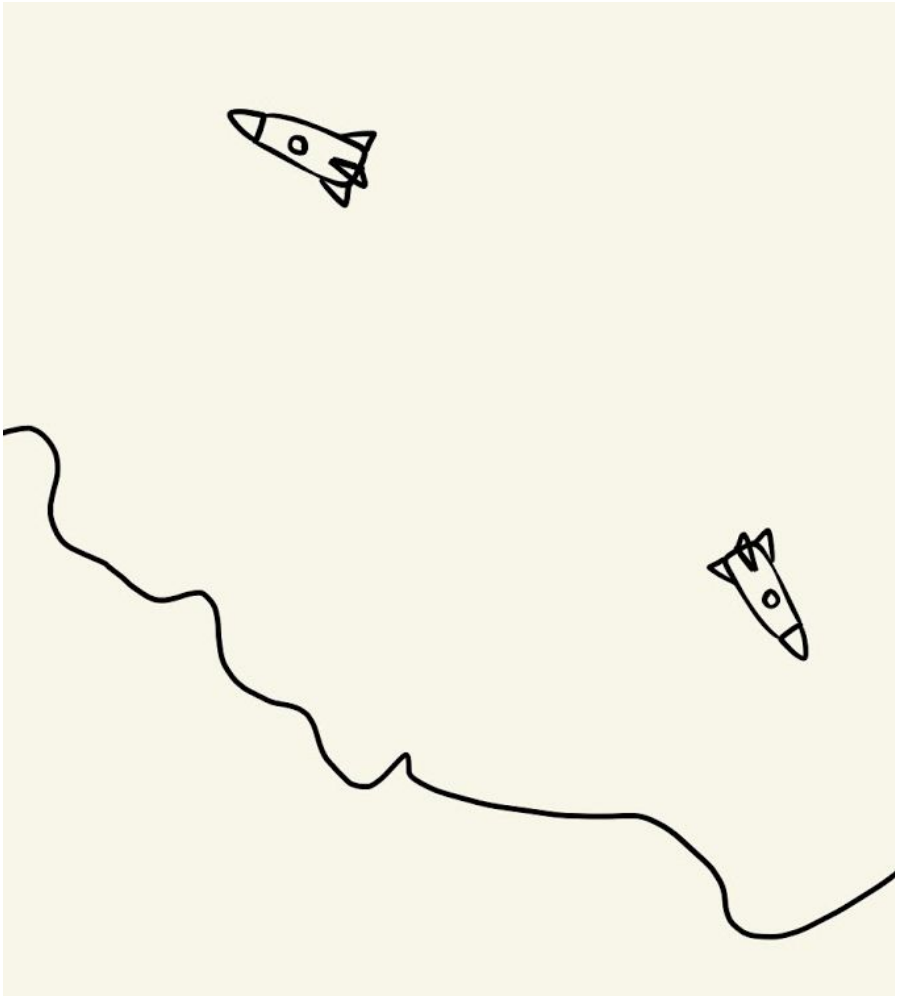
Lunar Landers
Project Report



Fall 2020

By: Group 12 - Dan Hrubec, Julian Gonzales, Jacob DiStefano, John Mistica

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

Lunar Landers is a multiplayer extension of the classic arcade game, *Lunar Lander*. The game pits multiple players against each other to see who is able to land their spacecraft on the surface first and prevent their opponents from doing so. The map will look familiar to fans of the original *Lunar Lander* as it is a similarly rocky and dangerous landing site on a moon. However new challenges may face the player during their match, such as lunar wind and worms, in addition to the threat of being blown away by their opponents. This game will challenge the skill and entertain any who choose to pick it up.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

This product is not catered to businesses, but rather individual gamers. These gamers may not know one another, and only seek to gain entertainment from the product, not profit.

2b Goals of the Project

The goal of this project is to entertain the players, and to offer players a challenge to share with their friends or others they find online. The competitive and simple to understand nature of the game are inherently entertaining for those who play. As the user continues to play with the game, they will hone their skills and become more familiar with new strategies and techniques to play. As the user's skill level increases they will find more value in the product as they best others in challenges and share the game with others.

2c Measurement

We will know that players are entertained by measuring the amount of playtime and matches played, in addition to new users. We can safely assume that players would not play a game that they do not enjoy. We can measure this by observing the amount of matches a user plays over time, and see how long it captures their interest. If a user is playing a consistent amount of matches per week over a period of months, then we can say that we have been successful in entertaining the consumer. In addition looking at sales and new user adoption will help us understand if the multiplayer and party play elements of the game motivate others to play and enjoy it.

3 The Scope of the Work

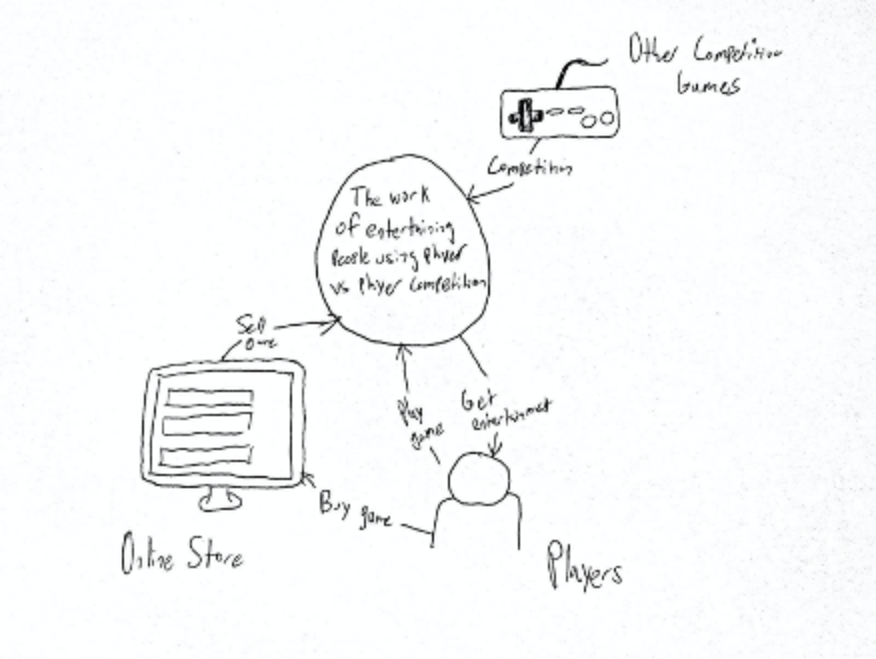
This game exists in the casual subset of the games industry, which exists for players who do not take video games terribly seriously. The work addressed by this product is to give quick simple entertainment to casual video game players. It does this by challenging the player and pitting them against other players.

3a The Current Situation

Currently there are many games that allow players to challenge one another, on many types of platforms. However many of these games have a high barrier to entry, either monetarily or due to being difficult to pick up and play.

3b The Context of the Work

Figure 1: Context of Work



3c Work Partitioning

Table 1: Work Partitioning

Event	Input	Output
Update Released	Development Work on Game	App Store is informed of update and sent new version released
New Player	New Player Logs into Game	Create Server Side Account for Player Statistics
Game Sold	App Store Sells Copy of Game	Update Internal Sales and Growth Figures
High Score	Player Gets a High Score in Game	Update Internal Leaderboard and send to other players

3d Competing Products

Games that force players to compete with one another in short challenges are not uncommon. Many recent examples of *Battle Royale* games also cater to this audience, by pitting players against each other and allowing them to create challenge by competing with one another.The appeal of this game is that the barrier to entry is very low, as the controls are simple and the objective straightforward. In addition the matches are very short in length, allowing users to quickly play a game instead of having to commit a large amount of time to sitting down and playing the game. The short time obligation and low barrier to entry are what sets this game apart.

4 The Scope of the Product

The work handled by the product would be to create an environment where players can pick up and play the game. As a result, the product would need to have a strong server established to allow client connections and create matches for players to play. Allowing for an entertaining yet simple UI to allow for players to appreciate the game and its simplicity.

4a Scenario Diagram(s)

Figure 2: Scenario Diagram 1

Figure 3: Scenario Diagram 2

4b Product Scenario List

1. People that want a casual game to play for highscores.
2. People that want to play a casual but competitive game to play with friends.
3. People that want to play for some fun.
4. People that played Lunar Lander arcade, but want to play a multiplayer version of the game.
5. People who are experienced with the game and want to show that they are the best of the best.

