it the user exits the program with result 0 (successful termination).
F-REQ8	1	The start-up page should display a configure button.
F-REQ9	3	The configure button displayed on the start-up page should be fully functional, that when pressed/clicked on it takes the user to another page where they can choose the maze generated; there should be two options: fixed maze, randomly generated maze.
F-REQ10	1	The start-up page should display a play button.
F-REQ11	1	The play button displayed on the start-up page should be fully functional, that when pressed/clicked on it changes the screen from the start-up page to the play screen.
F-REQ12	1	The play screen should display a maze, Pac-Man sprite (user), dots, power pellets, fruits, and four Monsters in a home-base, and the user should start a new game with 3 lives, which should be displayed to the user.
F-REQ13	1	The Pac-Man sprite on the play screen should be able to move in the maze also displayed on the play screen.
F-REQ14	2	The monster sprites on the play screen should move in the maze without any manipulation from the user.
F-REQ15	2	When a monster sprite and the Pac-Man sprite intercept without the Pac-Man sprite being in an ‘powered-up state’, the Pac-Man sprite should

		disappear and reappear in its initial starting spot on the maze, total lives for the user should be reduced by 1 and this should be displayed to the user, and the monster sprites should be reset in the home-base. The dots and power pellets should not recover.
F-REQ16	2	When a monster sprite and the Pac-Man sprite intercept with the Pac-Man sprite being in an 'powered-up state', the monster sprite should disappear and reappear at the Monster's home-base while still being affected by Pac-Man's 'powered-up state', with the user's points being increased by a certain amount with each monster eaten in a row (points earned in order: 1,200, 2,400, 3,800, 5,600) and this should be displayed to the user.
F-REQ17	2	When the Pac-Man sprite intercepts with a power pellet, the power pellet disappears, the user's points are increased by 50 and this should be displayed to the user, and the Pac-Man sprite is set to a 'powered-up state' where monster sprites change their direction to the opposite of their current direction, per monster.
F-REQ18	2	When the Pac-Man sprite intercepts with a dot, the dot disappears, and the user's points are increased by 10 and this should be displayed to the user.
F-REQ19	4	When the Pac-Man sprite intercepts with a fruit, the fruit disappears, and the user's points are increased in order of how many fruits they've eaten (points in order: 100, 300, 500, 700, 1,000).
F-REQ20	2	When the "Main Menu" button is pressed, the start-up page should be displayed.
F-REQ21	3	When the "Play Again" button is pressed, the game should reset and be newly displayed to the user.
F-REQ22	2	When the Pac-Man sprite intercepts with the last dot on the maze, the screen should display a pop-up box containing "You Win!" along with two buttons (one for "Play Again" and one for "Main Menu").
F-REQ23	1	The system should record the amount of dots on the maze and keep the record updated in the back-end.
F-REQ24	1	When the user has lost all of their lives (lives=0) the screen should display a pop-up box containing "You've lost." Along with two buttons (one for "Play Again" and one for "Main Menu").
F-REQ25	4	When the start-up page is being viewed, there should be suitable music playing.

F-REQ26	4	When the play screen is being viewed, there should be suitable music playing.
F-REQ27	1	The system should only have one level of the game.
F-REQ28	5	There should be appropriate sound effects played during game-play.
F-REQ29	3	There should be a “Main Menu” button displayed which, when pressed, immediately ends the current game (resets game) and takes the user to the start-up page.
F-REQ30	2	The maze should be able to be randomized if the user chooses a random maze in the configuration page.
F-REQ31	3	When the Pac-Man sprite is moved to collide with a wall, the Pac-Man sprite changes direction and moves in the opposite direction than it was going.
F-REQ32	1	When the user presses and/or holds down the left arrow-key, the Pac-Man sprite will move to the left direction, unless faced with a wall in which the Pac-Man sprite will continue on its current course of direction.
F-REQ33	1	When the user presses and/or holds down the right arrow-key, the Pac-Man sprite will move to the right direction, unless faced with a wall in which the Pac-Man sprite will continue on its current course of direction.
F-REQ34	1	When the user presses and/or holds down the down arrow-key, the Pac-Man sprite will move to the down direction, unless faced with a wall in which the Pac-Man sprite will continue on its current course of direction.
F-REQ35	1	When the user presses and/or holds down the up arrow-key, the Pac-Man sprite will move to the up direction, unless faced with a wall in which the Pac-Man sprite will continue on its current course of direction.
F-REQ36	2	Two Monster sprites (Monster A and Monster B) will move randomly around the maze.
F-REQ37	2	Two Monster sprites (Monster C and Monster D) will move around the maze intelligently, following the user’s (Pac-Man's) movement route.
F-REQ38	2	The Pac-Man’s ‘powered-up state’ should last for 20 seconds, whereafter the state is returned to normal.

