

Finals Weak: Project Part 2

Team: Brooke Robinson, Dylan Cole, Michael Feller, Jesus Ortiz Tovar

Title: Finals-Weak: *a game to prove who is weak and who is strong in the face of FINALS!*

Description: We will use the Lightweight Java Game Library (LWGL) to code a 2D, top-down game called "Finals-Weak" where the player must navigate various parts of campus to survive finals week at CU. The player can increase their health by picking up powerups and their health will decrease when they collide with various campus dangers (bikers, tray-carriers, talkative friends, etc.).

Data Storage: All data (highscores, save data, etc) will be stored in text files that will be hosted in the project folder.

Git-Hub Link: <https://github.com/brobinson124/finals-weak.git>

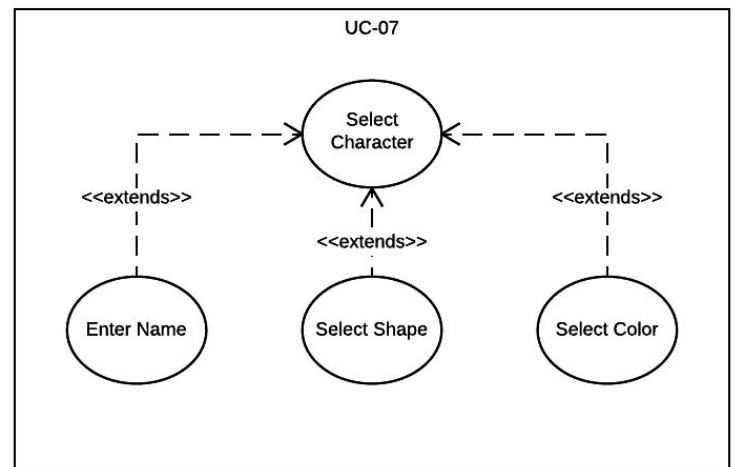
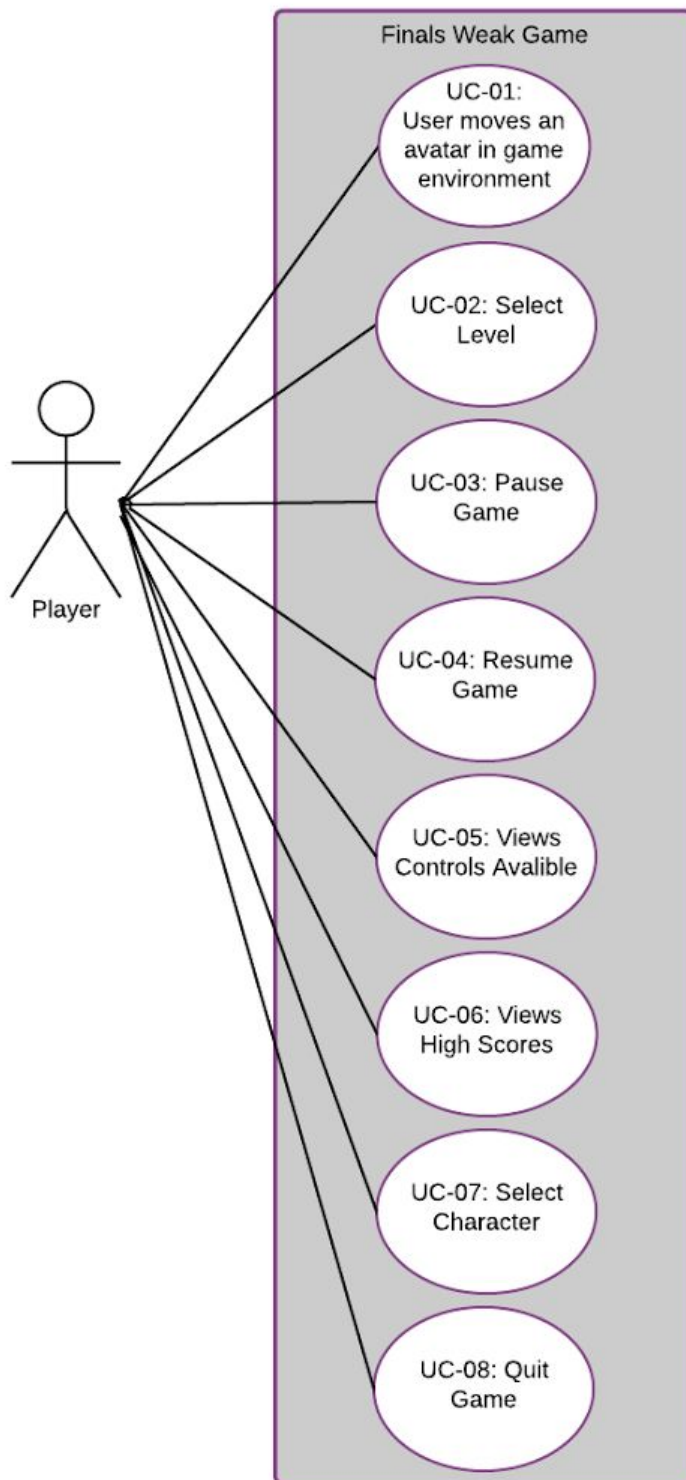
Table of Contents:

Project Requirements	2
Use Case Diagrams	3
Use Case Documents	4
Activity Diagrams	12
Mock Ups	16
Sequence Diagrams	19
Class Diagram	23
State Diagram	24

Project Requirements

Business Requirements			
No Business Requirements			
User Requirements			
ID	Requirement	Actor	Priority
UR-01	Easy to learn/Intuitive	User	Medium
UR-02	As a user I would like simple and responsive controls	User	Medium
UR-03	As a user I would like a way for the game to end	User	Low
UR-04	As a user I would like to play a game that provides sufficient challenge	User	High
Functional Requirements			
ID	Requirement	Actor	Priority
FR-01	Be able to handle I/O operations and move player accordingly	Developer	High
FR-02	End the levels when predefined objectives are met	Developer	Medium
FR-03	Prevent the user from putting the game in an unusable state	Developer	Medium
FR-04	Inform the user how to play the game	Developer	Low
FR-05	Provide a UI that adequately informs user of game events	Developer	Medium
Non-functional Requirements			
ID	Requirement	Actor	Priority
NR-01	Weekly meeting times for the team	Team members	High
NR-02	Running on different operating systems	Developer	High
NR-03	Multiple levels for variety within game	Developer	Low
NR-04	Comment code and make it readable	Developer	Medium
NR-05	As a developer I need a java library with game building (user inputs and rendering)	Developer	Critical
NR-06	As a developer I need Github for version control and collaboration	Developer	Critical
NR-07	Save user data (high scores, etc)	Developer	Low

Use Case Diagrams



Use Case Documents

Use Case ID:	UC-01
Use Case Name	Moves an avatar in game environment
Description:	User starts game and uses keyboard inputs to play the game

Actors:	User								
Pre-Conditions:	The game is currently running								
Post-Conditions:	The user completes/fails/quits the level								
Frequency of Use:	Every time the user starts a level								
Flow of Events:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>Press W, A, S, or D</td><td>-Checks potential avatar -Location -Updates avatar’s new location -Displays avatar’s location</td></tr></table>				Actor Action	System Response	1	Press W, A, S, or D	-Checks potential avatar -Location -Updates avatar’s new location -Displays avatar’s location
	Actor Action	System Response							
1	Press W, A, S, or D	-Checks potential avatar -Location -Updates avatar’s new location -Displays avatar’s location							
Variations:	1. Is solid object? <ul style="list-style-type: none">- Yes: don’t move location, exit- No: continue with system response 1. Is enemy? <ul style="list-style-type: none">- Yes: reduce player health, exit- No: continue with system response 1. Is powerup? <ul style="list-style-type: none">- Yes: increase player health, increase player stats, exit- No: continue with system response 1. Is objective? <ul style="list-style-type: none">- Yes: end level, exit- No: continue with system response								
Exceptions:	None								
Developers Notes:	Make it fun!								