4c Individual Product Scenarios

People playing for fun: Most players will be attracted to the game for its easy controls and will be playing the game for fun and to relax.

People playing competitively: More experienced players will want to compete with other experienced players to be the best.

People playing alone: Some people will not want to play with others, and can play the game alone and give more of an arcade game feel.

5 Stakeholders

Potential stakeholders may include companies that are interested in cross-platform capable games or social media companies.

5a The Client

The developing organization will act as the client since the project will have the same features for every customer. However, when releasing the product for schools or large organizations there may be larger lobbies implemented.

5b The Customer

The customers for this product are expected to be a wide range of individuals such as anyone who is interested in a casual multiplayer video game. The customer base would have no association with any companies or organizations.

5c Hands-On Users of the Product

Companies that are interested in cross-platform games such as Epic Games would have hands-on-users ranging from teenagers to adults. Most of these hands-on-users would be highly experienced with cross-platform games and would be highly competent in learning how to play. Cross-platform video game companies could host tournaments for games which could allow for larger lobbies where multiple individuals compete with each other. Another organization that would be interested would be schools.

The hands-on-users would include students which would already be proficient in playing multiplayer games online with their friends. The school could host after school clubs that relate to competitive gaming (such as a competitive gaming club or after school social). The product could be used as a way to bring people together for social activities and entertainment.

5d Maintenance Users and Service Technicians

The consumer will be responsible for installing and updating the product. Similar to how most applications work, there will be an update available for users to download and install via the online store where they downloaded the product from. The user simply has to navigate to the store and update the newest version of the product. The maintenance will be solely delegated to the organization. All server updates, bugs or glitches, and patches will be the organization's responsibility.

5e Other Stakeholders

There would be some stakeholders that may not regard the product as a useful tool for their mission statement. Some stakeholders may include companies that value health and physical activity for their customers. They would see it otherwise as a distraction and setback for customers. K-6 educational institutions may also not value the product as much as they would potentially be a distraction for their young students rather than an enhancement.

5f User Participation

Users will have the ability to aid in development by participating in closed alpha and beta versions of the product where user feedback will be used to fix any shortcomings or bugs. The feedback will be essential to the final release of the product.

5g Priorities Assigned to Users

Key users would include any hands-on-users of cross-platform video game companies. These users are the most important as they comprise a large majority of the products expected user base. Secondary users would include students because while their requirements are valued key users have higher priority. Unimportant users include anyone outside the former user base.

6 Mandated Constraints

6a Solution Constraints

The product must be an online multiplayer game. This will create an engaging experience that will keep users playing. It must also be compatible with mobile devices. This is because most of the population and younger population have a mobile device and can play the game and create a larger audience

6b Implementation Environment of the Current System

The product should be compatible with ios and android specifically ios 12.4.8 or newer since that is the latest ios version that the iphone 6 is compatible with. The Iphone 6 is still popular so allowing users to download on older devices should be done. In the case for android anything above 8.0 should be fine since newer affordable android phones come with more recent versions of android installed. It should also be compatible with pc and cross platform.

6c Partner or Collaborative Applications

It does not require any partner applications

6d Off-the-Shelf Software

Does not require any off the shelf software

6e Anticipated Workplace Environment

The product is expected to be played by the younger population using the mobile application. Because this is the case the game is expected to have easy and simple game controls and possibly different options for game controls to give users a preference. The game will be used outdoors which could make darker color schemes difficult to see in the sun, the product should implement a brighter color scheme to avoid this problem.

6f Schedule Constraints

The product should be released in at most 2 years with more features being added over the following 2 years. For the complexity of the game this should be more than enough time and create profit over the 2 years of features being added. Releasing the game in this short time span will create profit early and also start creating a fanbase early.

6g Budget Constraints

The product should not exceed more than 100 mb on the mobile side since users with lower quality devices have smaller storage space. This will prevent the user from wanting to delete the application because of storage space. On the PC side the product can be much bigger around one gigabyte max. This is including features which means the game could start with much less storage space and work itself up to the 100mb cap. The budget should be 300k based on other games with the same complexity and being online. For example hearthstone being a large online multiplayer game had a cost of around 300k.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Cross-Platform: A term used to describe how a software can be used across different types of computers.

Cross-Platform Gaming: A term used to describe players using different video game hardware while playing with each other at the same time. For example being able to play in the same online session with players using Xbox, PS4, or a Nintendo Switch.

Cross-Platform Gaming Companies: Used to describe companies that hold an interest in investing in games that can be played on multiple computers/consoles simultaneously. Such examples of companies may include Epic Games or Blizzard.

7b UML and Other Notation Used in This Document

Diagram in 3b: Arrows indicate an action or relationship between objects.

7c Data Dictionary for Any Included Models

N/A

8 Relevant Facts and Assumptions

8a Facts

Feedback and data collected from the target users and the strategies players have to give them the best possible advantages will be under review and to understand how players are approaching the game. For example, if the damage dealt by the engine is too high, and can prevent players from competing, there could be scaling on the damage dealt and will be adjusted accordingly to maintain fairness while still keeping the integrity of the game. With this in mind, data and feedback will also adjust the difficulty of the maps, if there are inconsistencies within the difficulties, like if on the hardest difficulty, players always land on the safest area, we could make the safer areas less rewarding, while increasing the reward for landing on the more difficult terrain

8b Assumptions

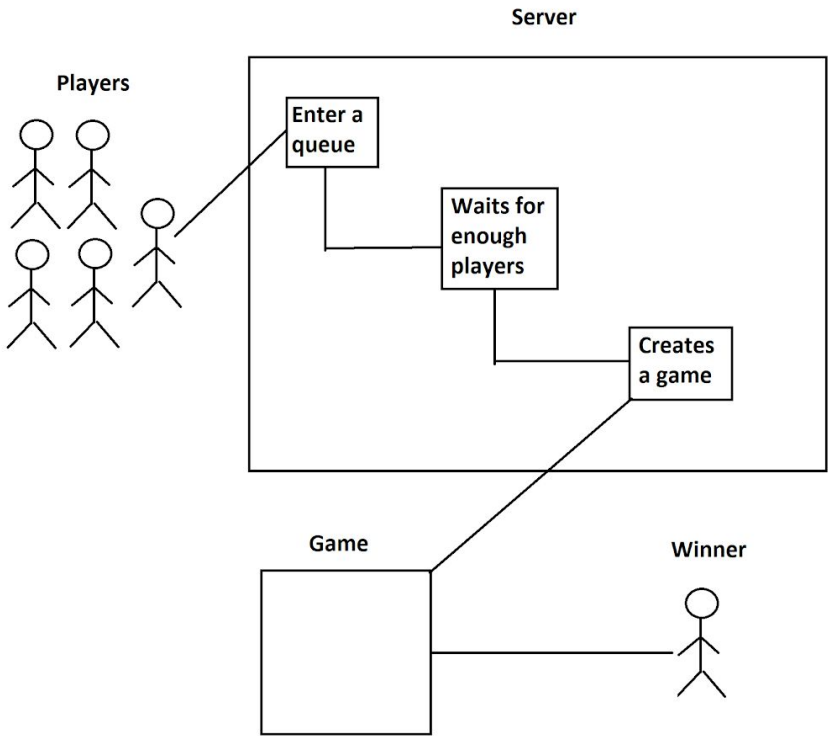
An assumption will be made that players can play with each other regardless of the platform they play on. This will encourage the multiplayer aspect of the game, while not burdening the consumer to have to restrict themselves to one platform to play with others. With this in consideration, the server hosting the multiplayer connections will need to allow for clients to connect on all platform

II Requirements

9 Product Use Cases

9a Use Case Diagrams

Figure 4: Scenario Diagram



9b Product Use Case List

Unranked Play: If the players decide to join an unranked lobby, then it will just wait for enough players before creating a game.

Ranked Play: If a player chooses to enter into a ranked game lobby, the server will have to group people within a certain range based on skill so the game remains competitive, and that players are evenly matched.