Table 1: Functional Requirements

2.2 Non-functional requirements

Identifier	Priority	Requirement
NF-REQ1	5	*Usability* – The system should have a “How to Play” page as documentation for the player to easily use the system.
NF-REQ2	5	*Usability* – The system should be aesthetically pleasing, and consistent with the original 1980’s Pac-Man game.
NF-REQ3	5	*Supportability* – The system should not require a large amount of computer memory in order to play locally.
NF-REQ4	5	*Performance* – The system should require minimal computing speed in order to play locally.
NF-REQ5	5	*Functionality* – The system should not require any authorization in order to run in its entirety locally.
NF-REQ6	5	*Reliability* – The system should be able to be run/executed on a portable device, such as a laptop with Windows 10 installed.
NF-REQ7	5	*Functionality* – The system should be able to have responsive sizing of the game.

Table 2: Non-Functional Requirements

2.3 Use case diagram

https://lucid.app/lucidchart/051b645d-3520-4396-a1fd-90f8a756716f/edit?beaconFlowId=594E53411BF427C8&page=0_0

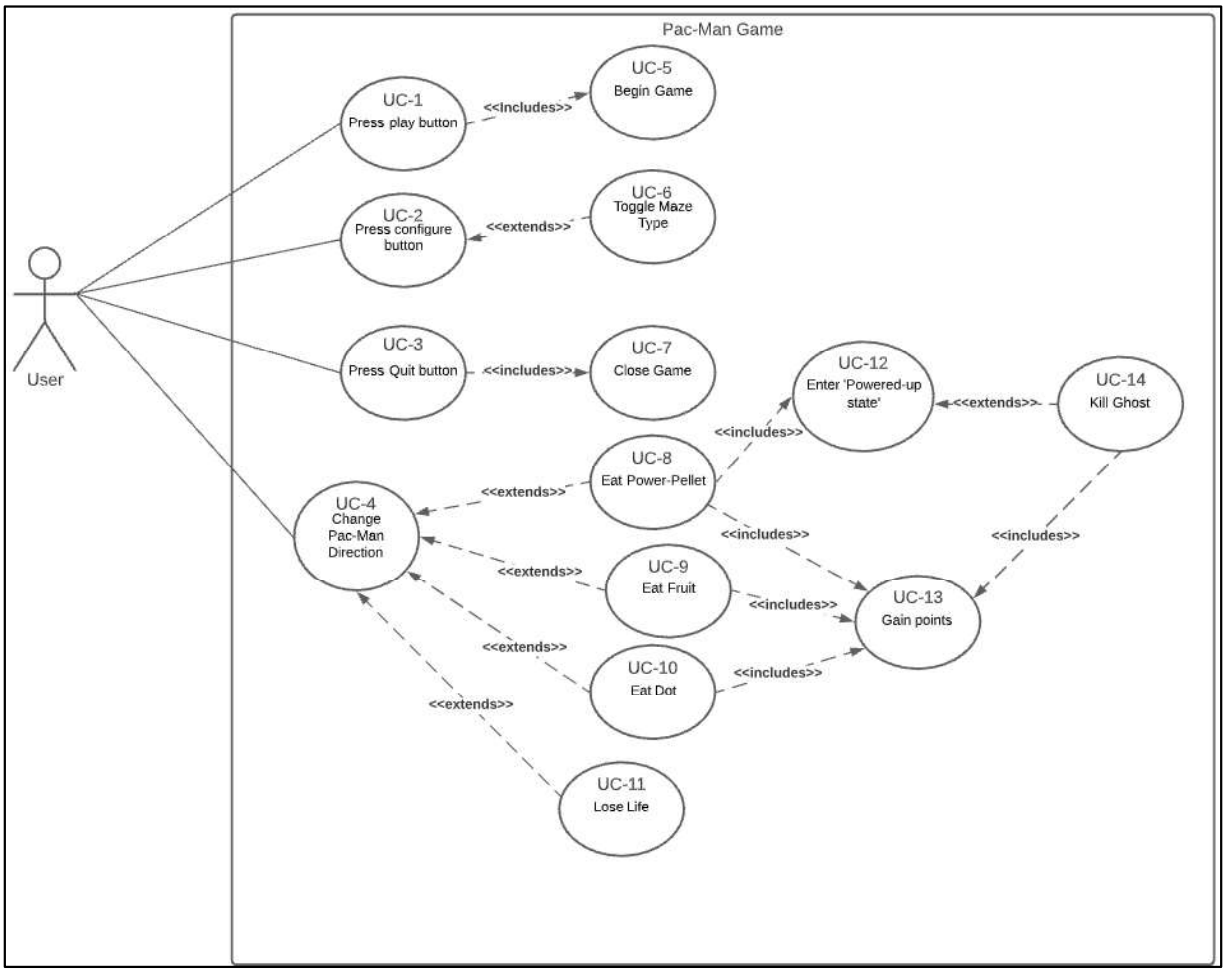


Figure 2: Use Case Diagram

2.4 Full use case description

<table>
<tr><th colspan="3">Fully Developed Use Case</th></tr>
<tr><td>Use Case Name:</td><td colspan="2">Eat Power-Pellet</td></tr>
<tr><td>Scenario/s:</td><td colspan="2">Pac-Man sprite intercepts with a power pellet while in normal state
Pac-Man sprite intercepts with a power pellet while in powered up state</td></tr>
<tr><td>Triggering Event:</td><td colspan="2">The user uses the arrow keys to move the Pac-Man sprite which intercepts with a power pellet in the maze.</td></tr>
<tr><td>Brief Description:</td><td colspan="2">User moves the Pac-Man sprite around the maze to a location where a power pellet is, thus resulting in the power pellet being ‘eaten’ by the Pac-Man sprite. This results in Pac-Man changing to a ‘powered-up’ state and the ghosts changing to a ‘ghost-mode’ state.</td></tr>
<tr><td>Actors:</td><td colspan="2">User, System</td></tr>
<tr><td>Related user Cases:</td><td colspan="2">Change Pac-Man Direction, Eat Fruit, Eat Dot, Kill Ghost</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Players, Developer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">- Pac-Man sprite must intercept with power pellet