Use Case ID:	UC-02
Use Case Name:	Selects Level
Description:	User selects a level from a selection of multiple environments to play.

Actors:	User								
Pre-Conditions:	At level select screen								
Post-Conditions:	User can choose a character (UC-07)								
Frequency of Use:	Whenever player wants to play level								
Flow of Events:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>User selects level from options on main menu</td><td>-Saves level selection -Displays character creation options and prompts user (see UC-07)</td></tr></table>				Actor Action	System Response	1	User selects level from options on main menu	-Saves level selection -Displays character creation options and prompts user (see UC-07)
	Actor Action	System Response							
1	User selects level from options on main menu	-Saves level selection -Displays character creation options and prompts user (see UC-07)							
Variations:	None								
Exceptions:	User selects one of the options								
Developers Notes:	Have at least 4 level choices.								

Use Case ID:	UC-03
Use Case Name:	User Pauses Game
Description:	User presses pause button to temporarily stop the game

Actors:	User							
Pre-Conditions:	Game is running							
Post-Conditions:	Game is paused							
Frequency of Use:	Whenever the user wants to stop playing temporarily							
Flow of Events:	<table border="1"> <thead> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User presses pause key (space bar)</td><td> -Stop timers/counters -Halt player movement -Halt enemy movement -Display pause screen </td></tr> </tbody> </table>			Actor Action	System Response	1	User presses pause key (space bar)	-Stop timers/counters -Halt player movement -Halt enemy movement -Display pause screen
	Actor Action	System Response						
1	User presses pause key (space bar)	-Stop timers/counters -Halt player movement -Halt enemy movement -Display pause screen						
Variations:	None							
Exceptions:	None							
Developers Notes:	Make it obvious how to pause game.							

Use Case ID:	UC-04
Use Case Name:	Resumes Game
Description:	User presses resume button to resume game

Actors:	User								
Pre-Conditions:	Game is paused								
Post-Conditions:	Game is running and user has control of avatar								
Frequency of Use:	Whenever User wants to resume game from a paused screen								
Flow of Events:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>User presses pause key (space bar)</td><td>-Resume timer/counters -Resume Game State -Resume User State -Remove pause screen</td></tr></table>				Actor Action	System Response	1	User presses pause key (space bar)	-Resume timer/counters -Resume Game State -Resume User State -Remove pause screen
	Actor Action	System Response							
1	User presses pause key (space bar)	-Resume timer/counters -Resume Game State -Resume User State -Remove pause screen							
Variations:	None								
Exceptions:	None								
Developers Notes:	Be obvious how to resume game from paused screen								

Use Case ID:	UC-05
Use Case Name:	Views Controls Available
Description:	User can toggle controls to see what keys are sticky in order to properly play game

Actors:	User		
Pre-Conditions:	Game running		
Post-Conditions:	None		
Frequency of Use:	Engaged when user first starts game, can be turned on/off at any time		
Flow of Events:			
		Actor Action	System Response
	1	User presses control display key (c) at any time	-Controls for each key are displayed on screen
	2	When controls are displayed, press (c) to close display	-Control display disappears from screen
Variations:	None		
Exceptions:	None		
Developers Notes:	When game is opened, this option is toggled on		

Use Case ID:	UC-06
Use Case Name:	Views High Scores
Description:	User can view high scores from main menu.

Actors:	User										
Pre-Conditions:	User is in main menu.										
Post-Conditions:	User is able to view high scores/stats.										
Frequency of Use:	As often as the user cares to look at scores.										
Flow of Events:	<table border="1"> <thead> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>User selects "High Scores" option from main menu screen</td><td>-Queries and updates current high score data from log. -Displays High Scores screen.</td></tr> <tr> <td>2</td><td>User selects "Return to Main Menu"</td><td>-Displays Main Menu (level select screen)</td></tr> </tbody> </table>			Actor Action	System Response	1	User selects "High Scores" option from main menu screen	-Queries and updates current high score data from log. -Displays High Scores screen.	2	User selects "Return to Main Menu"	-Displays Main Menu (level select screen)
	Actor Action	System Response									
1	User selects "High Scores" option from main menu screen	-Queries and updates current high score data from log. -Displays High Scores screen.									
2	User selects "Return to Main Menu"	-Displays Main Menu (level select screen)									
Variations:	None										
Exceptions:	None										
Developers Notes:	Log is a file stored locally.										

Use Case ID:	UC-07
Use Case Name:	Chooses Character
Description:	User selects a character's color and shape from predetermined options and is able to input a name.

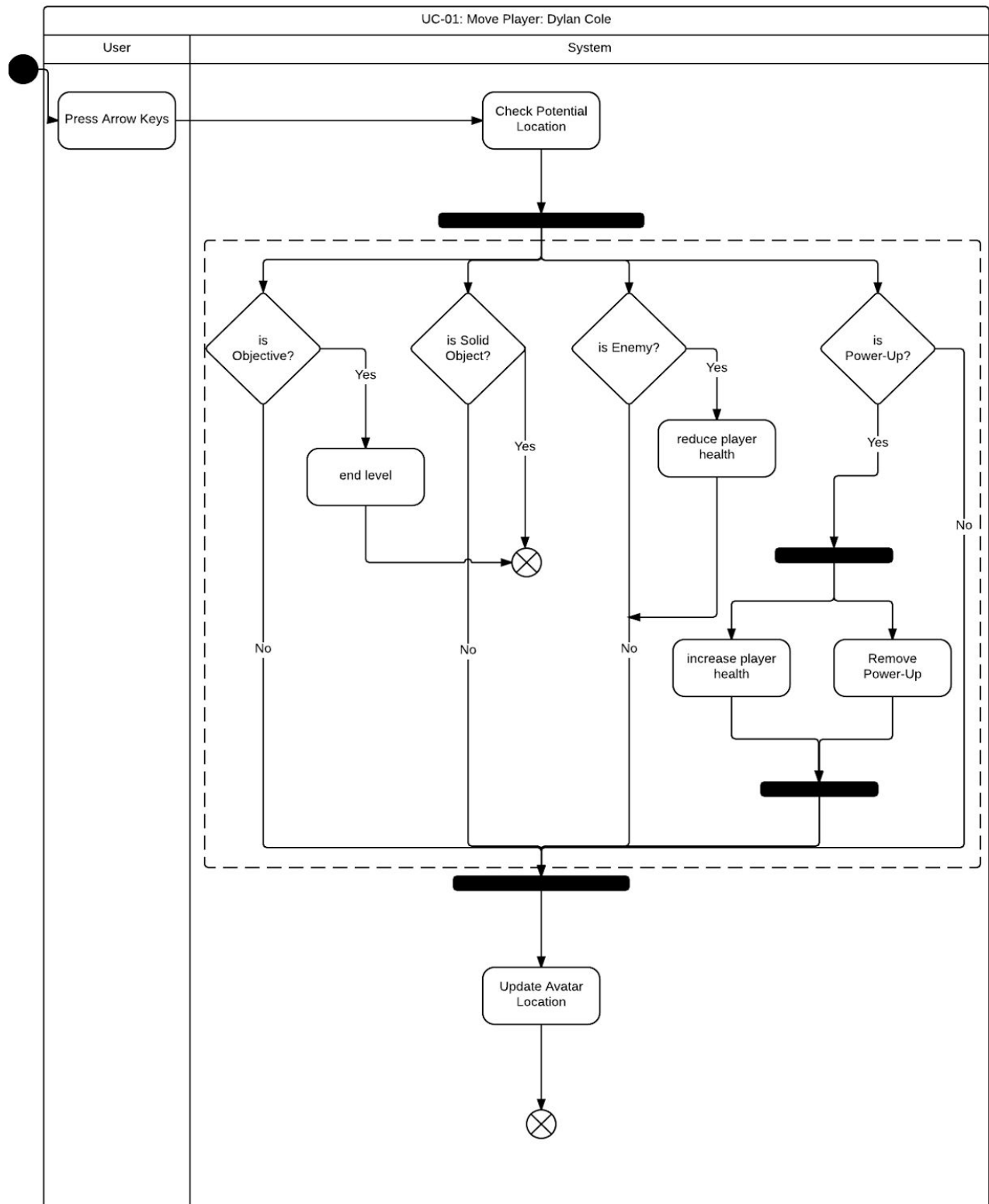
Actors:	User		
Pre-Condition:	User has selected level (UC-02)		
PostCondition :	Level is loaded and created avatar is placed in game environment		
Frequency Use:	Every time user selects a level		
Flow of Events:		Actor Action	System Response
	1	Selects level from UC-02	-Prompts user for character name
	2	Enter desired name	-Checks name for validity -Sets name -Displays shape options
	3	Selects desired shape	-Sets shape selection -Display shape in color options
	4	Selects desired color	-Sets color choice -Displays level with player character rendering in selected shape and color with name as label at start location
Variations:	User can choose various names, shapes and colors.		
Exceptions:	Name must be more than 0 characters (non-empty) and less than 8 characters.		
Developers	None		

