Assignment
Stage One Submission
2805ICT/3815ICT/7805ICT

Student name: Yasin Çakar Student ID: s2921450 Enrolled Course Code: 3815ICT

Student name: Mohammad Mari Student ID: s5185052 Enrolled Course Code: 7805ICT

Student name: David Todorovic Student ID: s5167246 Enrolled Course Code: 3805ICT

Table of 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	Planed Time	Cumulative Time	Finished Date	Time	Cumulative Time	Finished Date
1	Project plan	Mohammad Mari / Yasin Çaker	6 hours		Aug 8	8 hours		Aug 8
2	Identify Functional Requirement	Yasin Caker / David Todorovic	8 hours		Aug 14	12 hours		Aug 14
2	Draw diagrams	Mohammad Mari / Yasin Çaker	12 hours		Aug 20	24 hours		Aug 20
3	code analysis and language decision	Yasin Caker / David Todorovic / Mohammad Mari	1 hours		Aug 24	2 hours		Aug 24
4	code implementation	David Todorovic	16 hours		Aug 28	30 hours		Aug 28
5	software testing	Mohammad Mari/ Yasin Caker	2 hours		Aug 28	4 hours		Aug 28
6	code updates and refactor	David Todorovic	10 hours		Aug 31	16 hours		Aug 31
7	video preparation	Yasin Caker / David Todorovic / Mohammad Mari	2 hours		Sep 1	4 hours		Sep 2

1.2 Total working hours

Student Name (#ID)	Plan (hours)	Actual (hours)
David Todorovic	24	30
Yasin Çakar	27	28
Mohammad Mari	22	26
Total working hours	73	84
Average working hours per person	25 hours per person	28 hours per person

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
Mohammad Mari	5	5	
Yasin Çakar	5	5	
David Todorovic	5	5	
Total	15	15	

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

1.4 Version Control System

[Your group needs to use a version control system (VCS) to manage the source code development. Please use screenshot to demonstrate that a suitable VCS system has been applied in developing this project.]

```
Author: DavidTodoroviic <davidt965@gmail.com>
Date:   Wed Aug 25 18:03:13 2021 +1000

    Prototype v1

commit 73c210d3493ad3219162ec6ca73ffed0a0f008d3
Author: DavidTodoroviic <davidt965@gmail.com>
Date:   Sun Aug 22 18:22:31 2021 +1000

    Adding sample project

commit 5c72b2363c6e699a3c73e65008852ca4d2a1a7e4
Author: Mohammad Mari <mohammad.mari@griffithuni.edu.au>
Date:   Sun Aug 8 10:46:19 2021 +1000

    added blank template

commit 9dd82d7a58589c94d927a8ad20e168fa6253bbd2 (origin/template)
Author: Lev0071 <74816473+Lev0071@users.noreply.github.com>
Date:   Sun Aug 8 09:52:43 2021 +1000

    Initial commit
:...skipping...
commit 639416f121dede1361565bf679c833861429bffb (HEAD -> main, origin/main, orig
in/HEAD)
Author: DavidTodoroviic <davidt965@gmail.com>
Date:   Fri Aug 27 17:43:24 2021 +1000

    Pacman no sound

commit 333abc4a909f5afaaa6d1099ac41070aa0b04fbd
Author: DavidTodoroviic <davidt965@gmail.com>
Date:   Wed Aug 25 18:03:13 2021 +1000

    Prototype v1

commit 73c210d3493ad3219162ec6ca73ffed0a0f008d3
Author: DavidTodoroviic <davidt965@gmail.com>
Date:   Sun Aug 22 18:22:31 2021 +1000

    Adding sample project

commit 5c72b2363c6e699a3c73e65008852ca4d2a1a7e4
Author: Mohammad Mari <mohammad.mari@griffithuni.edu.au>
Date:   Sun Aug 8 10:46:19 2021 +1000

    added blank template

commit 9dd82d7a58589c94d927a8ad20e168fa6253bbd2 (origin/template)
Author: Lev0071 <74816473+Lev0071@users.noreply.github.com>
Date:   Sun Aug 8 09:52:43 2021 +1000

    Initial commit
~
south-10-20-57-20:SE_Stage1_Submission yasin$
```