9c Individual Product Use Cases

Table 2: Product Use Cases

Use case ID: U1 Name:Game Creation Pre-conditions: N/A Post-conditions: N/A Initiated by: Players Triggering Event: Joining Queue Additional actors: Other Players
Sequence of Events 1. Players choose either a casual or competitive game to play a. Players have a timer for how long they have been in the queue, while the server waits for more players to join. 2. The Server has enough players to create a game/lobby, connecting all the players. a. If the queue is a competitive game, the server will have to allow only players of the same skill level to play together. 3. The Players are then brought into the game lobby where they can now play the game against each other 4. Players compete for the best score, the winner is declared and the players are brought back to the client side screen where they can choose to play another game or quit.
Alternatives: As stated in the sequence of events, if players choose to play a competitive game, the server will have to wait for enough players around their skill level to create a lobby, in order to keep games competitive and to maintain the integrity of the game. If the player simply chooses to be in a casual lobby, it will just wait for enough players before creating the game.
Exceptions: Exceptions would be dependent on the server's status. If the server is too full, or if errors happen during the game creation, the players will be booted off the server and brought back to the client menu with an appropriate error message to describe what happened.

10 Functional Requirements

ID#1 - Map Creation

Description: The system must be able to generate completely different maps, including all obstacles.

Rationale: Maps need to be different in order to maintain the enjoyment of the game. If there were no different map generations, players will get bored of practicing landing on the same map multiple times. Allowing maps to be different will keep the game fresh, new and more enjoyable.

Fit Criterion: Check to make sure that the map generation produces different maps each time a map was created, so the overall layout of the landing is different in each scenario.

Acceptance Tests: Map Generator

ID#2 - Player Queue

Description: The system must be able to handle at most 8 players to join a queue, and to be matchmade together in the same game.

Rationale: The game is intended to allow for multiple players to be able to play together. If players are not able to play with each other, the premise of the game would be non-existent leaving little reason to play.

Fit Criterion: The system needs to remain fully functional when the players join, making sure the server can handle a multitude of requests.

Acceptance Tests: Queues

ID#3 - Winner and scoring calculations

Description: The system must be able to analyze each player's performance giving a score, with possible tiebreaker scenarios.

Rationale: If the scoring system was not working it would be very difficult to determine which player won, or did the best. Having the scoring system be known to the users, allows them to know how to play and how to maximize their score.

Fit Criterion: The system needs to be able to produce one winner each game to prevent any issues when playing competitively.

Acceptance Tests: Scoring and Win Conditions

11 Data Requirements

ID#4 - Player Statistics

Description: The system must be able to store statistics about each player, like the amount of games played, win rate percentage, average score.

Rationale: A way of showing players and giving feedback on the amount of time played in the game. Shows players a range of statistics so players can choose which aspects of the game they want to improve on.

Fit Criterion: Statistics displayed on the client screen through any menu where it is easily accessible to players in between games.

Acceptance Tests: Player Statistics

12 Performance Requirements

12a Speed and Latency Requirements

ID#5 - Latency

Description: The system must allow for the player to continuously be able to play the game.

Rationale: The user's must have their inputs work at a decent latency so as to not take away from the overall gameplay. If the latency is too high, players can get frustrated and will lose interest in the game.

Fit Criterion: When testing the game, if the controls are smooth without interruption then the requirement is met.

Acceptance Tests: Latency Threshold.

12b Precision or Accuracy Requirements

ID#6 - Precision

Description: The collision detection between the players ship and other objects like other players and the ground must have well defined boundaries.

Rationale: Having well defined collision boundaries allow the game to seem very fair, if

the collisions were not tightly bound to their respective object, it would be frustrating to players as they would expect the boundaries to be as close to the object as possible.

Fit Criterion: Accurate grid positioning and outlining on the collision detection will satisfy this requirement.

Acceptance Tests: Precision

12c Capacity Requirements

ID#7 - Capacity

Description: The system must be able to keep track and store each of the player's ships, the map, and their respective positions.

Rationale: In order for the players to know where they are in terms of the map, the need to know their position with respect to the map. It helps the system with checking collisions, and helps the players know where they are.

Fit Criterion: The system will allow for a certain capacity for each game that is created and played. All games must be within this threshold.

Acceptance Tests: Capacity

13 Dependability Requirements

13a Reliability Requirements

ID#8 - Reliability

Description: The system should not fail frequently, otherwise the game would be impossible to play or otherwise have poor performance.

Rationale: Users would not be able to play a game that shuts down and has frequent interruptions.

Fit Criterion: The system must run through the entire gameplay of the game without any shutdowns or interruptions. It must also be able to load and display data accurately without delay and be able to take in user input without lag or delay as well, then the requirement is satisfied.

Acceptance Tests: Reliability

13b Availability Requirements

ID#9 - Availability

Description: System must be up and available at all times unless a scheduled maintenance is required.

Rationale: Having the system up at all times would be a requirement for users to enjoy the game at anytime meeting the demands of the user.

Fit Criterion: System must be up at all possible times and scheduled maintenance can only take place when during times where there is lower user activity. System must go down only at these times for this requirement to be satisfied.

Acceptance Tests: Availability

13c Robustness or Fault-Tolerance Requirements

ID#10 - Robustness

Description: The system must be able to handle network outages and be able to run offline.

Rationale: Even after an error or fault has occurred, the game must maintain functionality.

Fit Criterion: If there is a situation where a fault or error arises, the system must be able to handle it in a way that data is not corrupted or lost by saving the most recent data before the point of error, then the requirement is satisfied.

Acceptance Tests: Reliability

13d Safety-Critical Requirements

ID#11 - Safety

Description: The system must provide a safe environment for anyone regardless of age or demographic.

Rationale: The game must be family-friendly and block any toxic, dangerous, or predatory behavior between users in order for the game to be available and compatible to a wide variety of people.

Fit Criterion: The system must block any toxic, dangerous, or predatory behavior between users and must be secure of any malicious attacks or leaks, if these hold true then the requirement will be met.

Acceptance Tests: Safety

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

ID#12 - Maintenance

Description: The system must be frequently maintained by developers during scheduled maintenance to which any active users will be notified of downtime.

Rationale: The system must be regularly maintained in order for the game to run properly and functionally.

Fit Criterion: The system must undergo routine maintenance and be able to handle any unexpected faults or errors in order for the game to maintain functionality. If these are met than the requirement is satisfied.

Acceptance Tests: Availability

14b Supportability Requirements

ID#13 - Supportability

Description: The system must provide support to its users and be able to give feedback in a timely manner in case of any user dissatisfaction.

Rationale: Any feedback will be beneficial to the game as the users can report anything they were satisfied/dissatisfied with.

Fit Criterion: Help and feedback features should be available at all times and provide/assist in meaningful changes to the game.
Acceptance Tests: Player Statistics

14c Adaptability Requirements

ID#14 - Adaptability

Description: System must support cross-platform gameplay between different consoles/pc's.

Rationale: Having users be able to the game on any device allows for flexibility and allows for the game to be available to a wider audience.

Fit Criterion: Cross-platform gameplay must be available at all times besides during scheduled maintenance. If this is fulfilled then the requirement is met.

Acceptance Tests: Reliability, Latency

14d Scalability or Extensibility Requirements

ID#15 - Scalability

Description: System must be designed to scale with new designs such as maps or features.

Rationale: This allows for the game to have frequent updates and a fresh feeling so that users can enjoy non-stagnant gameplay.

Fit Criterion: System must be able to be scaled easily for any new features such as new maps or objects in order for the scalability requirement to be met.

Acceptance Tests: Scalability, Maintenance, Robustness

14e Longevity Requirements

ID#16 - Longevity

Description: The system must be expected to last at least 10 years

Rationale: This allows more users to participate and also allows users the ability to progress within the game for a long period of time.

Fit Criterion: Scalable software is a must since longevity depends upon the scalability and adaptability of the software. If this is fulfilled the requirement is met.