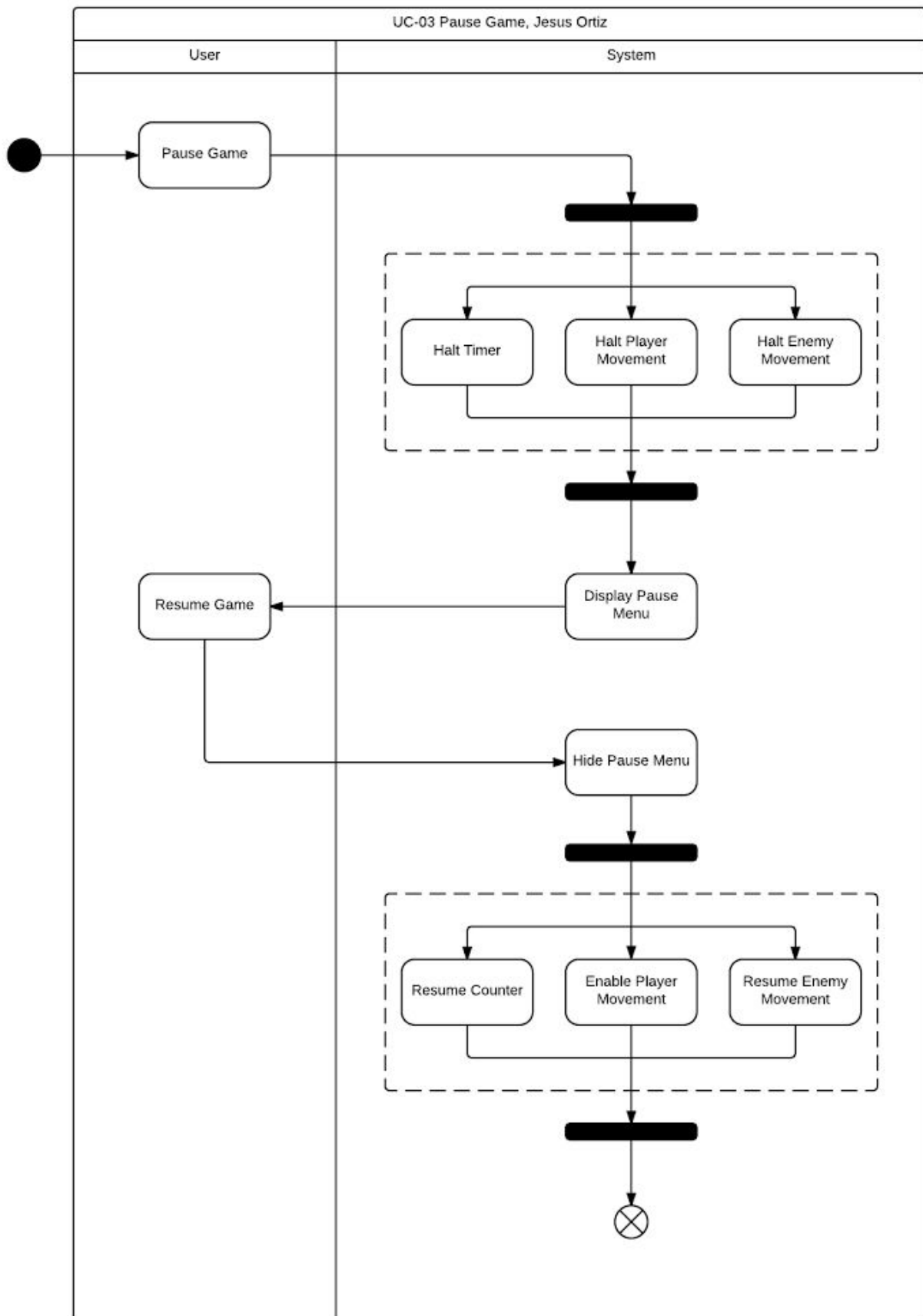
Notes:	
---------------	--

Use Case ID:	UC-08
Use Case Name:	Quits Game
Description:	Ability to exit application at any point in application