2.0 Requirements Analysis

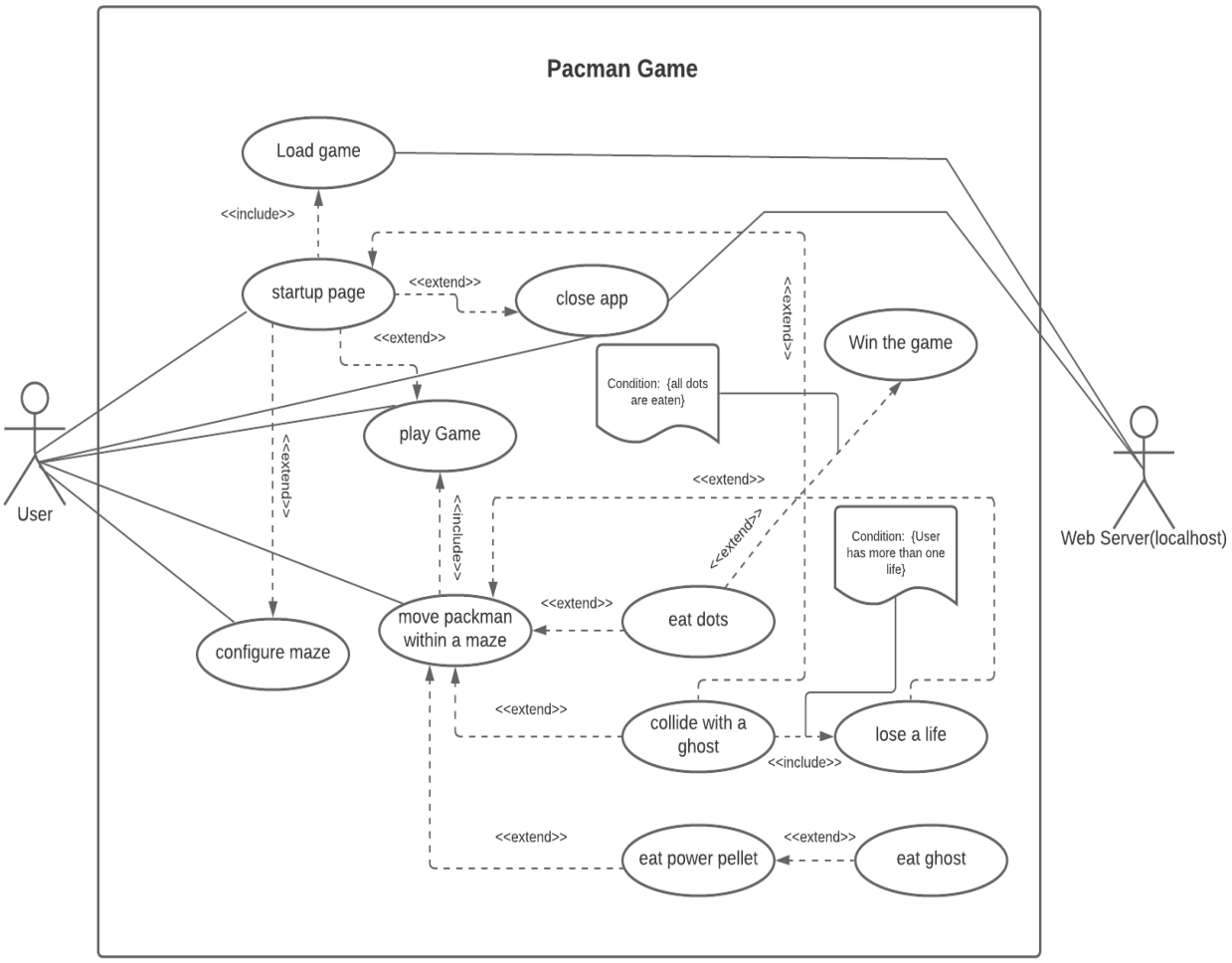
2.1 Functional requirements

Identifier	Requirement
REQ1	The game should initially load a startup page
REQ2	The start up page should contain the title and logo of Pac-Man
REQ3	The start up page should contain the year and course code
REQ4	The start up page should contain the list of all students in your group
REQ5	The start up page should contain an exit button
REQ6	The start up page should contain configure button.
REQ7	The start up page should contain a play button to take the player to the game
REQ8	The game application should at least be able to execute in 2 different operating systems
REQ9	User can choose the standard maze or random maze in the game configuration page
REQ10	The game must have an enclosed maze.
REQ11	The game must have 5 characters, 4 ghosts and a pacman
REQ12	The game has a centre box in the middle of the maze.
REQ13	The 4 coloured ghosts spawn from the centre-box when the game starts.
REQ14	PacMan has 3 lives at the beginning of every level.
REQ15	Remaining lives are shown in the game screen
REQ16	At the beginning of each game PacMan spawns below the center-box.
REQ17	Pacman moves inside the game maze as soon as the user presses any key
REQ18	The user can control PacMan direction to go up, down, right or left in the maze
REQ19	pacman will continue to move inside the maze unless it collides with a ghost or with maze wall
REQ20	Pacman can enter the warp tunnel and come out of the other warp tunnel.
REQ21	There are 4 “Power pellets” within the maze.
REQ22	Eating power pellet dots causes the 4 coloured ghosts to turn blue for a pre-set time.
REQ23	pacman can eat the ghosts if ghost color is blue
REQ24	After a certain pre-set time has elapsed the blue ghosts will turn back to their normal colours.
REQ25	When PacMan eats a blue ghost they disappear and respawn in the centre-box in its default colour
REQ26	Eating the ghosts adds 200 points to the scoreboard
REQ27	Eating a dot adds 10 points to the score for PacMan
REQ28	PacMan will lose a life if it collides with a ghost when its not blue
REQ29	When all three lives are lost the game is over.
REQ30	The game start again after Pacman loses a life if there is any life remaining.
REQ31	When PacMan eats all the dots in a level the user wins the game
REQ32	Pressing the exit button will close the application