Acceptance Tests: Scalability, Adaptability

15 Security Requirements

15a Access Requirements

ID#17 - Access Requirements

Description: The users data should be protected

Rationale: Since user data like payment information and emails is sensitive these should be protected and possibly encrypted for safe usage

Fit Criterion: This should be met by encrypting data like passwords and payment data and only allowing certain developers access to the user account data

Acceptance Tests: Security, Storage

15b Integrity Requirements

ID#18 - Integrity

Description: Protection against overload attacks

Rationale: Since some users may have the motivation to shutdown the server due to overloading and other malicious attempts the server should protect against these types attacks

Fit Criterion: Once the server is able to notice these attacks and prevent them without causing any disruption of the game play then this criterion has been met

Acceptance Tests: Security

15c Privacy Requirements

ID#19 - Privacy

Description: Storing payment information securely

Rationale: Since payment data for in app purchases may be handled the storage of this data should be secure to give users confidence in purchases and not cause any legal problems

Fit Criterion: Once access to payment information is secure by encryption or some form of secure storage and the access of this data has been restricted then this criterion has been met.

Acceptance Tests: Security, Legal

15d Audit Requirements

ID#20 - Audit

Description: In app purchase transactions should be recorded

Rationale: This is needed since a user may claim whether they have or have not made a purchase and could cause legal problems if not proved if they did or not

Fit Criterion: This is met once a secure database is created storing these records and can easily be looked up securely if needed.

Acceptance Tests: Security, Legal

15e Immunity Requirements

ID#21 - Immunity

Description: Protection from possible hackers

Rationale: Some hackers will possibly want to modify gameplay and customize how the game is run to their advantage and this should be prevented to give users a fair gameplay experience.

Fit Criterion: This is met once attacks have been made by the developers and successfully stopped by the application. Will also need to be updated as the game releases if any vulnerabilities are found

Acceptance Tests: Security

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID#22 - Ease of Use

Description: Because of the casual use of the game the ease of usability should be simple

Rationale: This will cause users to easily pick up the game and possibly become hooked on the game. Also the controls and UI should not be the frustration of the game but the competition between the players.

Fit Criterion: For this criterion to be met testing with users should be required focusing on the teen and young adult ages. 90% or more should say that the game has simple and easy controls and also simple gameplay

Acceptance Tests: Ease of Use

16b Personalization and Internationalization Requirements

ID#23 - Personalization and Internationalization

Description: Ability to change language

Rationale: The user should be able to change the language accurately

Fit Criterion: This is met once user tests come back that the translations were accepted for at least 90 percent of the users that selected the language change.

Acceptance Tests: Style

16c Learning Requirements

ID#24 - Learning Curve

Description: The product shall be quick to learn by means of a small tutorial

Rationale: This is needed to show users how the game is played and what the controls are this will let the user easily get grasp of the game and possibly keep playing it

Fit Criterion: This will be met once a tutorial is made and tested with users. Results in user testing should come back positive with 90 percent of users stating the learning curve was not difficult.

Acceptance Tests: Style, Ease of Use

16d Understandability and Politeness Requirements

ID#25 - Understandability

Description: Minimalistic menu and goals during gameplay and tutorial

Rationale: This is needed to prevent confusion in the operation of the game and learning unneeded words and prompts that would possibly confuse the user

Fit Criterion: This is met during user testing when tests come back with 90 percent of users stating that there was no confusion during the menu and the gameplay

Acceptance Tests: Ethical

16e Accessibility Requirements

ID#26 - Accessibility

Description: The colors of the game must not result in confusion with color blind users

Rationale: Many of the users may be colorblind and tending to these users by simple choice of colors could result in a higher user base. Choosing colors that will give the user less confusion during gameplay will resolve this.

Fit Criterion: This shall be met in user testing with users stating they are color blind. If 90 percent of these users state that the gameplay was not confusing or distracting then this criterion is met.

Acceptance Tests: Ethical

16f User Documentation Requirements

ID#27 - Documentation

Description: Controls menu will be provided if users forget

Rationale: This is needed just in case users don't remember how to play the game and need a reference.

Fit Criterion: This is met once the menu is successfully created

Acceptance Tests: Ease of Use

17 Look and Feel Requirements

17a Appearance Requirements

ID#28 - Appearance

Description: The appearance of the product shall appeal to young adults and teens using vibrant colors but also meet the needs of the colorblind described above
Rationale: This is needed since the appearance will get the user engaged especially during gameplay since the game is retro it should give it that vibrant retro look

Fit Criterion: This will be met if users the targeted audience decides to install and play the game because of the looks of the game

Acceptance Tests: Style

17b Style Requirements

ID#29 - Style

Description: The game should appear fully finished and robust

Rationale: A product that does not appear to be fully finished does not seem trusted to most users for this case the product should have finished UI and gameplay.

Fit Criterion: This is met once the user agrees that the application feels trustworthy during gameplay with 90 percent of users trusting the product

Acceptance Tests: Style

18 Operational and Environmental Requirements

18a Expected Physical Environment

ID#30 - Physical Environment

Description: The user will possibly be using the application during moving environments

Rationale: The moving environments like car rides, bus rides, etc. could possibly introduce difficulty with controls. This should be taken into consideration when

designing controls where movement could affect them like tilting the device to move. In this case multiple controls should be implemented like on screen buttons and tilt to move. Tilt to move being easier and simpler while buttons being the most stable.

Fit Criterion: This is met once the application implements two forms of movement successfully and tests by users to be easy to use in above ease of use tests

Acceptance Tests: Availability, Reliability

18b Requirements for Interfacing with Adjacent Systems

ID#31 - Interfacing with adjacent systems

Description: The applications should be able to interface with google play game services and apple game center services

Rationale: This provides a leaderboard system that is robust and viewable by users outside of the game. Also provides the ability to add friends and invite friends from contacts. The data content is the scores for each user the interface is the google and apple services. The leaderboards should be updated at each game

Fit Criterion: Once the application has the ability to upload scores to the google and apple game services then this criterion has been met

Acceptance Tests: Production

18c Productization Requirements

ID#32 - Production Requirements

Description: The product shall require the installation on the app store and google play store

Rationale: These are the most popular app stores and should be able to download and install them from them both. This would open the application to both markets creating a bigger user base.

Fit Criterion: Once the application is successfully installable from both the app store and the google play store then this criterion is met.

Acceptance Tests: Production

18d Release Requirements

ID#33 - Release Requirements

Description: The releases and updates should be updated every month. Fixing bugs and features.

Rationale: Frequent updates and additions will keep the game fresh and fun for the users. This will keep the users engaged longer and increase profits further.

Fit Criterion: The type of maintenance that will be taken into consideration is any bug reports from users and anything the developers notice. New features will be considered from users and developers. Users being of priority.

Acceptance Tests: Maintenance

19 Cultural and Political Requirements

19a Cultural Requirements

ID#34 - Culture

Description: The product will not offend any religious, ethnic, or cultural groups.

Rationale: The product should be catered and compatible with everyone.

Fit Criterion: Once testers approve that the game is compatible everywhere then the requirement will be met.

Acceptance Tests: Safety

19b Political Requirements

ID#35 - Political

Description: The product shall be made available only through specific app stores.

Rationale: Licensing and agreements will be made with companies and must be kept.

Fit Criterion: If the product is found only to be available through specified app stores then the requirement will be met.

Acceptance Tests: Style, Ethical

20 Legal Requirements

20a Compliance Requirements

ID# 36 - Compliance

Description: The product must comply with the laws and views of the local government

Rationale: Product must be compatible with views of local government in order for the product to be properly released.

Fit Criterion: Requirement will be met when the local government approves the distribution of the product

Acceptance Tests: Safety, Security, Legal

20b Standards Requirements

ID#37 - Standards

Description: The product must be held to the age rating that it is given.

Rationale: Holding to the the age rating allows for more users to be able to play the game and mitigates the risks for anyone playing the game.

Fit Criterion: Requirement will be met when the game is tested to fit the age rating that it is given.