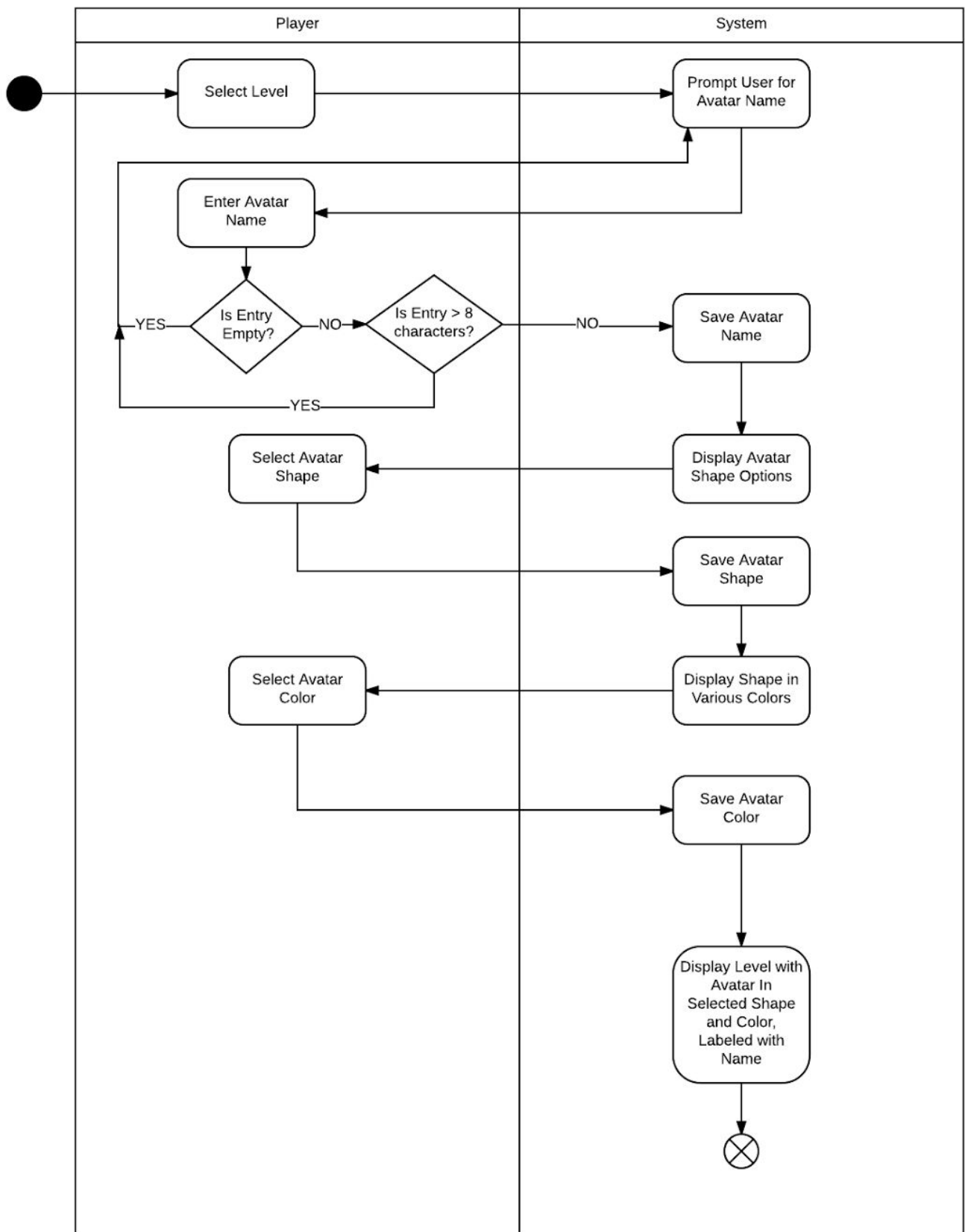
Actors:	User										
Pre-Conditions:	Game window is open										
Post-Conditions:	Application is closed										
Frequency of Use:	Whenever a user wants to quit a game										
Flow of Events:	<table border="1"> <thead> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Presses presses "ESC" key</td><td>-Checks if game is running mid-level -If yes, prompt user to confirm choice to quit</td></tr> <tr> <td>2</td><td>Click "Confirm" or "Cancel"</td><td>-If Confirm, close game window -If Cancel, resume game</td></tr> </tbody> </table>			Actor Action	System Response	1	Presses presses "ESC" key	-Checks if game is running mid-level -If yes, prompt user to confirm choice to quit	2	Click "Confirm" or "Cancel"	-If Confirm, close game window -If Cancel, resume game
	Actor Action	System Response									
1	Presses presses "ESC" key	-Checks if game is running mid-level -If yes, prompt user to confirm choice to quit									
2	Click "Confirm" or "Cancel"	-If Confirm, close game window -If Cancel, resume game									
Variations:	If game is not mid-level (i.e. game is sitting at level select or high score screens), close the game window without prompt										
Exceptions:	None										
Developers Notes:	User can quit from any screen in application										

