

Team: Ian Moore, Sam Klopsch, Dylan McKinney, Michael Chung

Title: MoodleJump

Summary: MoodleJump is a game which we hope emulates the hit game “DoodleJump”. The purpose of the game is for the user to control a single character, ‘jumping’ vertically from platform to platform in order to achieve the highest possible score before failing a jump, or ‘dying’ by coming in contact with an enemy.

Requirements:

Business: There are no business requirements for our project.

User:

ID	Description	Priority
US-01	As a player, I want to play a different level with each attempt. The map should not be uniform.	Critical
US-02	As a player, I would like to pause the game if I get distracted and need to do something else.	Medium
US-03	As a player, I want to be able to get the high score	High

Functional:

ID	Description	Priority
FR-01	The leaderboard should populate itself with the current high-scores in the game.	High
FR-02	The game will generate maps randomly, ensuring no player will have to play the same map twice.	Critical
FR-03	The game will take in player’s input, and change its state accordingly.	Critical

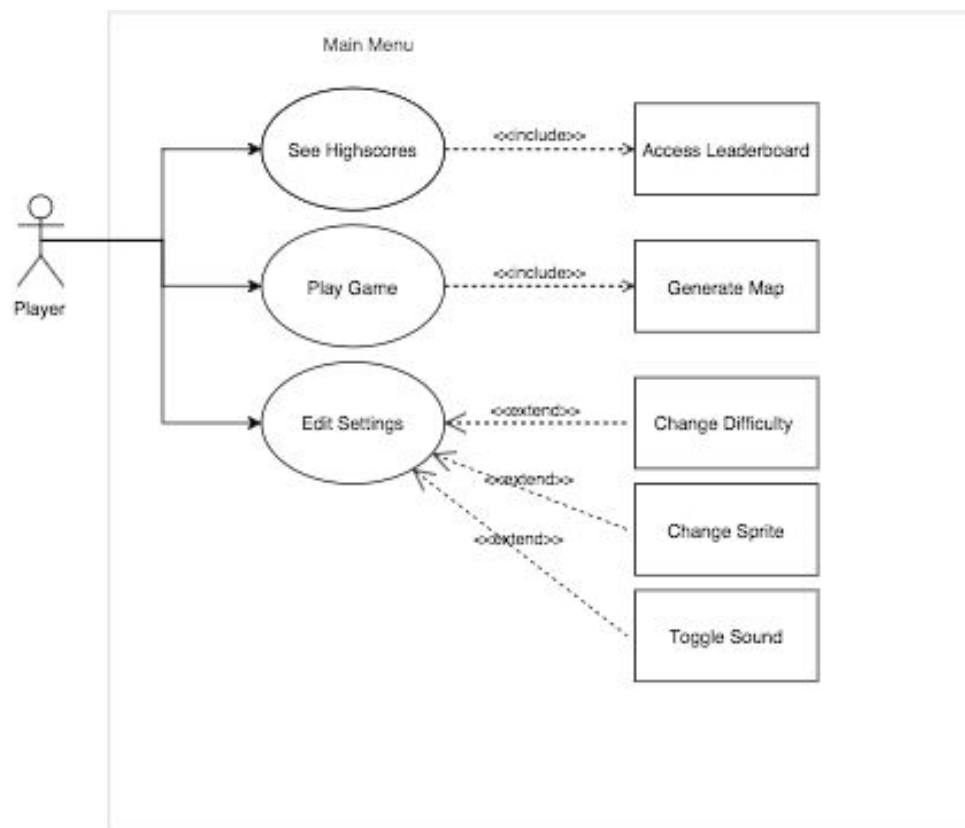
Non-Functional:

ID	Description	Priority
NFR-01	Although the map will be randomly generated, the game will never be impossible. Platforms will always be reachable.	Critical
NFR-02	The lag between player input and system reaction will be a non-factor. No player will feel as though lag caused them to lose the game	High

NFR-03	As we update and add features, the game will remain playable. Updates will also be pushed based off of bugs discovered during release stages.	Medium
--------	---	--------

Users and Tasks:

For our game, we will only have one user, which is the player. The task they are trying to accomplish is navigating a character to advance further into the game in order to set a high score. The player may also see the highscores and edit settings.



Use Case ID:	UC-01
Use Case Name:	Play the Game
Description:	Player can select "Play Game" on home menu.