2.2 Non-functional requirements

Requirement ID	Requirement type	Requirement description
REQ1	Performance	Minimum delay time regardless if game is hosted locally or on a server.
REQ2	Usability	All the actors in the game are clearly distinguishable. (User can tell which is PacMan, Ghosts etc.)
REQ4	Reliability	The game resumes every time without glitches after pausing the game.
REQ5	Usability	The game has different sound effects for different scenarios.
REQ6	Reliability	Game speed is reasonable at level and increments in reasonable steps for each level.
REQ7	Supportability	The game can load on multiple platforms

2.3 Use case diagram



UC1: Load Game
UC2: Startup Page
UC3: configure maze
UC4: Close App
UC5: Play Game
UC6: Move Pacman in a maze
UC7: Eat dots
UC8: Collide with a ghost
UC9 : lose a life
UC10: Eat Power Pellet
UC11: eat ghost
UC12: Win Game

2.4 Full use case description

Use case name	Win Game	
Scenario	Player Wins the game	
Triggering event:	User wants to win the game	
Brief description:	The player successfully eats all the dots (pellets) in the level, successfully avoids collision with the ghost.	
Actors:	User	
Related use cases:		
Stakeholders:	Developers, customers	
Preconditions:	User has a computer with monitor and keyboard web browser, Pacman game binaries are installed on the computer and the computer is in working condition. User has selected the play button, moved pacman to eat all the dots in the game without losing all lives.	
Postconditions:	User has one or more lives	
Flow of activities:	**Actor**	**System**
	1. a User indicates desire to play the Pacman and loads the game 2. the user enters start up page and selects start to play the game 3. the user interacts with the game using keyboard arrow buttons to move pacman 4. User eats all the dots by moving pacman. 5. User eats all the pellets in the level without losing all lives.	1. Game is loaded on the system providing output and input for the user to interact with 2.1. displays the start up menu page 2.2. navigate to the game play page 3.1. read in user input and move packman direction 3.2 remove each dot eaten from screen 4. remove all dots from screen. 5. Display game win prompt on screen.
Exception Condition:	1. System loses power. 2. System memory gets full 3. Lose access to localhost server.	