- There must be at least one power pellet present on the maze (>= 1 power pellet on maze)
- User must be actively playing the game and making Pac-Man move around the maze
- Pac-Man must have more than zero lives (> 0 lives)</td></tr>
<tr><td>Postconditions:</td><td colspan="2">- Pac-Man must have more than zero lives (> 0 lives)
- User will continue actively playing the game and making Pac-Man move around the maze
- Pac-Man will return to a ‘normal’ state
- Ghosts will return to an ‘attacking’ state
- Music will return to the ‘main’ sound</td></tr>
<tr><td rowspan="4">Flow of activities:</td><td>User</td><td>System (Pac-Man Game)</td></tr>
<tr><td>1. Press play button</td><td>1.1 Set-up game environment
1.2 Begin game</td></tr>
<tr><td>2. Presses the arrow keys to direct Pac-Man</td><td>2.1 Pac-Man direction changes on screen and moves according to the key presses</td></tr>
<tr><td>3. User intersects Pac-Man with a Power-Pellet</td><td>3.1 Power-Pellet disappears from screen
3.2 Score counter is increased by 50 and displayed to the user
3.3 Timer starts counting down 20 seconds which is not displayed to the user
3.3 Music changes to ‘Powered-up’ music
3.4 Ghosts change to ‘Ghost-Mode’ state
3.5 Pac-Man changes to ‘Powered-up’ State</td></tr>
</table>

	4. User intersects Pac-Man with a 'Ghost'	4.1 Ghost returns to home base 4.2 Score is increased by (1,200, 2,400, 3,800, 5,600) and displayed to the user
		5. Timer runs out
Exception Conditions:	1. Pac-Man sprite (user) eats a power pellet whilst already in a 'powered-up state'. The 'powered-up states' do not merge their accumulative time together, they instead restart the current time period for one 'powered-up state'. 2. Pac-Man sprite (user) eats last dot while powered-up state, game ends and powered-up state ends (game doesn't continue until powered-up state finishes to end game). 3. Pac-Man sprite (user) kills a ghost which returns to a normal state and then attacks Pac-Man causing a loss of life. This will reset the game as in the 'lost-life' use case and stop the 'Powered-Up' state. 4. Users presses keys that don't correspond with the 4 arrow keys 5. User quits the game.	