Acceptance Tests: Safety, Ethical, Legal

21 Requirements Acceptance Tests

21a Requirements – Test Correspondence Summary

Table 3: Requirement Tests 1-19

	Requirements 1 - 19																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	X																		
2		X																	
3			X																
4			X																
5				X									X						
6					X									X					
7						X													
8							X												

9									X			X							
10								X		X				X					
11																	X	X	X
12											X								
13																	X		
14															X				
15															X				
16															X	X			
17																X			
18																			X
19																			
20																			
21																			
22																			

Table 4: Requirement Tests 20-37

	Requirements 20 - 37																	
	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**
1																		
2																		
3																		
4																		
5																		
6																		
7																		
8																		
9											X							

10											**X**							
11	**X**	**X**														**X**	
12															**X**		**X**	**X**
13																		
14														**X**				
15																		
16																		
17																		
18	**X**																**X**	**X**
19			**X**		**X**			**X**										
20				**X**	**X**				**X**	**X**						**X**		
21						**X**	**X**									**X**		**X**
22												**X**	**X**					

21b Acceptance Test Descriptions

1. **Map Generator:** Several matches should be played and stages should be compared in order to confirm that levels are distinct and offer variety.
2. **Queues:** A party of 8 people should attempt to join a match together in order to play in the same match. In addition 8 unrelated parties should attempt to join a game together.
3. **Scoring:** Several matches should be played with a variety of outcomes and player actions, the players actions should be noted and compared to the end match score screen.
4. **Win Conditions:** The winner of each match should be the player who lands and has the highest score. Several matches should be played to confirm this against manual observation.
5. **Player Statistics:** After playing several matches the player's actions should be manually recorded. The Statistics screen should then be checked to confirm these observations are correctly registered.
6. **Latency Threshold:** The time between button input and screen action should be noted. This should be measured for both the player and opponents to determine network latency in addition to local latency.
7. **Precision:** The collision between players and objects should be tested at different speeds, and directions. This would be done by simply ramming a player against multiple objects and confirming that on screen collisions match the game's behavior.

8. **Capacity:** A match with 8 players should be played, with the maximum allowed stage hazards on. The game's memory consumption would then be tested to ensure that the capacity has not been exceeded.
9. **Availability:** A user should be able to log into the system at any time. The user should attempt to join a match at peak times like 5pm and at off peak times like 3am. In addition the serves should be accessible wherever internet connections exist.
10. **Reliability:** Whenever a network outage has been reported, check the user's saved data after the system comes back online to confirm nothing has been damaged.
11. **Security:** A user should not be able to access the information of any other user, this can be tested by confirming that the storage from the game only contains the player's information.
12. **Safety:** A user should not be able to threaten or otherwise assault other players. Users should attempt to send expletives or other offensive communication through the chat, appropriate action should be taken against that user.
13. **Storage:** The user should be able to play many games over time and not exceed the allocated memory of the game. This can be tested by playing over a long period of time and examining the storage utilized.
14. **Maintenance:** When an update is released, a user should be able to be able to see that a new update is out, and should be required to update the game before playing again. This behavior should be tested across multiple platforms
15. **Robustness**: When a new map or other feature has been added, a user should not notice any changes in other game behavior or any large difference in the user experience.
16. **Scalability:** The user experience during peak hours should be similar to that on off-peak hours. The number of players and installed games should not affect the user experience.
17. **Adaptability:** The user experience should not differ from device to device. This can be tested by running the game on multiple platforms and confirming similar behavior on each.
18. **Legal:** This game should abide by all legal restrictions placed upon it, and inform the user of their rights and proper usage of the game. This can be tested by confirming the presentation of a EULA agreement upon install, and conferring with lawyers to confirm that the game meets other legal regulations.
19. **Ease of Use:** The game's UI and controls should be relatively easy to use and pick up. This should be tested by showing the UI to a person with no previous interaction with the game, and confirming their ability to navigate the game and operate its controls.
20. **Style:** The game's presentation should be inviting and friendly. The user should be able to tell that it is a clearly lighthearted game, in addition they should be able to change the language to suit their needs.
21. **Ethical:** This game should not offend anybody or promote any unethical beliefs. This can be tested by showing the game to a diverse focus group, and confirming that the game is acceptable to all relevant parties.
22. **Production:** This product should be available on select storefronts on the relevant platforms. This can be confirmed by a user having little difficulty finding and installing the game on multiple platforms from the respective storefront.

III Design

22 Design Goals

The design goals focus on a simple and easy to use user interface to allow players to quickly create and join games. Another major design goal is to have a unique map generation via the use of a seeding system, so that no maps are identical to others, so each map is a unique and new experience.

23 Current System Design

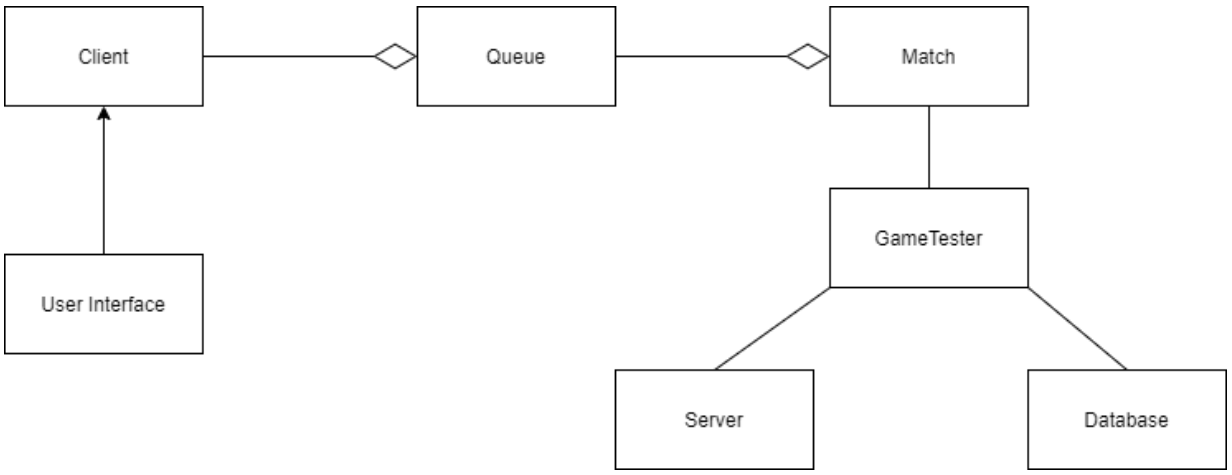
No current system has been implemented at this time.

24 Proposed System Design

24a Initial System Analysis and Class Identification

Below is a simplified UML Diagram depicting the interactions to allow the game to function.

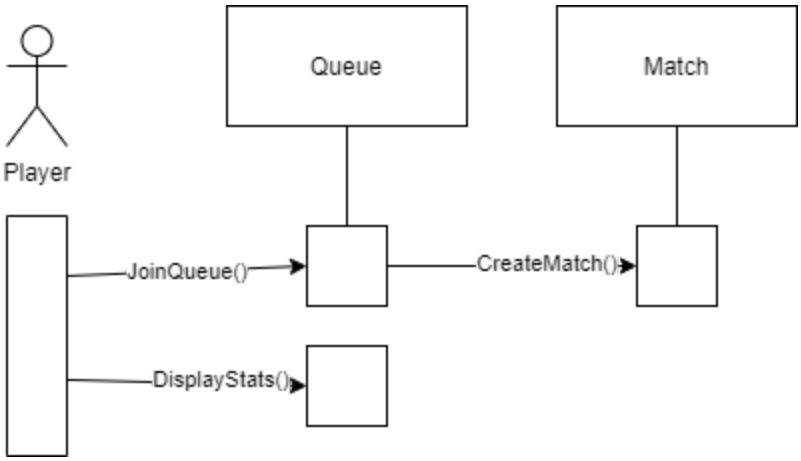
Figure 5: Initial Class Identification



24b Dynamic Modelling of Use-Cases

Below is a use-case diagram denoting the casual matchmaking queue.