Activity Diagrams

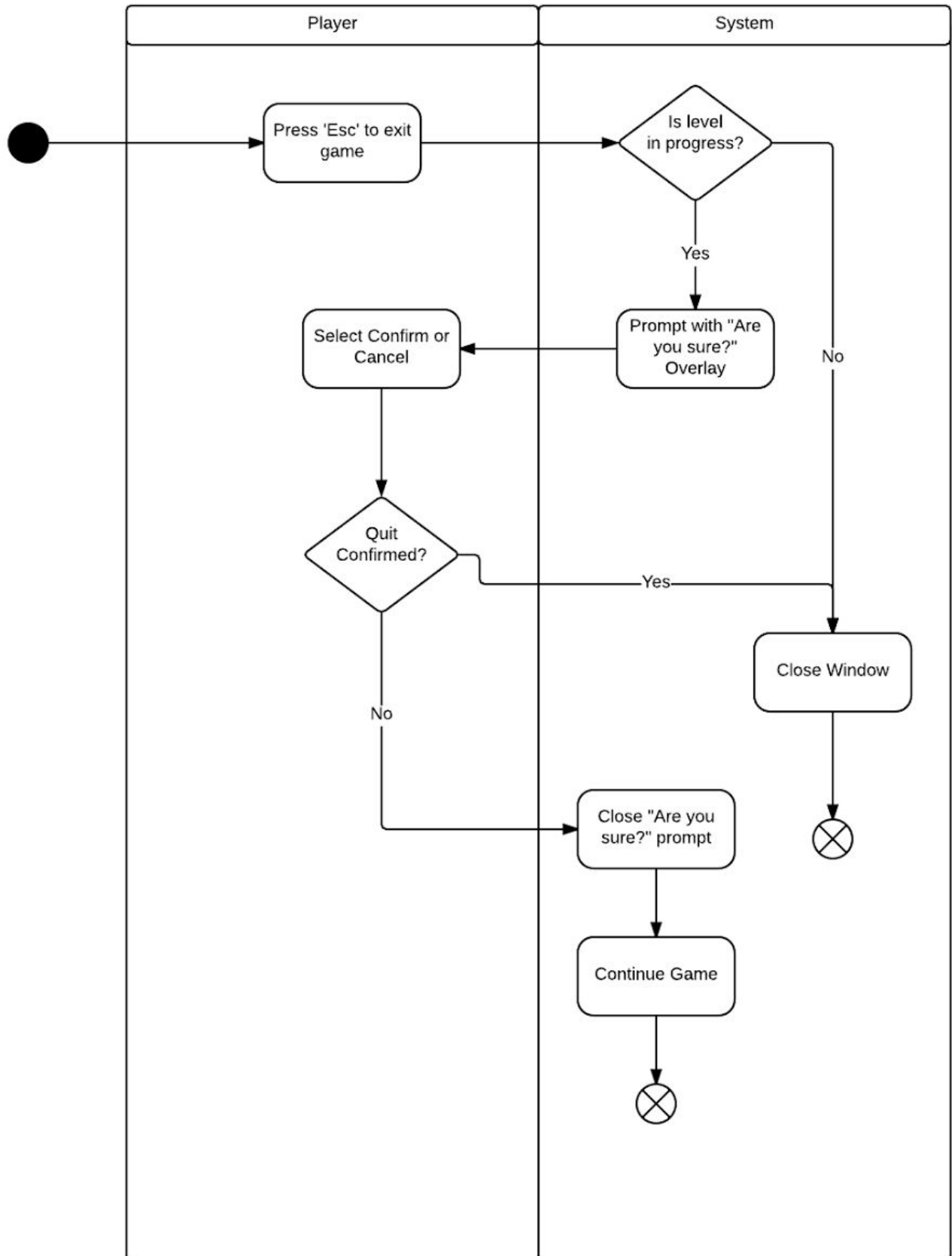




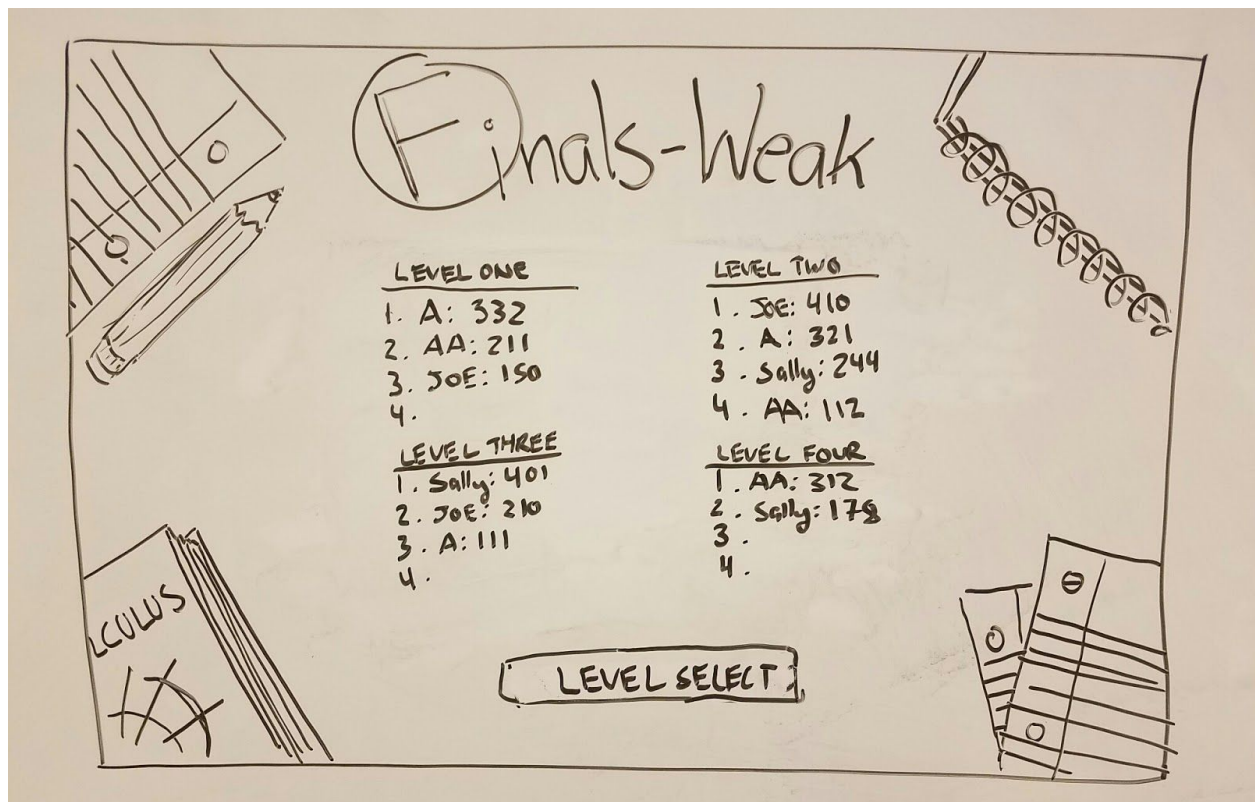
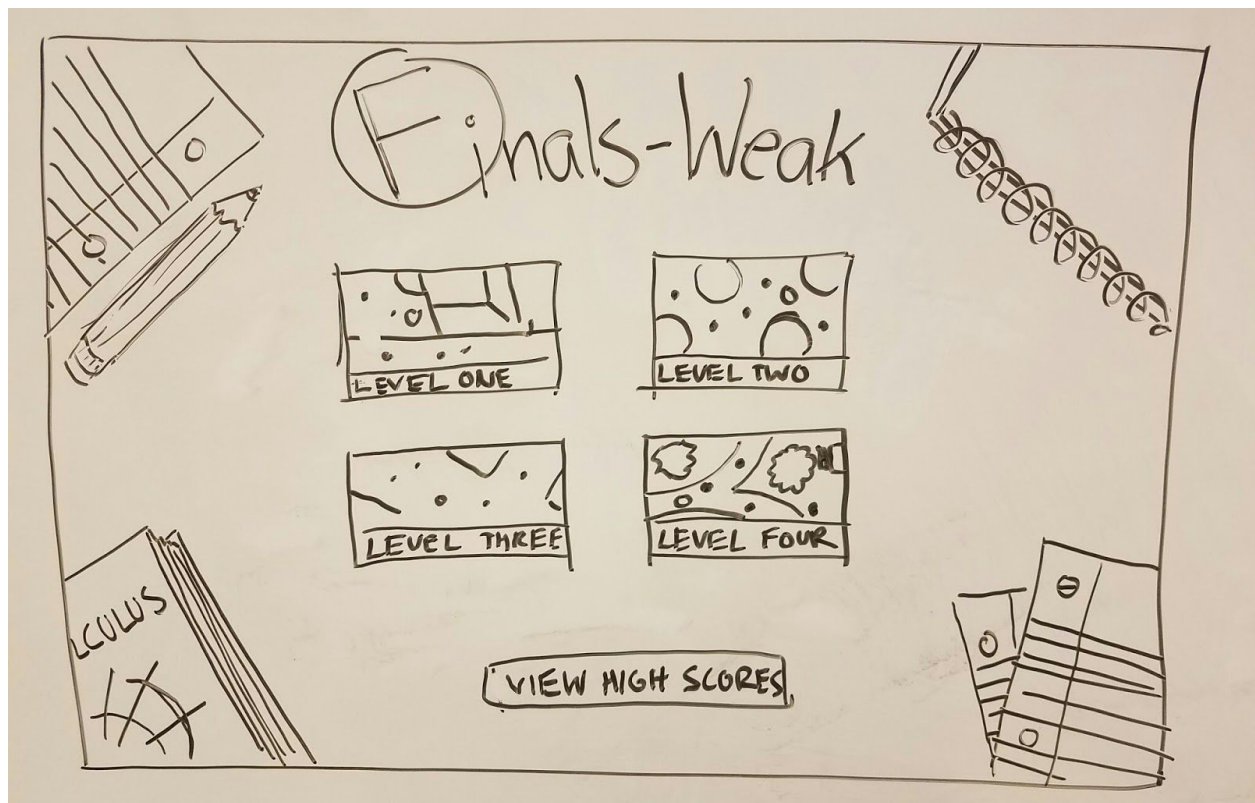
UC-07: CHOOSE CHARACTER
Michael Feller