Actors:	Player													
Pre-Conditions:	Game must be installed on a supported device.													
Post-Conditions:	Player is launched into a randomly generated map.													
Frequency:	Whenever the player wishes to play													
Flow of Events:	<table border="1"> <thead> <tr> <th></th><th>Player Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Press game icon</td><td>Launch the game</td></tr> <tr> <td>2</td><td>Select "Play Game" in home menu</td><td>Begin map generation</td></tr> <tr> <td>3</td><td>Play the Game</td><td>Varies based on player interaction.</td></tr> </tbody> </table>			Player Action	System Response	1	Press game icon	Launch the game	2	Select "Play Game" in home menu	Begin map generation	3	Play the Game	Varies based on player interaction.
	Player Action	System Response												
1	Press game icon	Launch the game												
2	Select "Play Game" in home menu	Begin map generation												
3	Play the Game	Varies based on player interaction.												
Variations:	N/A													
Notes and Issues:	N/A													
Developer Notes:	N/A													

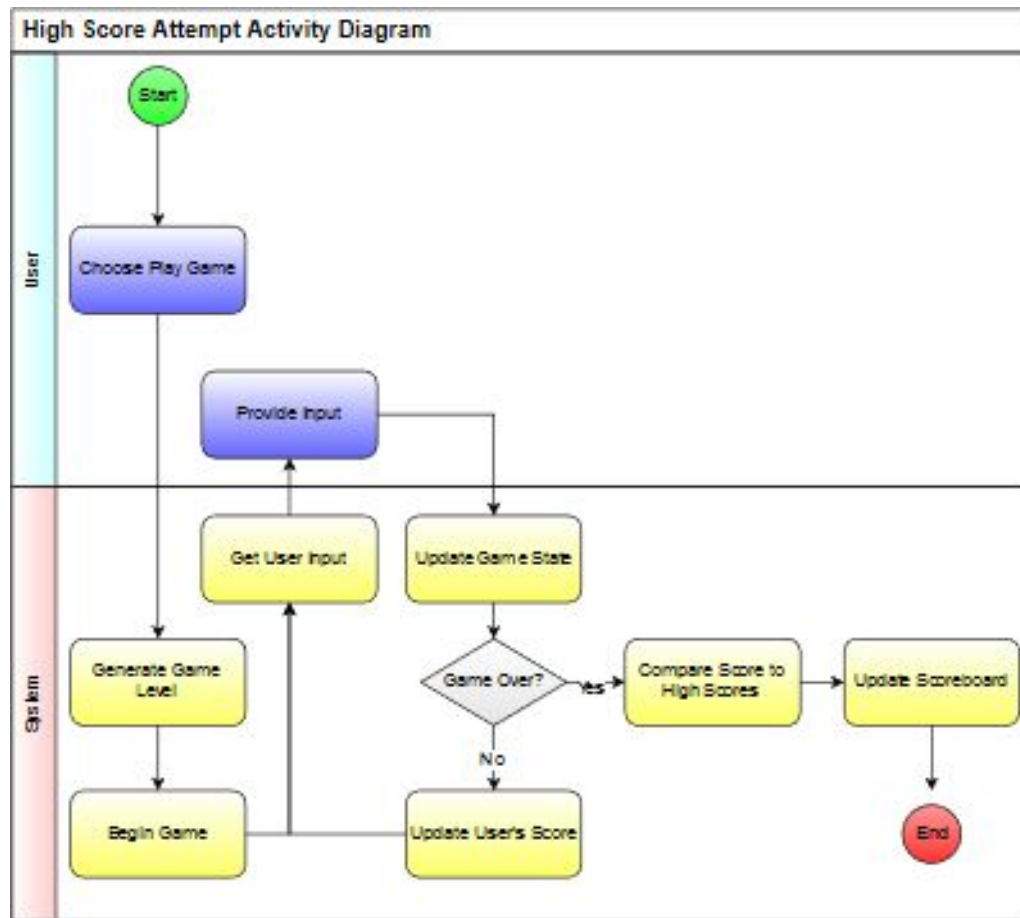
Use Case ID:	UC-02
Use Case Name:	Check the leaderboard
Description:	Player can view their score, in relation to top scores.

Actors:	Player										
Pre-Conditions:	Game must be connected to the internet.										
Post-Conditions:	Player gets the information they desire.										
Frequency:	Whenever the player wishes to view this information										
Flow of Events:	<table border="1"> <thead> <tr> <th></th><th>Player Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Press "High Scores"</td><td>Load the leaderboard</td></tr> <tr> <td>2</td><td>View the scores</td><td>Display the leaderboard</td></tr> </tbody> </table>			Player Action	System Response	1	Press "High Scores"	Load the leaderboard	2	View the scores	Display the leaderboard
	Player Action	System Response									
1	Press "High Scores"	Load the leaderboard									
2	View the scores	Display the leaderboard									
Variations:	N/A										
Notes and Issues:	N/A										
Developer Notes:	N/A										

Use Case ID:	UC-03
Use Case Name:	Edit the settings.
Description:	Player can edit the game settings.