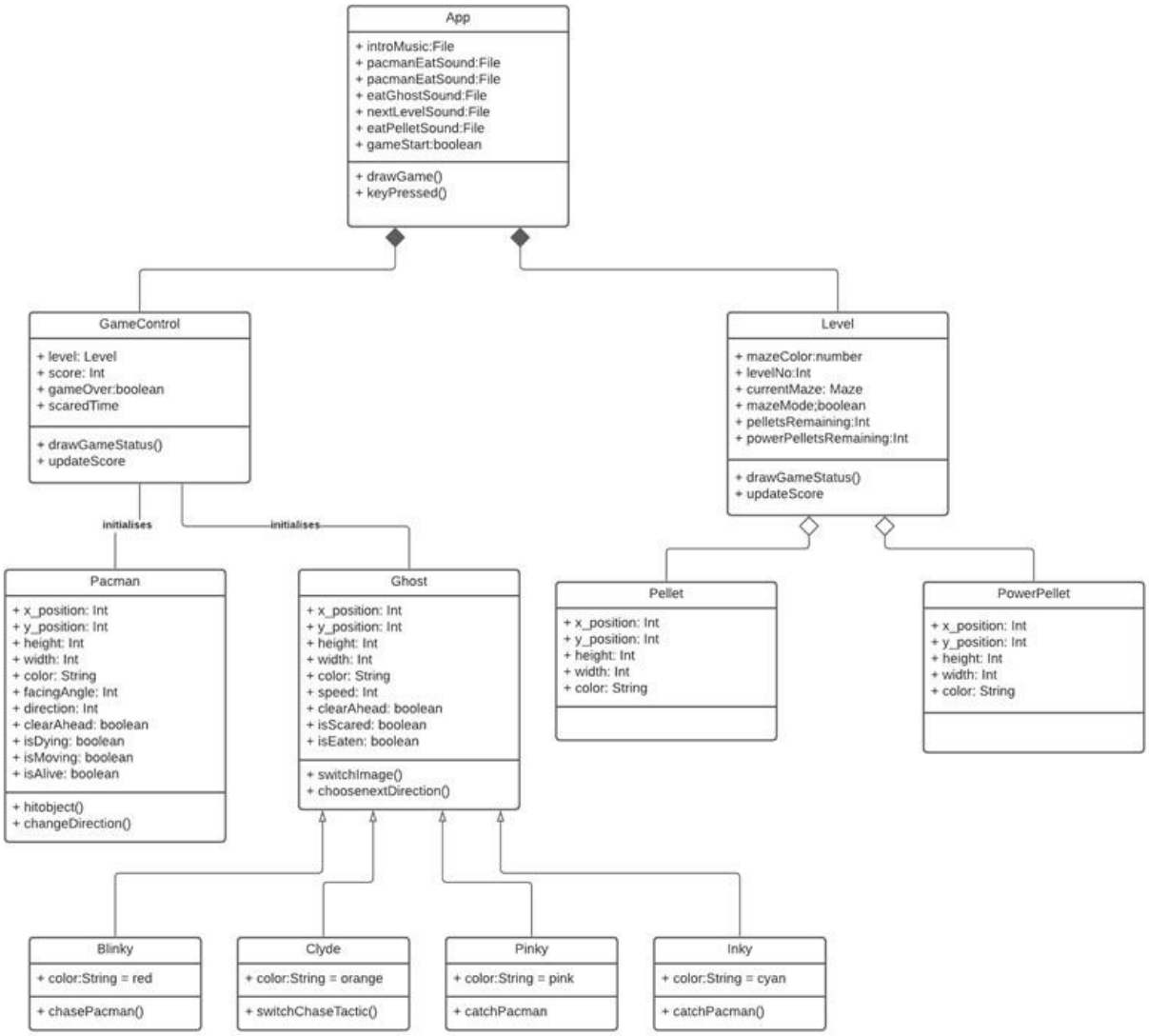
2.5 Requirement - use case traceability matrix

Req't	Count	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
REQ1	1	X											
REQ2	1		X										
REQ3	1		X										
REQ4	1		X										
REQ5	1		X										
REQ6	1		X										
REQ7	2		X			X							
REQ8	1	X											
REQ9	1			X									
REQ10	2					X	X						
REQ11	1					X							
REQ12	1					X							
REQ13	1					X							
REQ14	1					X							
REQ15	1					X							
REQ16	1					X							
REQ17	3			X		X	X						
REQ18	2					X	X						
REQ19	3			X		X	X						
REQ20	3			X		X	X						