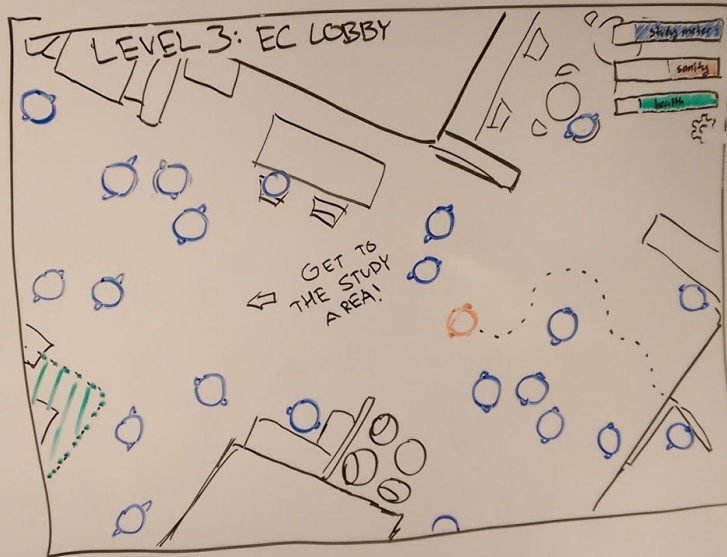
UC-08: Quits Game
Brooke Robinson



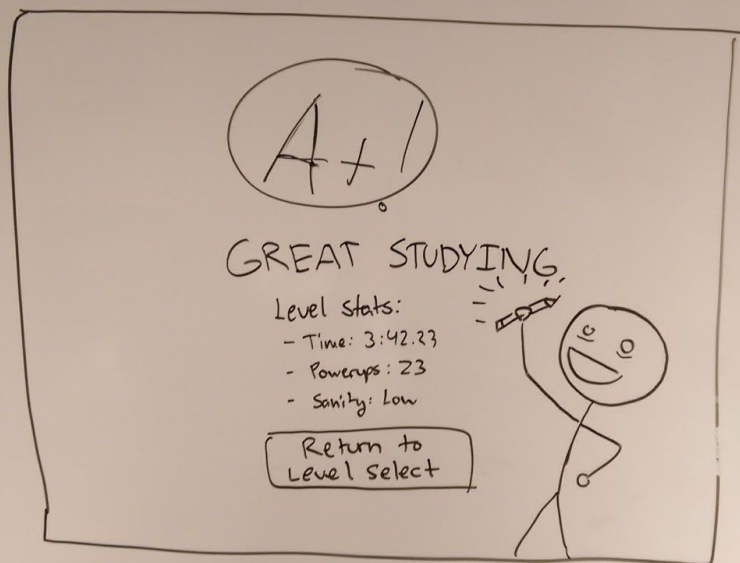
Mock Ups



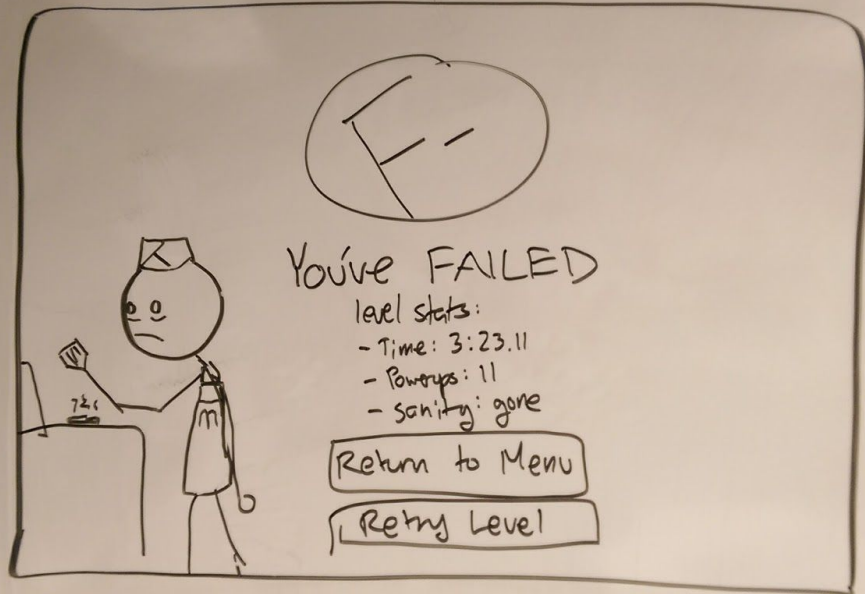
GAME SCREEN



COMPLETION SCREEN