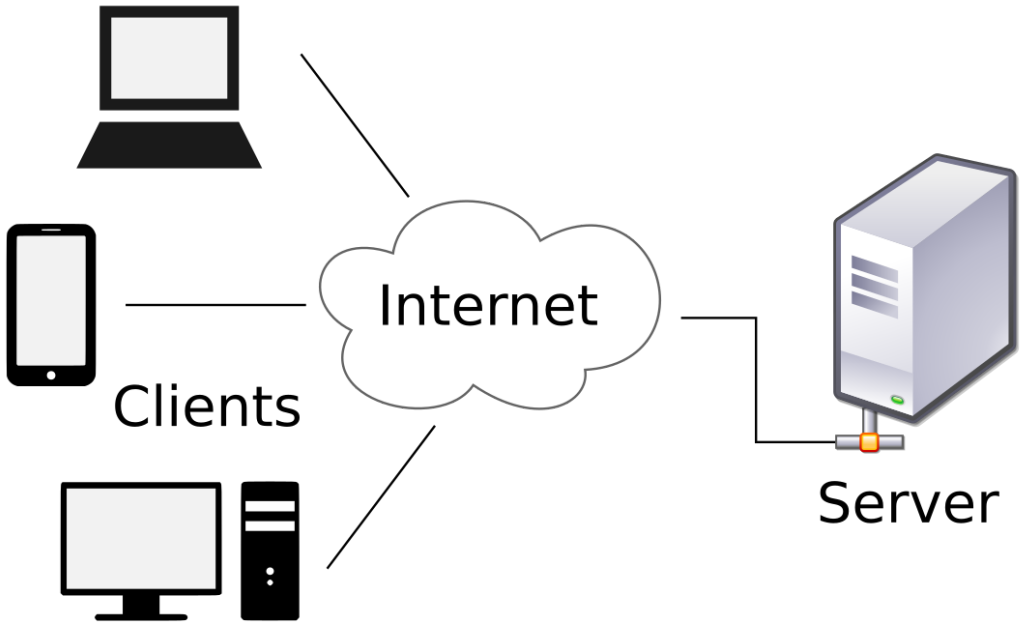
Figure 6: Matchmaking Use Case Diagram



24c Proposed System Architecture

The proposed architecture is based on the common client-server architecture. This will allow for multiple clients or players to join into a match that the server will host.

Figure 7: Client Server Architecture



24d Initial Subsystem Decomposition

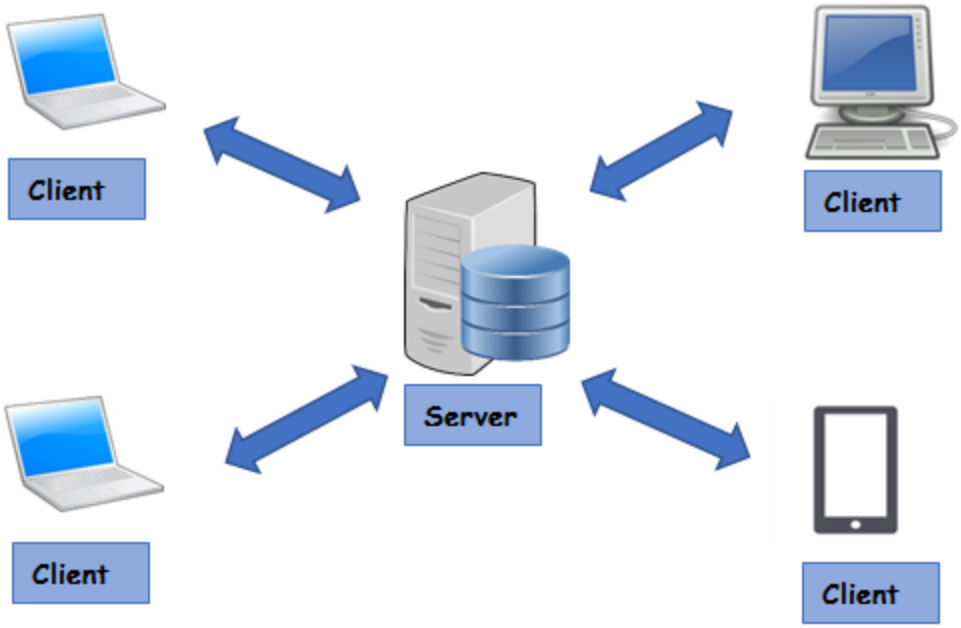
Subsystem components:

1. Client: This component is responsible for the user interface as well as utilizing resources to send players into the queue.
2. Server: This component connects the players from the queue into an actual match. The server has an open port in which the players connect to when finding a match.
3. Game/Match: This component is the actual game, which has a seed and generates a new map accordingly. Players all connect and interact with each other on this match and map.
4. Queue: This component pertains to the server, and listens in on a port. Depending on the type of match being played, will either wait for enough players and send them into a game, or wait for enough players within the same skill rating to send them into a match.
5. Statistics: This component displays all of the players statistics both in game and out of game. Statistics in game pertain to data about the map, for example the players position on the map, and out of game displays total games played, win percentage, and rating.

25 Additional Design Considerations

25a Hardware / Software Mapping

Figure 8: Hardware Software Mapping



25b Persistent Data Management

Some persistent data to be stored would include items such as a players username, highscores, friends, and any possible cosmetic personalization options.

25c Access Control and Security

Any players would have an account that would be associated with their data such as their username, highscores, etc… Accounts would need to be secured and protected at all times. So there will probably be an additional Account class that would need to be added.

25d Global Software Control

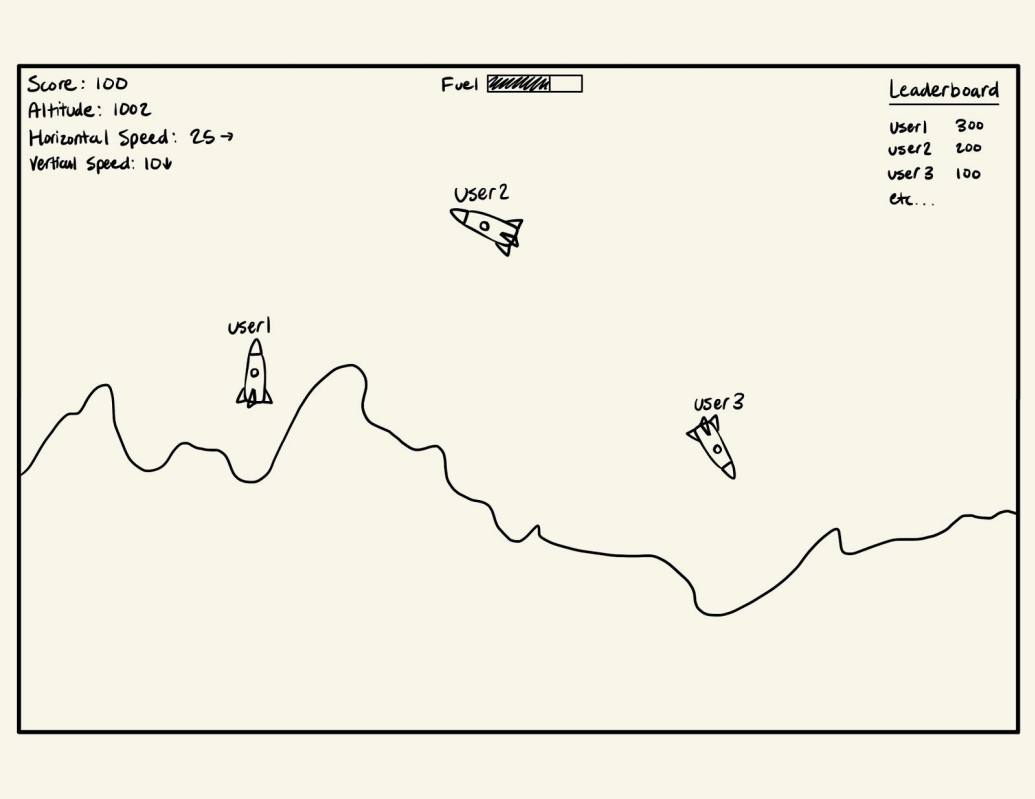
N/A

25e Boundary Conditions

Some boundary conditions would include having a maximum capacity of players in a lobby and making sure that any abnormal stress on the servers that might cause a shutdown are able to be handled. Any startups or shutdowns necessary to perform maintenance would also be considered.