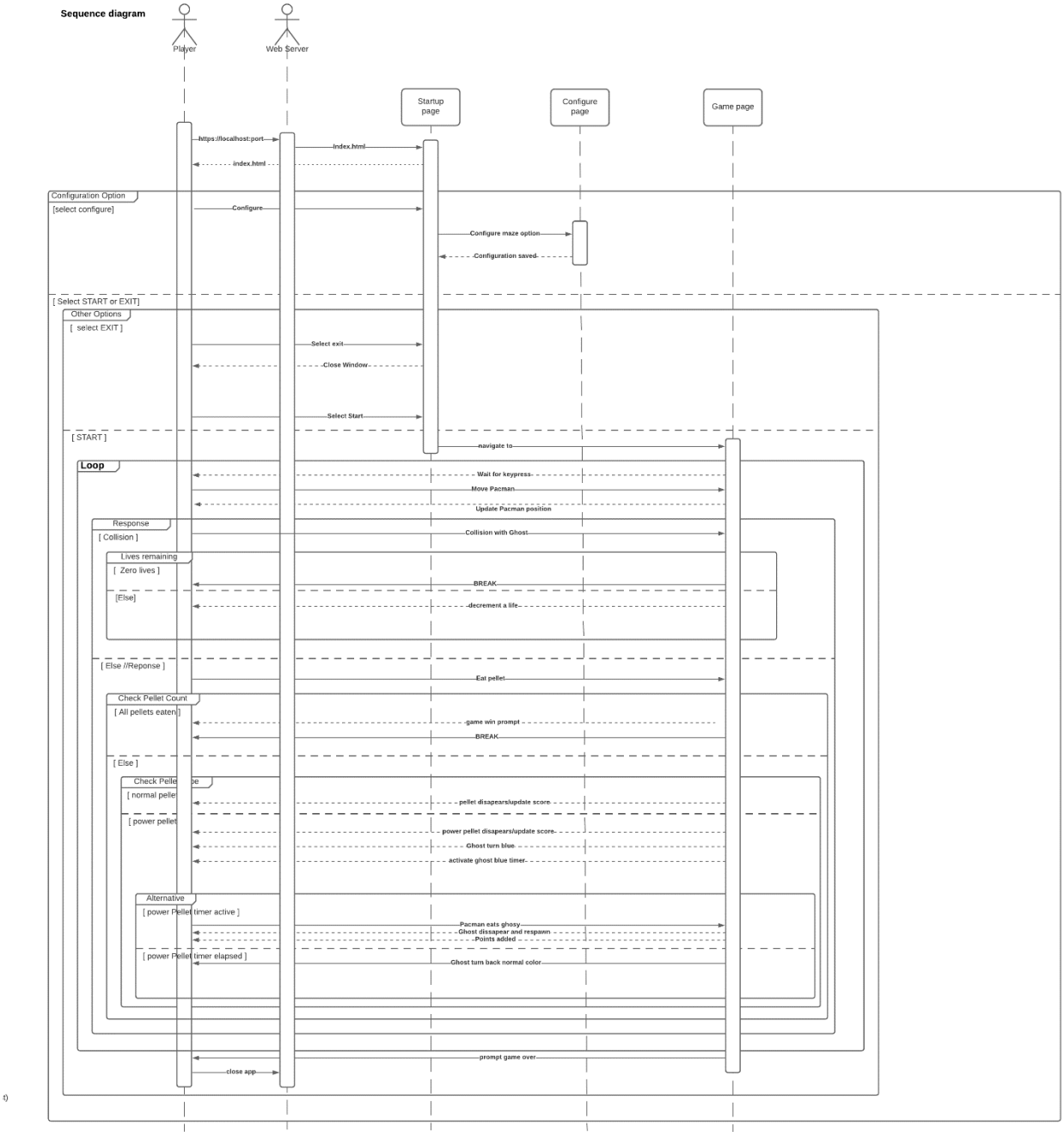
REQ21	1					X							
Req't	Count	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
REQ22	2					X					X		
REQ23	3					X					X	X	
REW24	2					X					X		
REQ25	3					X					X	X	
REQ26	3					X					X	X	
REQ27	3					X		X					X
REQ28	4					X	X		X	X			
REQ29	2		X			X							
REQ30	4					X	X		X	X			
REQ31	3					X	X	X					
REQ32	1				X								