Table 3: Full Use Case Description

2.5 Requirement - use case traceability matrix

Req.	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14
F-REQ1	X													
F-REQ2					X									
F-REQ3					X									
F-REQ4					X									
F-REQ5					X									
F-REQ6			X											
F-REQ7			X				X							
F-REQ8		X												
F-REQ9		X	X			X								
F-REQ10	X													
F-REQ11	X				X									
F-REQ12					X	X								
F-REQ13				X										
F-REQ14					X									
F-REQ15											X			
F-REQ16													X	X
F-REQ17								X				X	X	
F-REQ18										X			X	
F-REQ19									X				X	
F-REQ20							X							
F-REQ21	X				X									
F-REQ22									X				X	
F-REQ23									X				X	
F-REQ24							X				X			
F-REQ25					X									
F-REQ26					X									
F-REQ27					X									
F-REQ28					X									
F-REQ29			X				X							
F-REQ30		X				X								
F-REQ31				X										
F-REQ32				X										

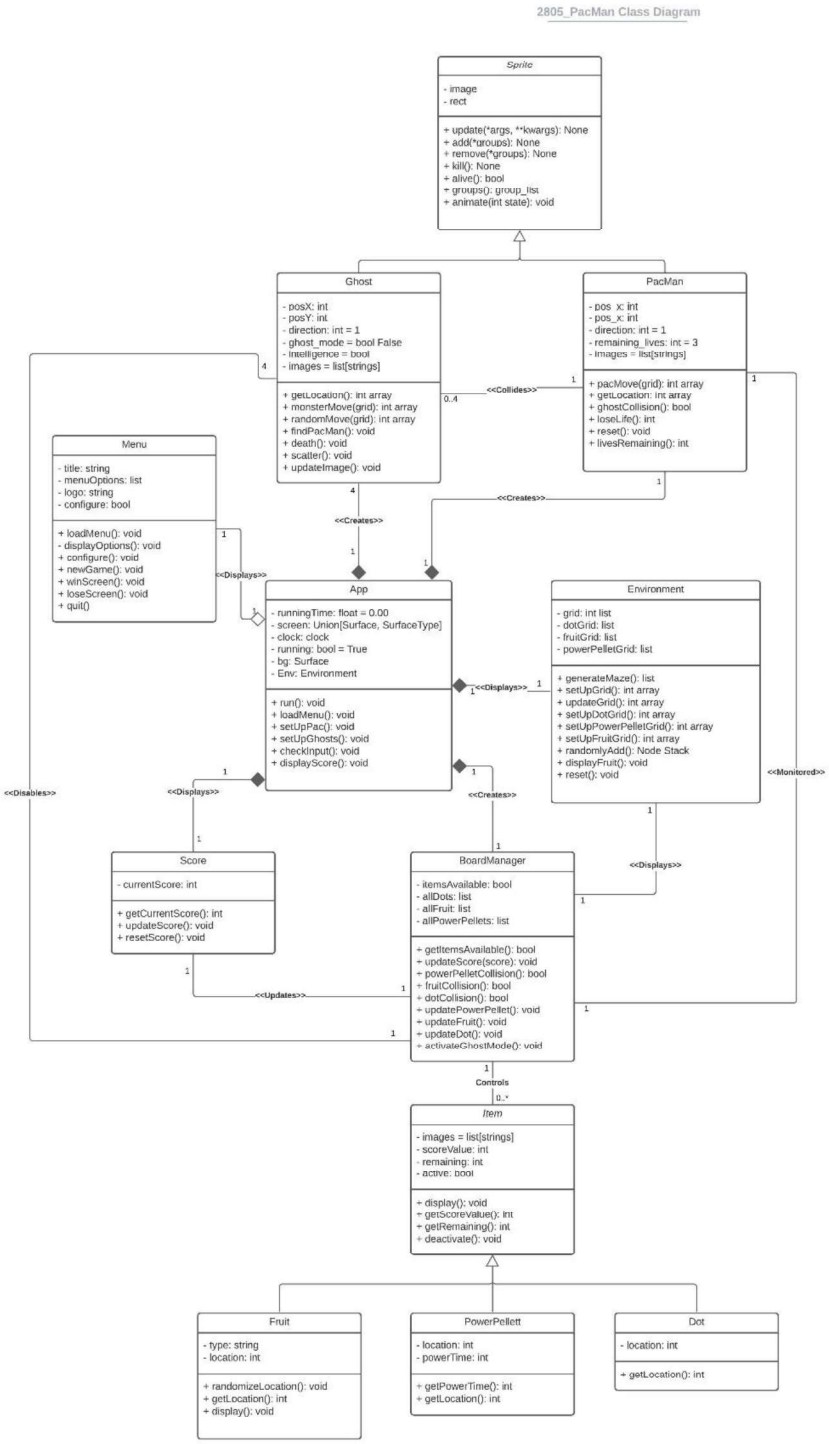
F-REQ33					X									
F-REQ34					X									
F-REQ35					X									
F-REQ36						X								
F-REQ37						X								
F-REQ38									X				X	