25f User Interface

Figure 9: User Interface Sketch

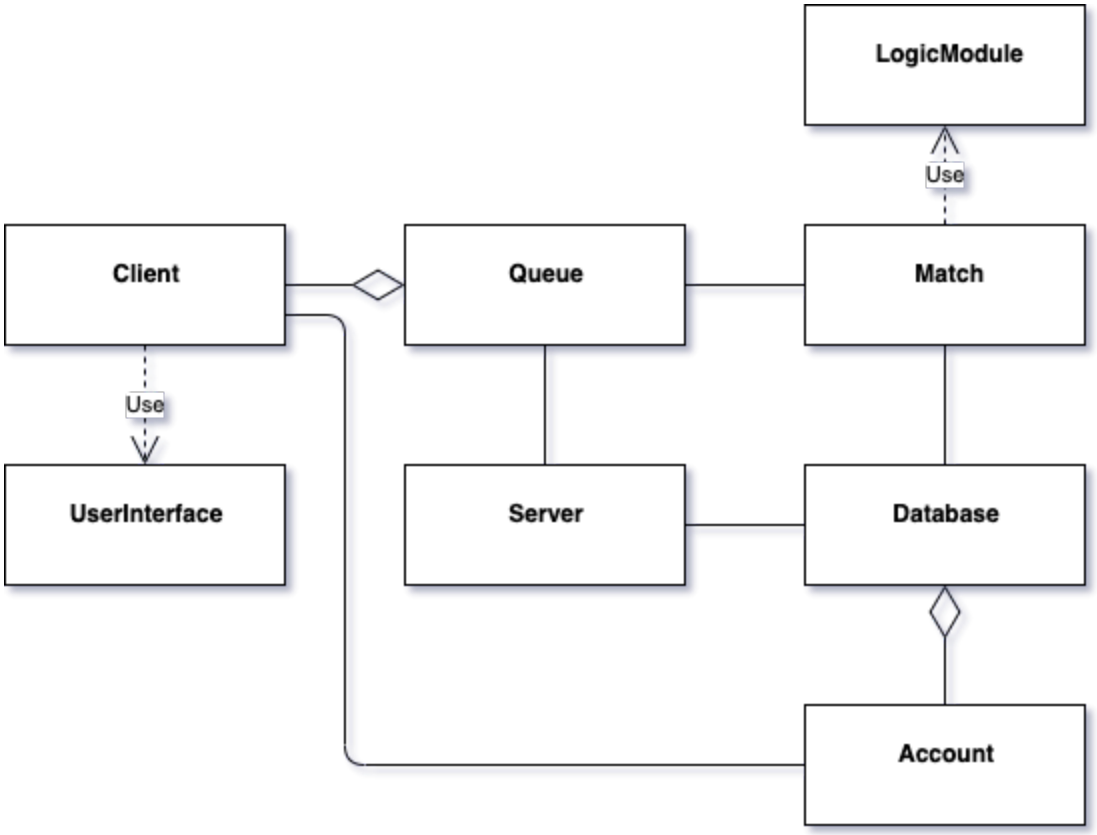


25g Application of Design Patterns

N/A

26 Final System Design

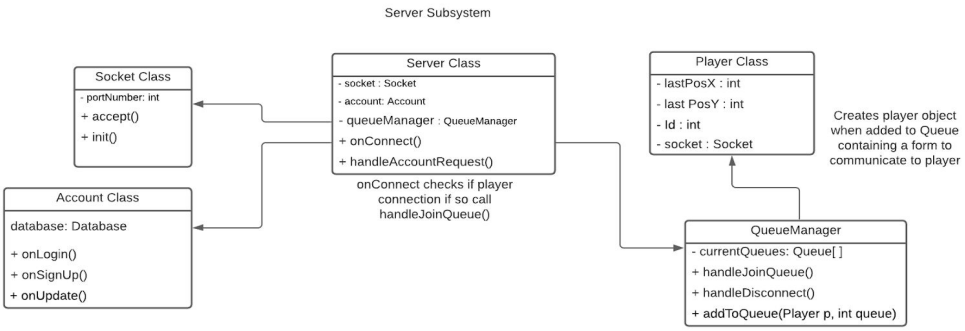
Figure 10: Final System Design UML



27 Object Design

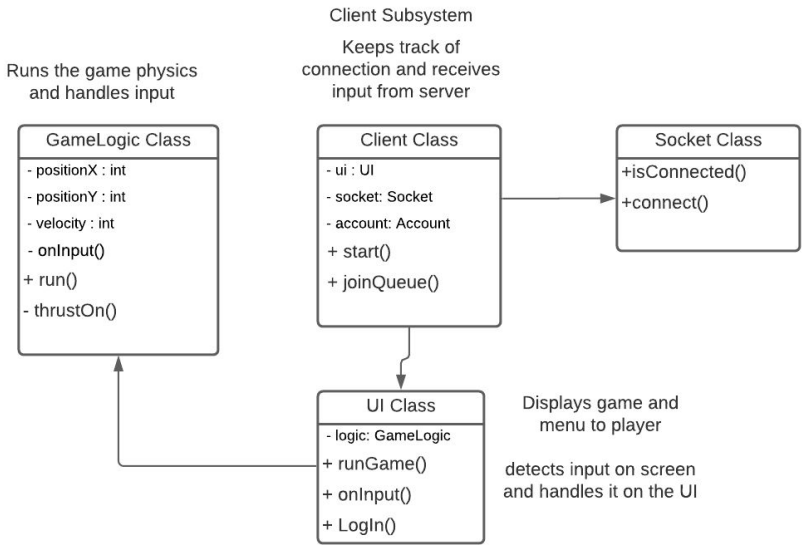
27a Server Subsystem

Figure 11: Server Subsystem



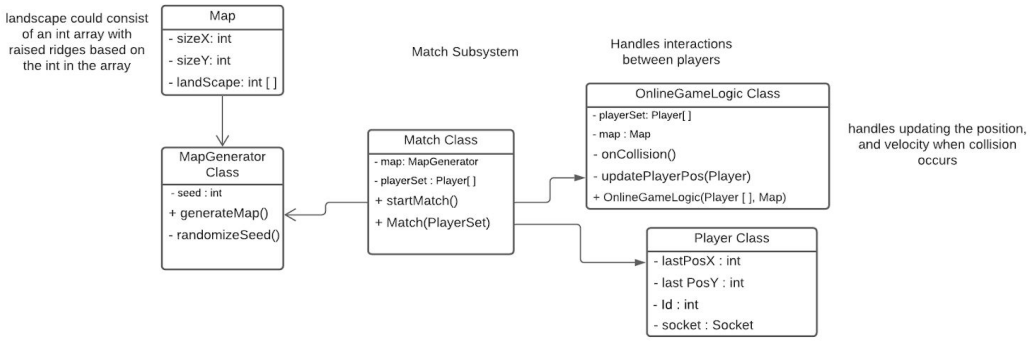
27b Client Subsystem

Figure 12: Client Subsystem



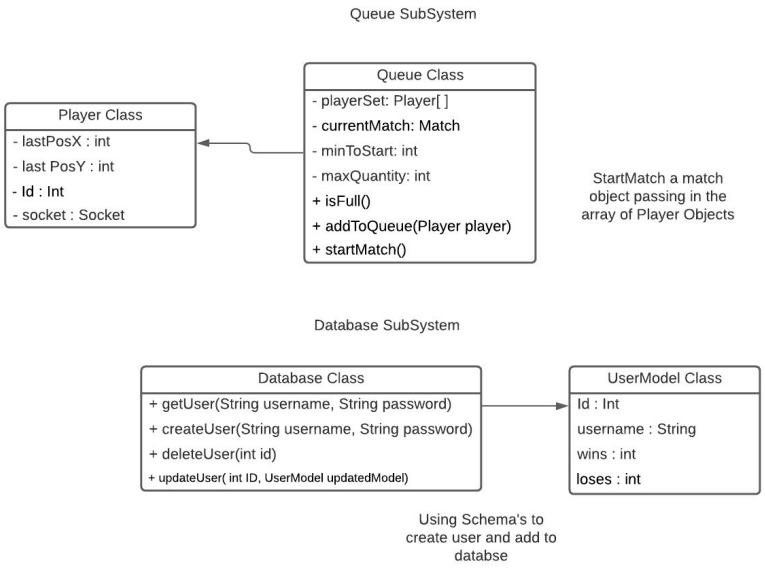
27c Match Subsystem

Figure 13: Match Subsystem



27d Queue Subsystem

Figure 14: Queue Subsystem



IV Project Issues

28 Open Issues

We have yet to determine whether queueing and matchmaking should be split into ranked pools, or all players competing with one another in one large pool. Each of these methods have benefits and drawbacks, such as the splintering of the community for ranked pools and the dominance of veteran players in an unranked pool. The matchmaking algorithm must be deliberated further to determine the course of action.