3.0 Design and software architecture

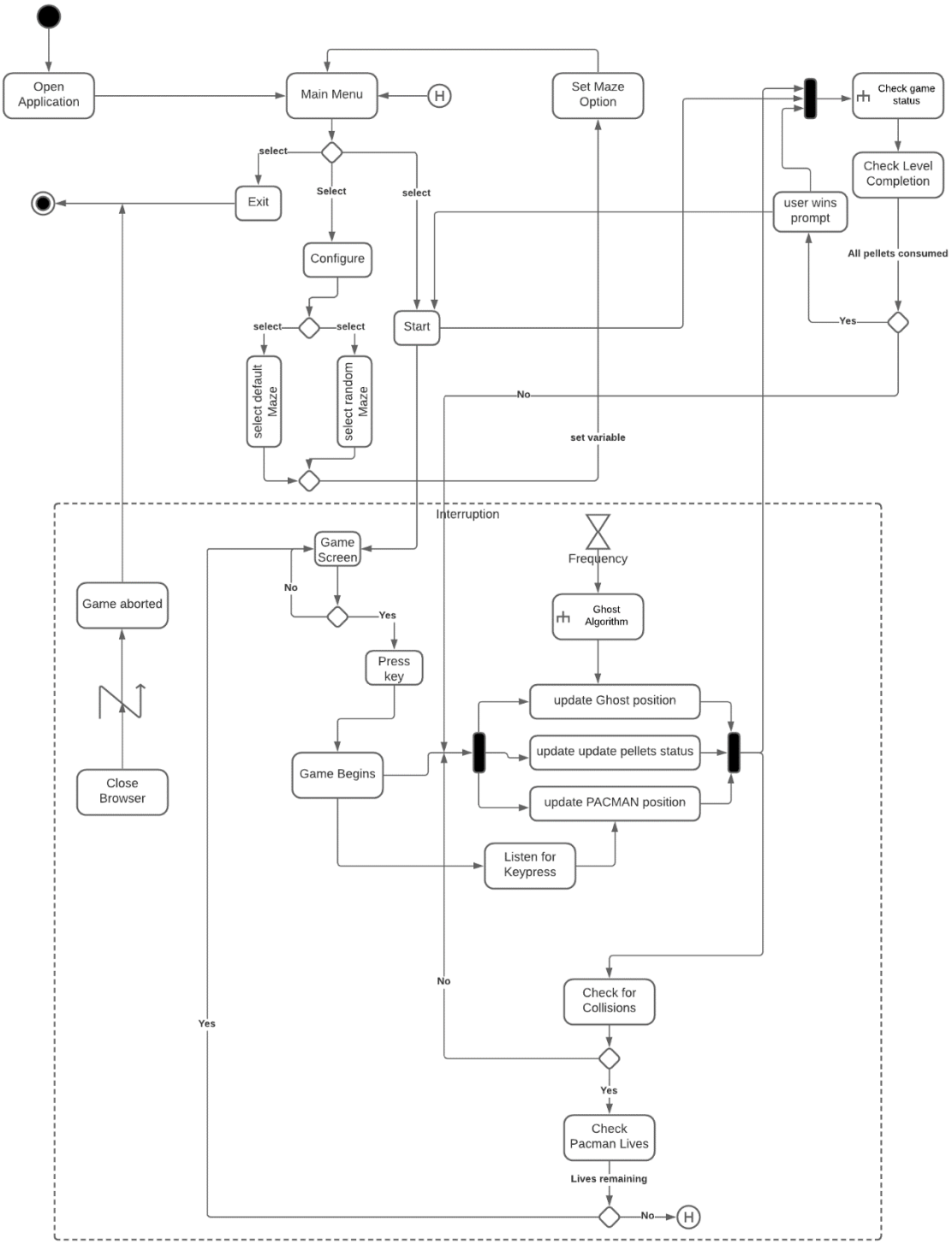
3.1 Class diagram



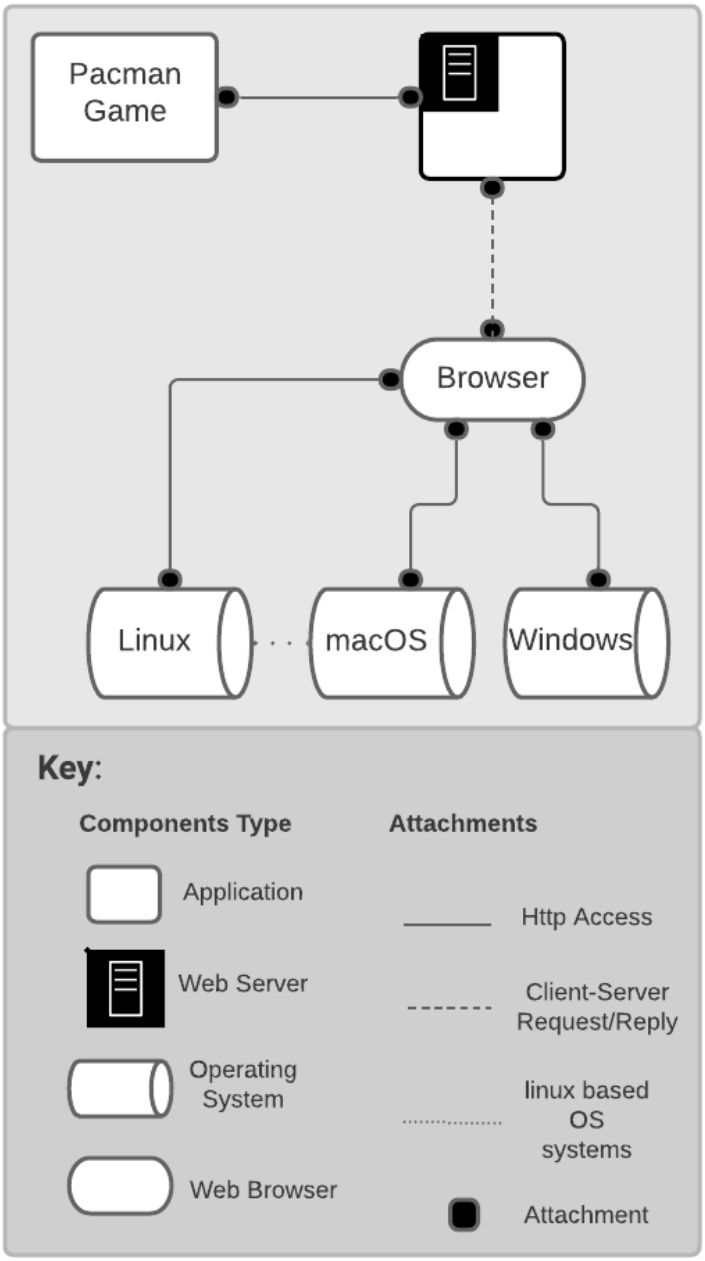
3.2 Sequence diagram



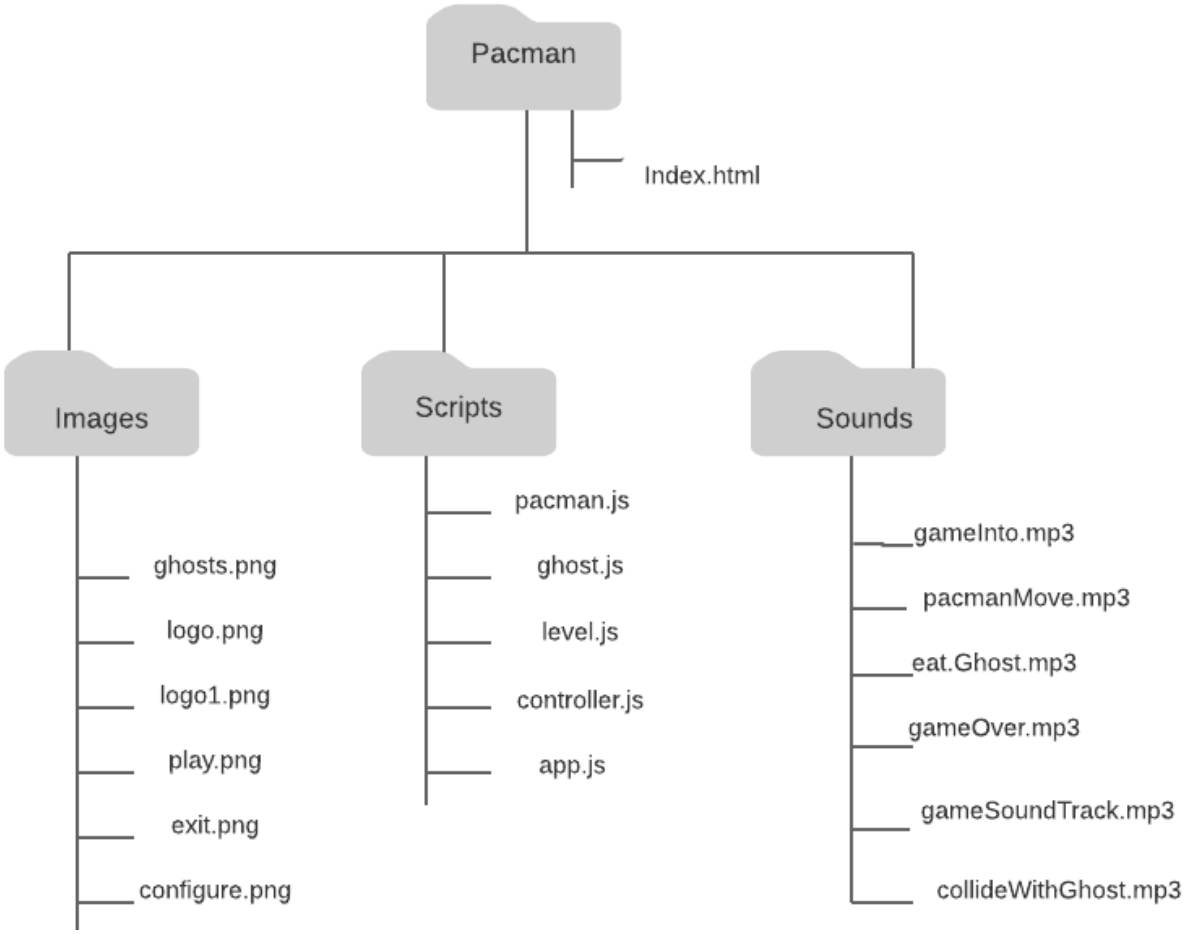