Table 4: Use Case Traceability Matrix

3.0 Design and software architecture

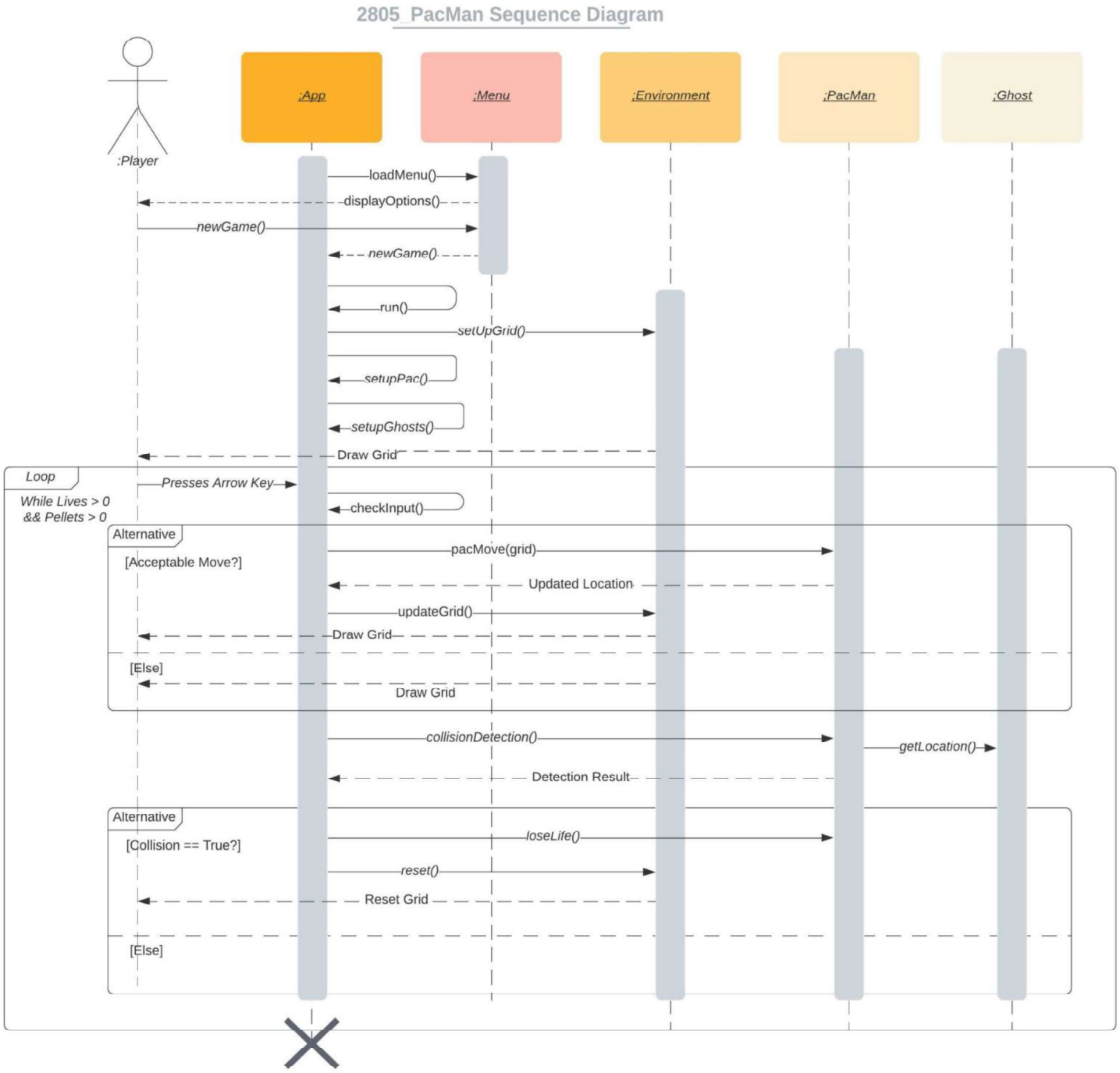
3.1 Class diagram

Full Screen View: https://lucid.app/documents/view/8735ec0d-0882-49d3-8373-833628c0aef6