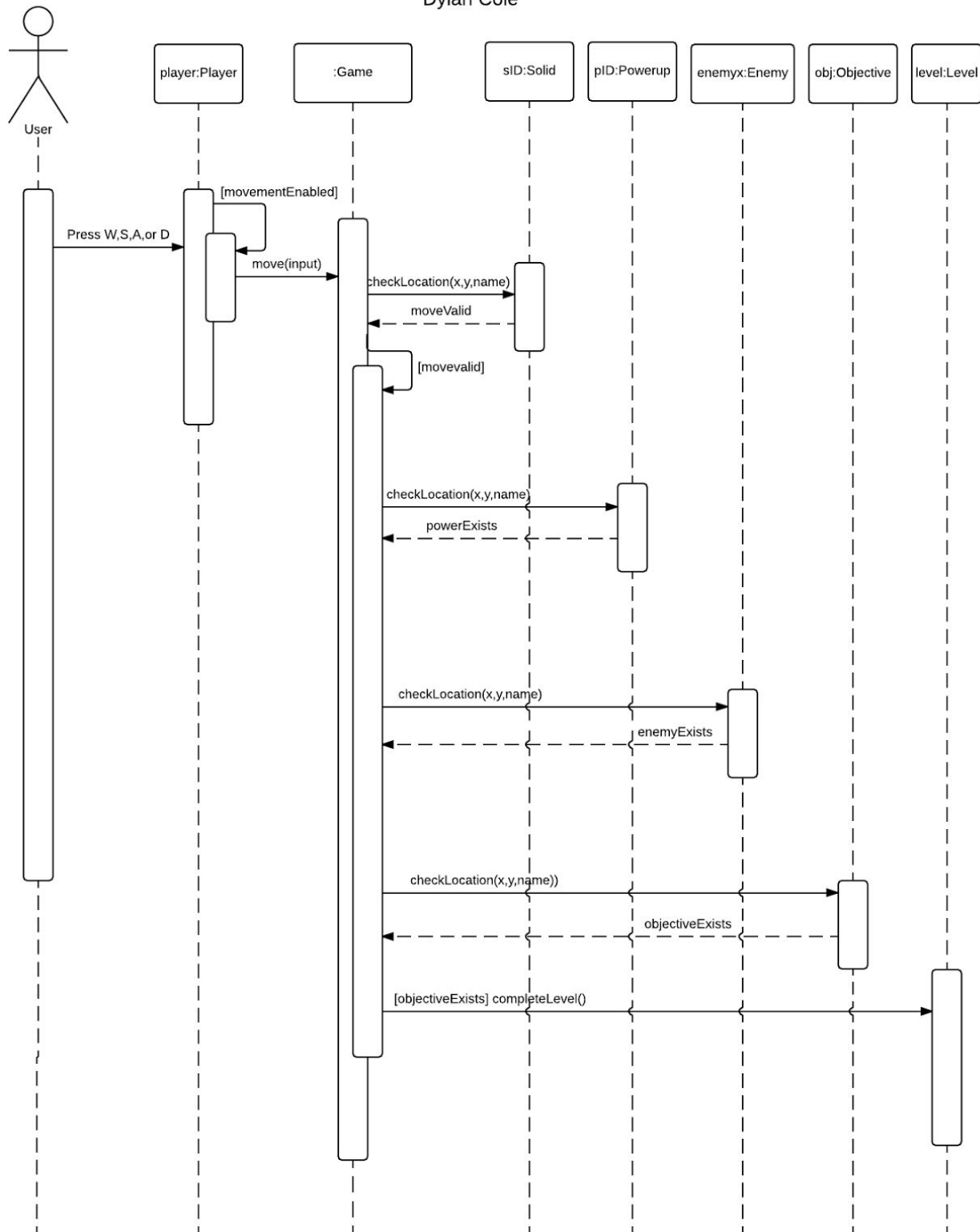


Failure Screen

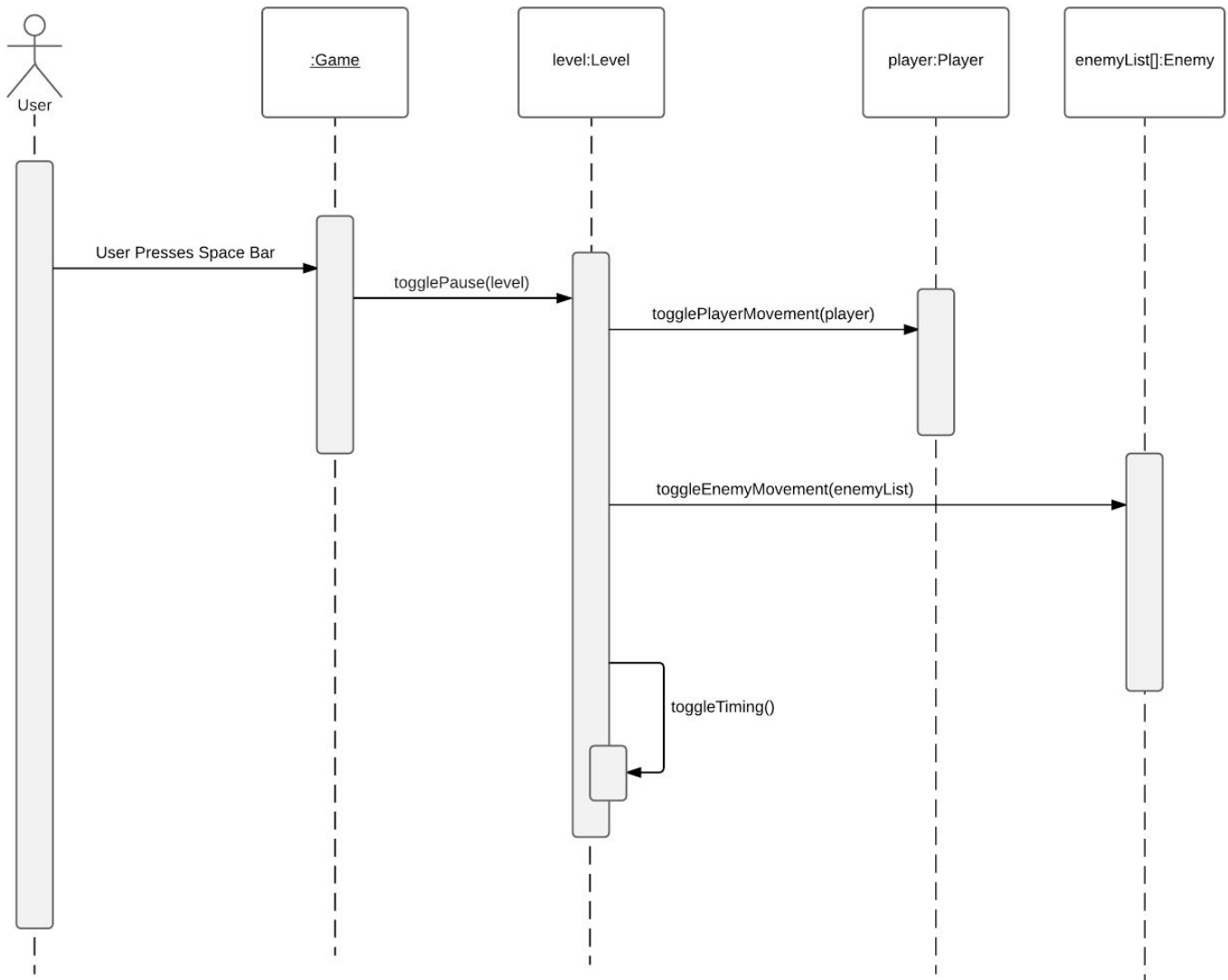


Sequence Diagrams

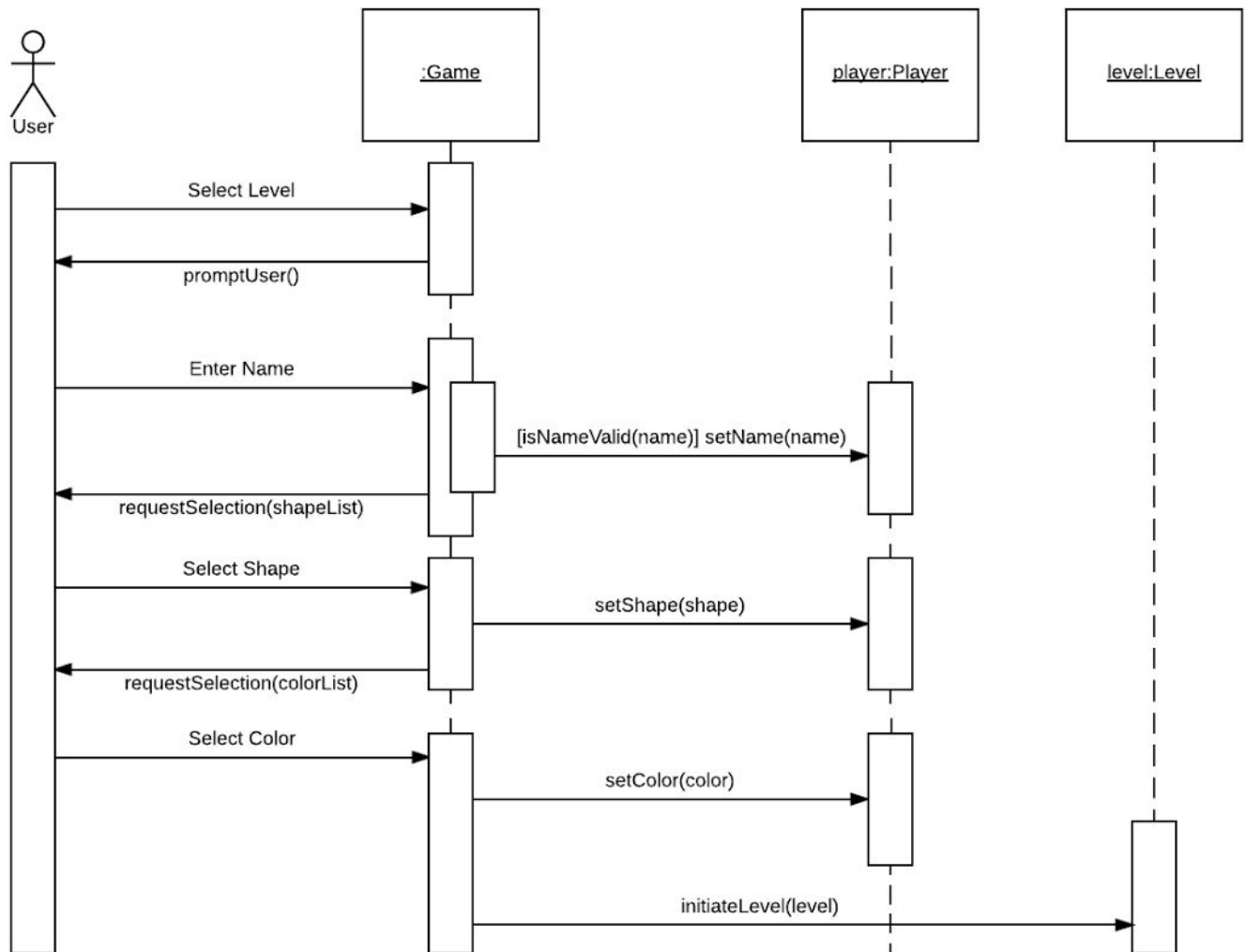
UC-01: Moves an avatar in game environment
Dylan Cole