3.3 Activity diagram



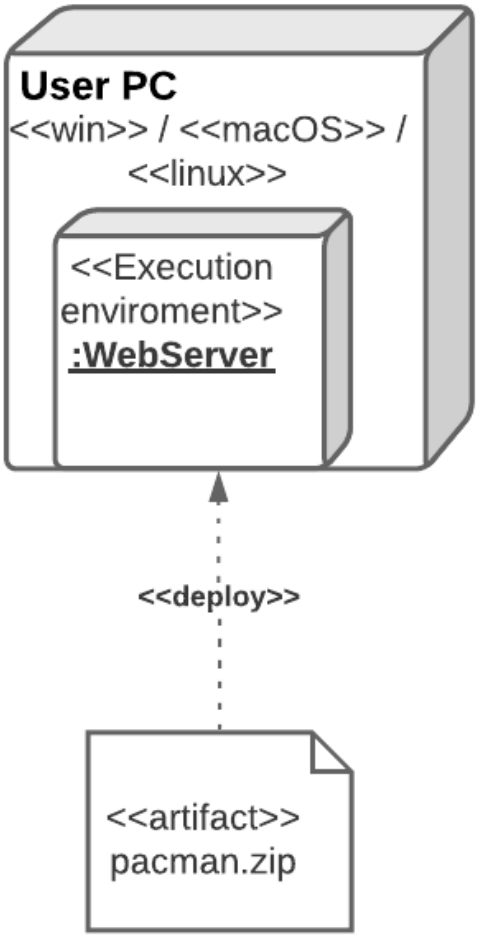
3.4 C & C View


Pacman Game
Browser
Linux
macOS
Windows
Key:
Components Type
Application
Web Server
Operating System
Web Browser
Attachments
Http Access
Client-Server Request/Reply
linux based OS systems
Attachment

3.5 Implementation style view



3.6 Deployment style view



4.0 Video link

[Click on the image below to redirect to video recording]