3.2 Sequence diagram

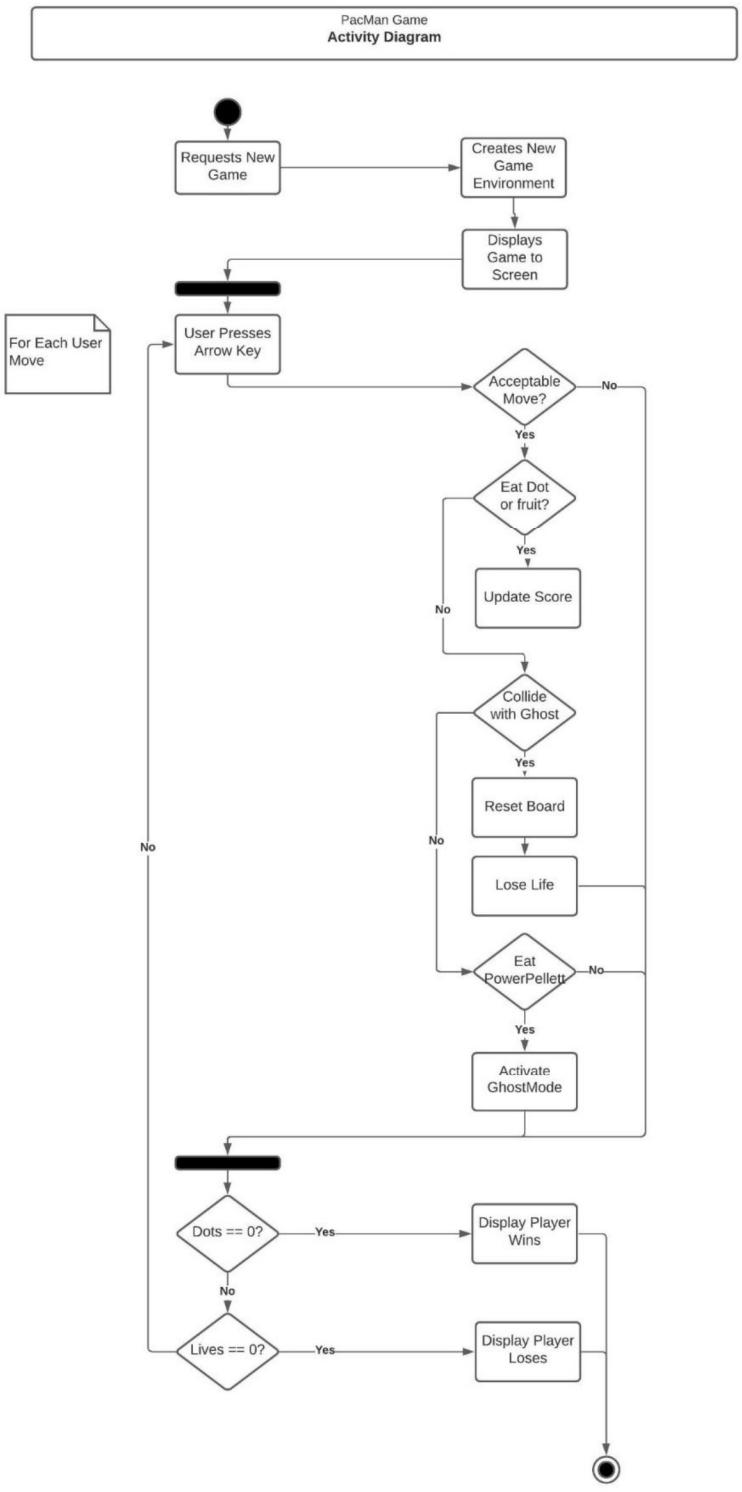
Full Screen View:
https://lucid.app/documents/view/6ccbd244-d643-46bd-b9c9-fb94aef50f64