The inclusion of multiple game modes, such as a classic mode with no hazards or weather conditions, has not been determined yet. Currently we believe there should only be one game mode, yet we would not want to force players to play in a way which is too confusing or not engaging for them.

29 Off-the-Shelf Solutions

We have considered several third party game engines to build our game in, rather than developing an in house engine. This would allow for faster development and more stability.

29a Ready-Made Products

In regards to the game engine, there exist many third party engines we could easily license for our game. The current favored engine is Unity, for its expansive libraries (see 29b) and long history of stability and continued support. Other possible third party engines considered could be the Unreal engine, or possibly GameMaker.

29b Reusable Components

The Unity platform has many libraries available that can be leveraged to make development of this game much faster. One such library is the Unity Physics engine, which will make the behavior and movement of landers, winds and hazards much more adjustable. Another such library that could be leveraged is the plethora of database access libraries included with Unity, which would make the retrieval of player statistics much simpler.

29c Products That Can Be Copied

Since our game is intended to evoke the feeling and essence of the arcade classic *Lunar Lander,* we could study the source material for more insight into gameplay mechanics. Namely we could either purchase an original arcade cabinet, attempt to source a ROM of the game, or purchase one of the many home console ports of the

game. We could study the game to get an idea of how the physics of the game should behave, both for the gravity of the moon, weight of the lunar lander and the thrust supplied by the thrusters. In addition we could study the source material for inspiration for the art style and map generation for our iteration on the original playable area.

30 New Problems

30a Effects on the Current Environment

Depending on the popularity and adoption of this game, especially in mobile spaces where hot games are quickly popularized and abandoned, this game could disrupt the landscape of short term simple games. Some current (December 2020) games that this could possibly absorb the player base of may include; *Among Us, Candy Crush, Fortnite, Fall Guys*.

30b Effects on the Installed Systems

The game would have minimal impacts on installed systems for the PC and home console platforms. However on mobile devices where resources are more limited, this game may have a great impact on these devices. The game may consume a large amount of network resources, which could take valuable bandwidth away from other applications which may need it. Also virtual storefronts such as *Google Play Store* and the Apple *App Store* may be affected, especially due to their possible roles as alternate login accounts.

30c Potential User Problems

Network usage by the game, as discussed in 30b, may negatively impact users with data caps on their internet connections.

With the development of any video game comes the risk of addiction by its players. The current status of video game addiction is not fully understood and requires further research to gauge the true nature of addiction to video games. However it is apparent that some players become incredibly attached to the game, which has a negative impact on their mental and social health.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

This game is incredibly dependent on network resources, as it is basically unplayable without an internet connection due to the inherent multiplayer elements. While games could be played locally in the PC and home console skews, a network outage would greatly affect the user experience. This outage could come from the game's servers being down, either from widespread network outage or from failure of cloud resources. Network outages on the user end may also negatively impact the user

experience, such as players with unexpected outages or unreliable connections leaving a match before completion.

30e Follow-Up Problems

Our team has yet to determine whether using the likeness and name of the original product *Lunar Lander* may incur legal penalties from use of copyrighted material. Atari was the original publisher and developer of *Lunar Lander* and it is not known whether our team would need to consult Atari before using the IP [1]. Legal consultation would likely be needed to resolve this concern.

31 Migration to the New Product

31a Requirements for Migration to the New Product

N/A

31b Data That Has to Be Modified or Translated for the New System

N/A

32 Risks

A main risk to the project could be the inaccurate reading of the coordinates or collisions. Bugs like these can be dangerous to the product as it directly affects the end users. If there are inaccurate readings in either the collisions or the coordinate system, it can overtly affect the players gameplay. If players feel like the integrity of the game is being compromised by bugs, they will lose interest and stop using the product. Making sure that these bugs are resolved as soon as possible is important to keeping users interested and continue their desire to use the product.

Other risks could be if the server side encounters issues. If the server has issues and fails to run, none of the end users will be able to use their product. Making sure the server is stable is key to ensuring the product will work. Minimizing these risks are essential to the longevity of the product.

33 Costs

The costs can be fairly simple to breakdown. To start off with, the development of the game, so paying staff, to create and work on the game. Also with that could be using external resources to host a server and to maintain it. After the game is off its feet, a smaller team can be left to manage and update the game so that it is not neglected. So after the initial

development is done, there would be a maintenance team that would work on any bug fixes and push out balance patches. A reasonable amount of development time to make a fully working and appealing game could be half a year of development. Other costs could be time after development to be able to put it on all platforms whether it be on the playstation network, steam, the app store, etc.

34 Waiting Room

Some other ideas that would be worth considering that could make things fresh could be adjusting some of the mechanics. For example, instead of strictly landing on the moon, there could be a map where you can land on mars, where the gravity, speed, and other factors would be different compared to the standard moon map.

Another idea could be an implementation of different types of lunar landers, where ships can have different stats like its resistance to other ships, its weight depending on how much the wind will affect it, thruster power for how well you can act against gravity. Having different ships and having the players pick upon the start of each match would add more mechanics and require more skill to perfect.

35 Ideas for Solutions

We feel that Unity would be best suited for the development of this game. It would make implementation much easier, especially due to its included physics engine. Unity's extensive toolset for the development of two dimensional games makes it a promising choice for development. In addition Unity is very easy to port between skews, and would allow for wider adoption of the game.

It would likely not be practical to use vector graphics like the source material. Instead we should look into simulating vector graphics with more modern pixel graphics, or possibly a two dimensional compression of three dimensional generated models and landscapes.

The scores and statistics for players would be best stored in a mySQL database, this would make storage and retrieval very convenient.

36 Project Retrospective

What Went Well

Communication within the group went very smoothly, especially due to the use of Discord as a communication medium during remote development. It allowed us to share files and resources in the same space in which we held meetings and discussed plans and reports. The ability to communicate on mobile devices with Discord allowed us to talk to each other without being directly in front of a computer and committing to meetings like other platforms like Zoom or Skype might have.

The division of labor during this project was fairly even and did not lead to much blocking

of other portions. Each team member successfully completed work they committed to, which allowed reports to get done much faster than if the work was improperly spread between team members. By planning in advance how to split up reports, we were able to reduce interdependence of team members, allowing parallel work on the same report.

The use of Google Docs ensured that all team members were working on the same version of a document at any given time, and entirely eliminated merge conflicts.

What Went Poorly

Formatting in this document was a nightmare, due to the conversion from Microsoft Word format to Google Docs formatting. A great deal of time was spent and a great deal of time was lost from the conversion from the given template to Google Doc formatting.

Oral presentations for this report were often rushed and poorly thought out. We should have spent more time planning the presentations, and gotten more of the report done before a presentation to minimize the new work needed for each presentation.

Some sections in the report were not fully understood, and thus the group lost points on it when we could have clarified on Piazza the section.

Some figures were rushed and poorly made, more work should have been put into these.

Suggestions

We should have thought more in advance about the system design of the project, so we didn't have to do all the brainstorming while writing the report itself.

Errors in early versions of the report should have been corrected so we did not lose points on successive versions of the report for the same errors.

If this section should include suggestions for the professor and administration of the course, ***PLEASE*** create a version of the template on Google Docs so students can clone it into their own Google Drives! Microsoft Word is not very helpful when multiple people are editing the same document, though I am aware that Microsoft is adding features to newer versions of Word to remedy this. In addition students who don't use Windows, or have not purchased Office products (Computer Science students mostly use Unix systems) will have a very unpleasant experience filling in this report in Word alternatives such as LibreOffice Writer.

V Glossary

Cross-Platform: A term used to describe how a software can be used across different types of

computers.

Cross-Platform Gaming: A term used to describe players using different video game hardware while playing with each other at the same time. For example being able to play in the same online session with players using Xbox, PS4, or a Nintendo Switch.

Cross-Platform Gaming Companies: Used to describe companies that hold an interest in investing in games that can be played on multiple computers/consoles simultaneously. Such examples of companies may include Epic Games or Blizzard.

Lunar Lander, Lander: Spacecraft which were used to transport astronauts from a rocket to the surface of a moon safely

Match: An individual play session of a video game.

Map: The playable area, generated for a single match.

Skew: A platform for which a game is developed.

VI References / Bibliography

[1] "Lunar Lander". Retro Gamer. No. 79. Imagine Publishing. July 2010.

VII Index

N/A