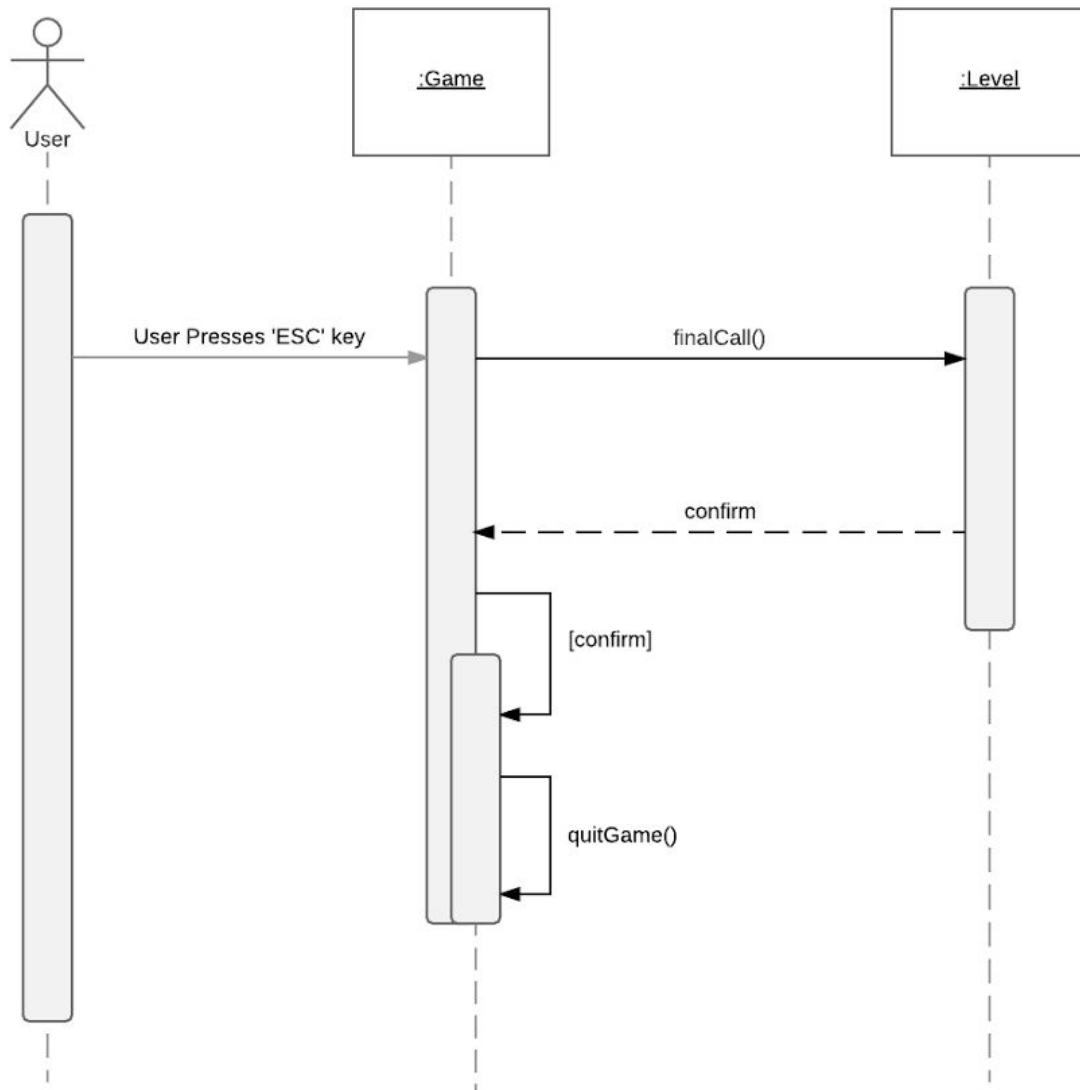
UC-03 User Pauses Game
Jesus Ortiz Tovar



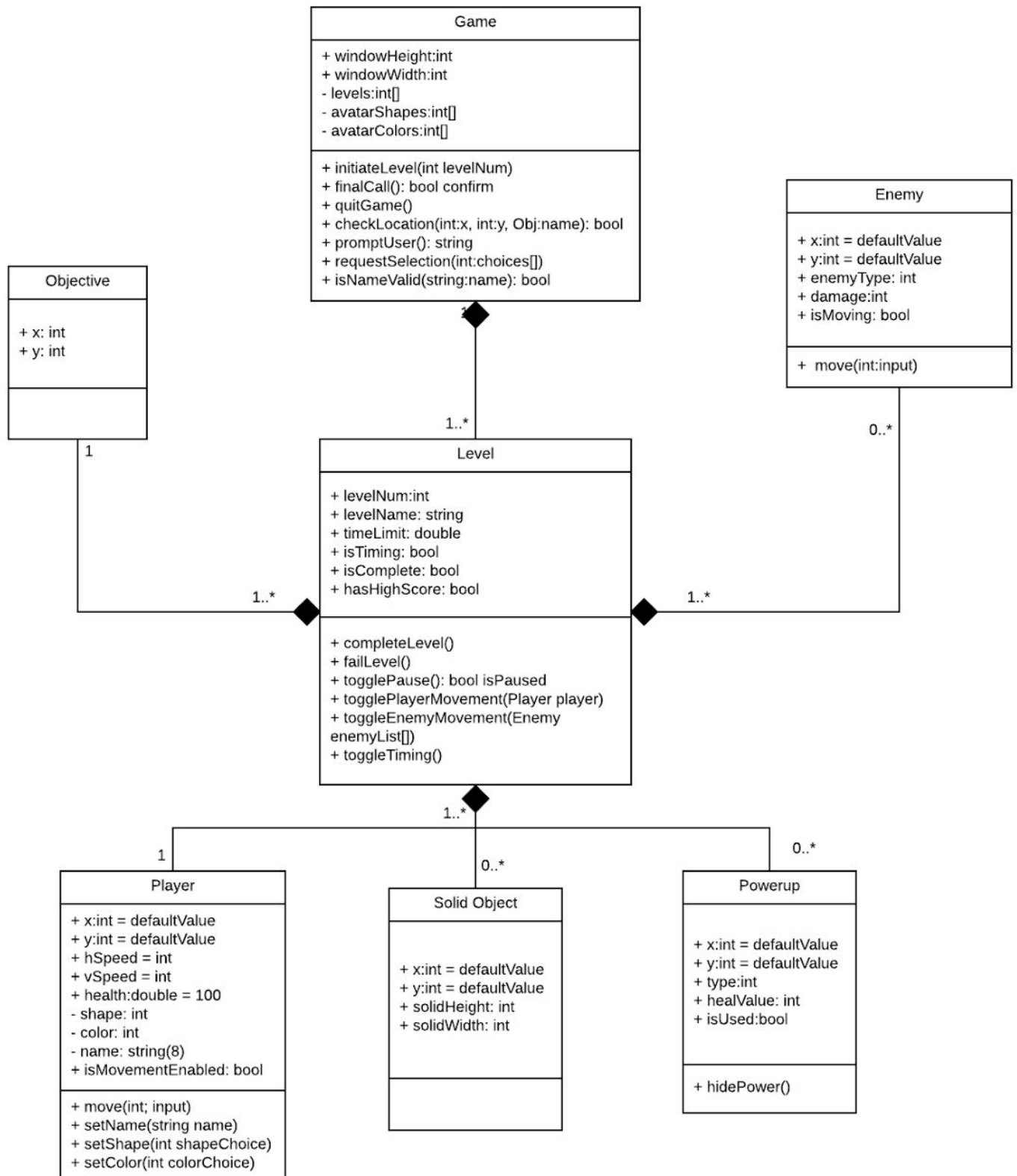
UC-07: Choose Character Sequence
Diagram
Michael Feller



UC-08: Quits Game Sequence
Diagram
Brooke Robinson



Class Diagram



State Diagram