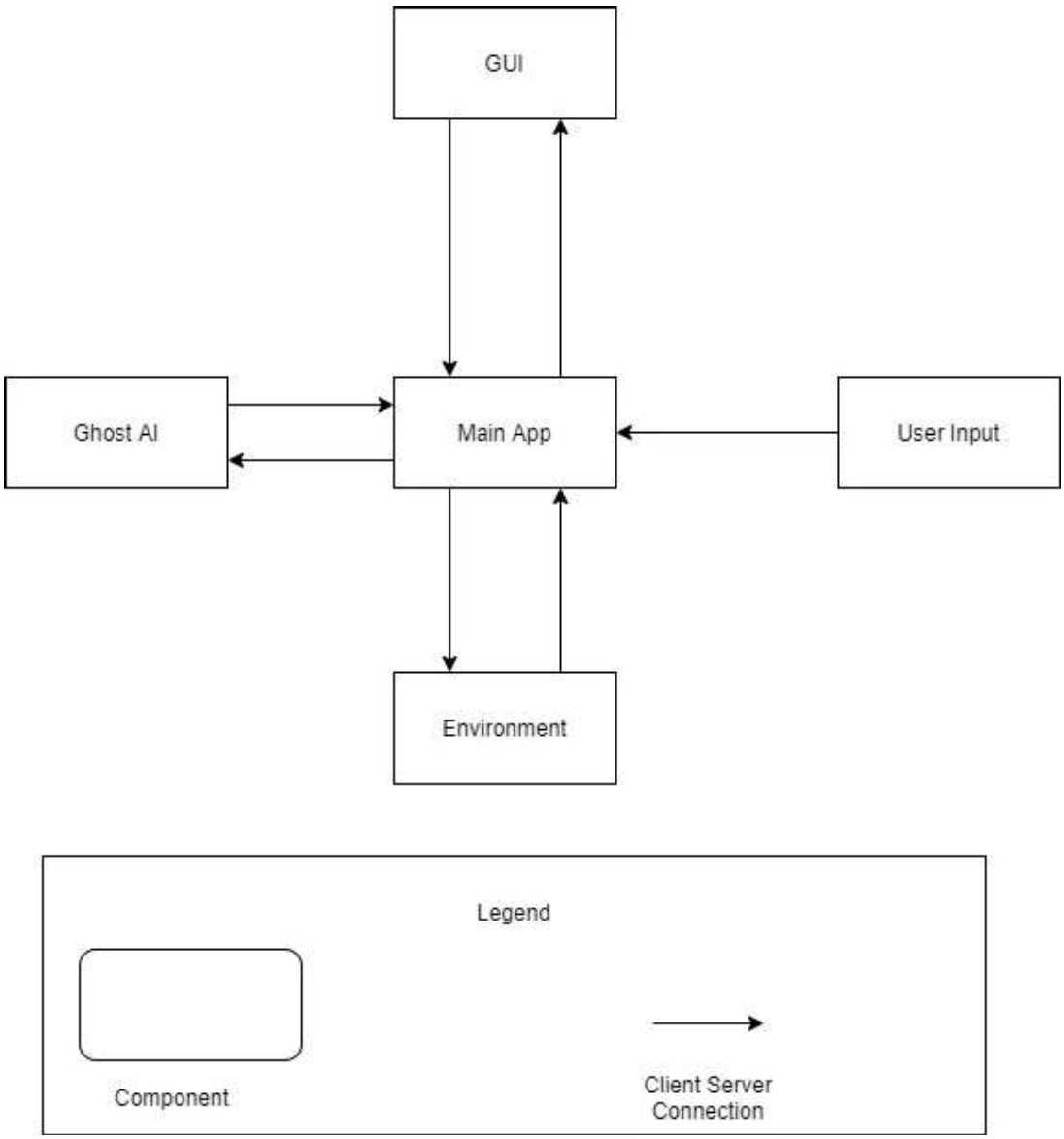
3.3 Activity diagram

Full Screen View:

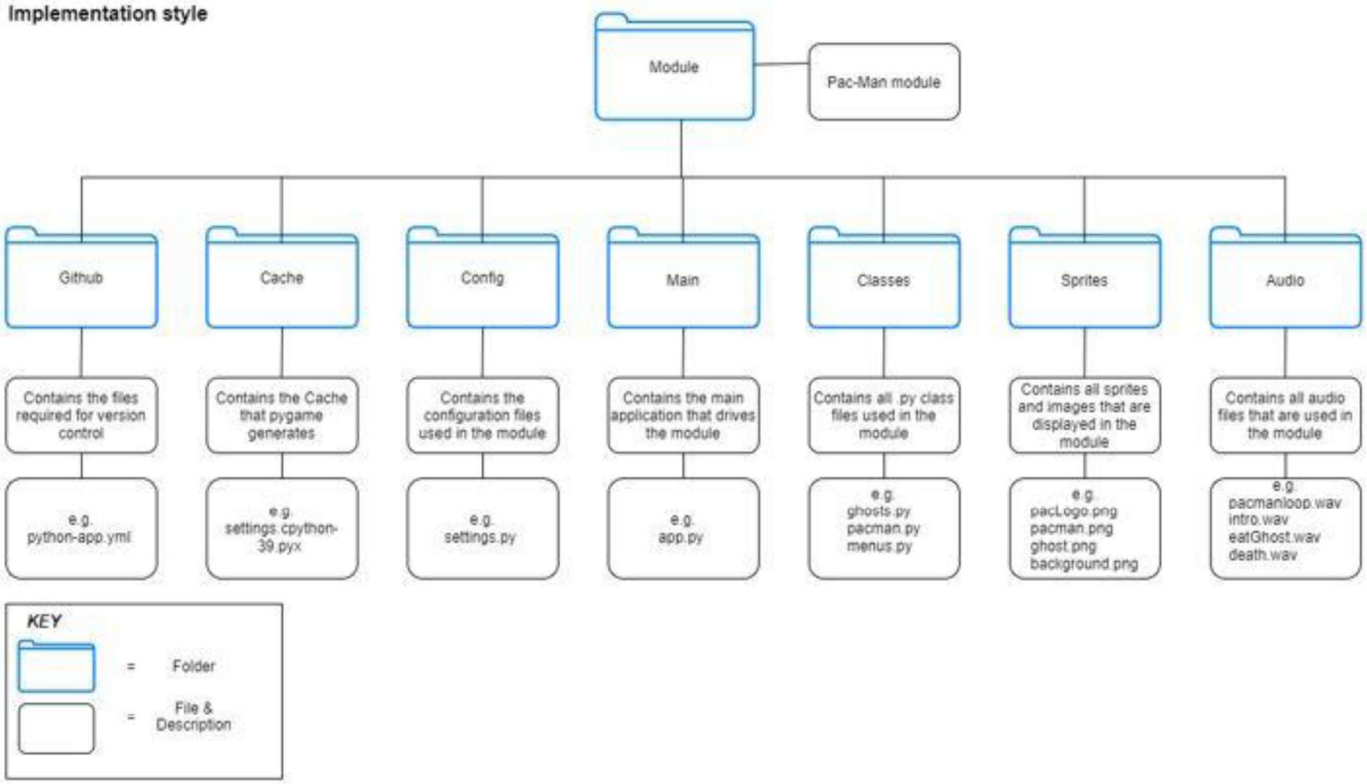
https://lucid.app/documents/view/2c33f040-f4dc-4ea8-a364-c2be3c6e0e93



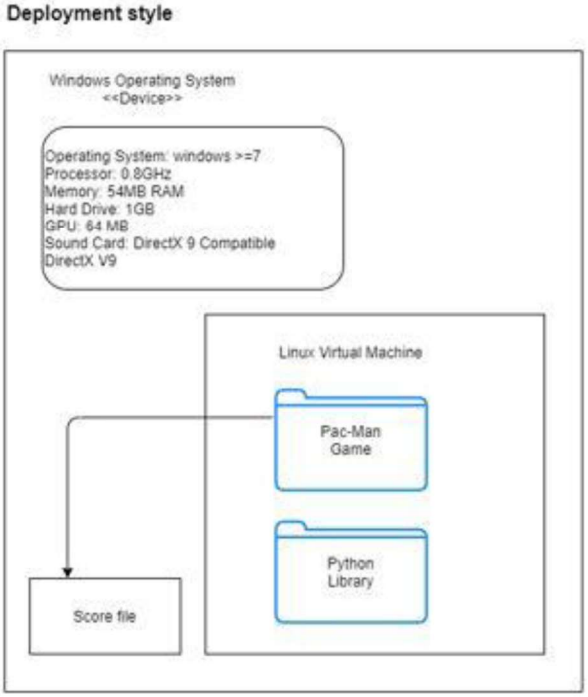
3.4 C & C View


GUI
Ghost AI
Main App
User Input
Environment
Legend
Component
Client Server
Connection

3.5 Implementation style view



3.6 Deployment style view



4.0 Video link

Link to the demonstration of the working Pac-man implementation. Part 1:

https://drive.google.com/file/d/1km7PjP_Bpd1BTtK1ilb2YeAa_S6NeGpd/view?usp=sharing

Figure 3 Menu Screenshot

4.1 Video Defence:

The video above demonstrates the program operating on a Linux-based environment (Laptop on left) and a windows-based environment (Laptop on right). The procedure performed on the program is the same for both environments and that is:

- Open Program to Main menu (with student and course names displayed).
- Demonstrate 'Quit' functionality
- Display configure button (Limited functionality)
- Move Pac-man through maze using keyboard
- Demonstrate Ghosts moving through the maze on their own

Note that the ghosts appear to get stuck at certain points in the maze. This occurs only due to their limited intelligence as no path finding algorithms have been developed for them yet, this will be implemented at a later milestone. Similarly, it can be seen in the video that when a ghost touches Pac-Man, nothing happens. This is because the 'Death' functionality will be implemented at a later milestone.

The image above displays the menu that is shown in the video. Since the limited time allowed for the video, please see the screenshot to demonstrate that all required elements are present.