Actors:	Player											
Pre-Conditions:	None											
Post-Conditions:	Player's adjustments change the system accordingly.											
Frequency:	Whenever the player wishes to play											
Flow of Events:	<table><tr><td></td><td>Player Action</td><td>System Response</td></tr><tr><td>1</td><td>Press "Settings"</td><td>Load settings screen</td></tr><tr><td>2</td><td>Change desired settings</td><td>Change game state accordingly.</td></tr></table>				Player Action	System Response	1	Press "Settings"	Load settings screen	2	Change desired settings	Change game state accordingly.
	Player Action	System Response										
1	Press "Settings"	Load settings screen										
2	Change desired settings	Change game state accordingly.										
Variations:	Changes to game will vary based on which settings are changed.											
Notes and Issues:	N/A											
Developer Notes:	N/A											

Activity Diagram:



Data Storage: As of now, our idea for persistence is a single text file for configuration and scores. We will have a class that has the sole job of parsing this file and sending data to the main class. When the application loads, it will parse this persistence file or create a default one if it doesn't exist, and update this data within the system as needed while the game is running. When a game ends, the system will modify this data if necessary. When the application is closed, it will overwrite the text file with any changes to the configuration or data values.

UI Mockups:

Title Screen

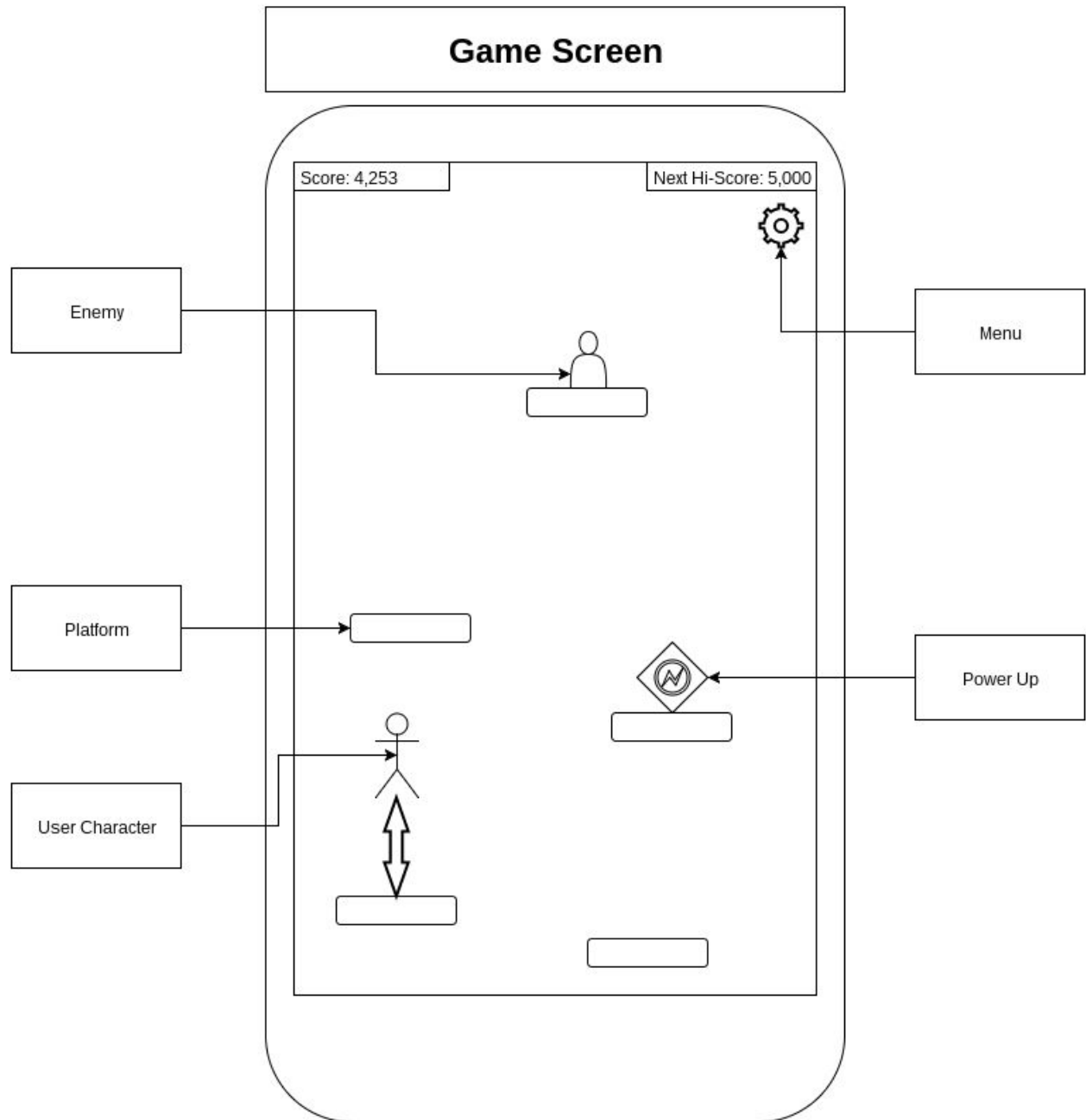
Moodle Jump

Start Game

High Scores

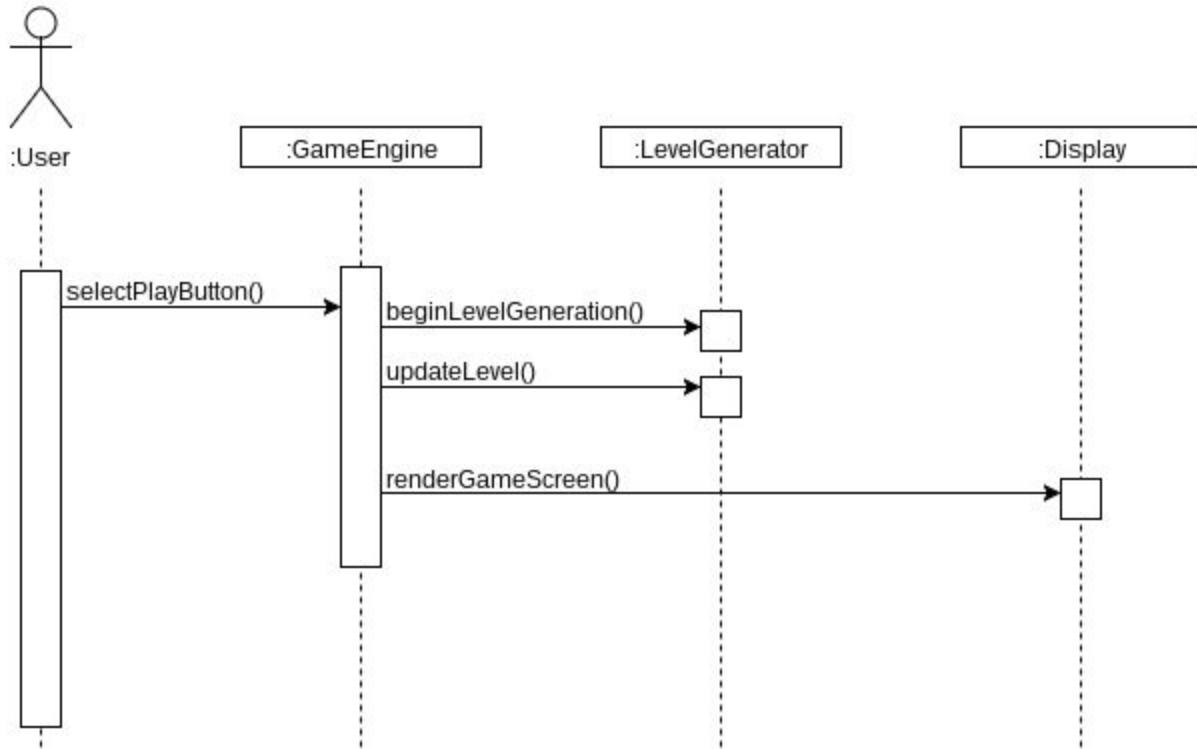
Settings

Exit

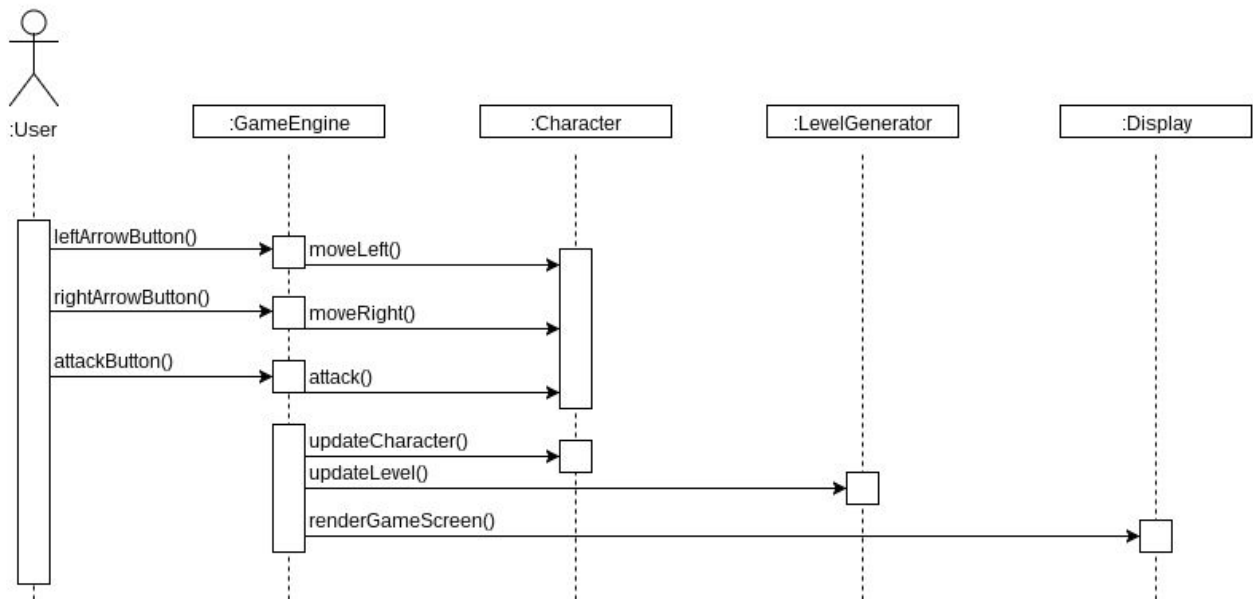


User Interaction:

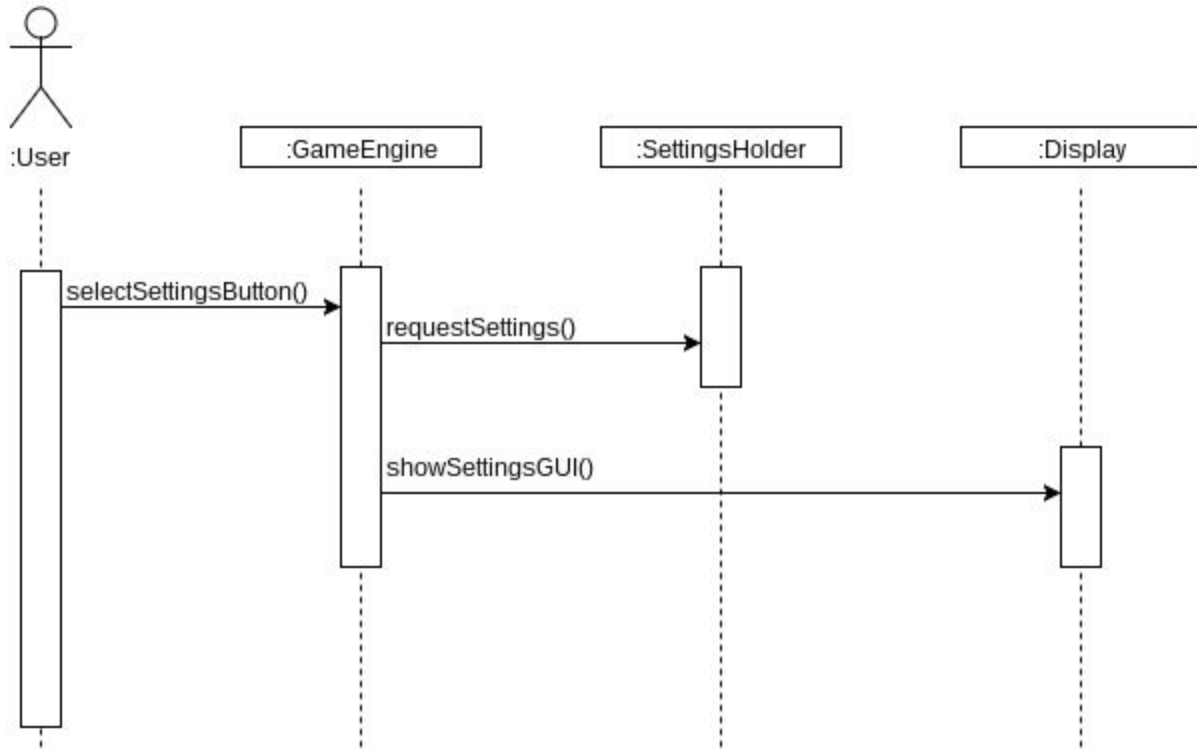
Generate New Level Sequence Diagram



Game Play Sequence Diagram



Settings Menu Sequence Diagram



Class Diagram:

