

System Analysis and Design Report.

Intended for Jonathan Weightman.

Prepared by

Group PA2207:

- *Djed Curtis (19744647)*
- *Hasan Karabork (19679091)*
- *Jarrold Baker (20487821)*
- *Toufic Tannous (18528419)*

Executive Summary

The main objective of this project is to identify all problems and risk that that will occur when developing Clara world into 3D. The goal for the team is to design a test plan for the Clara so that the system won't crash or cause delays when the game is running. The team intentions are to clearly control any issues that arises and resolve them as quickly as possible without cause any delay. For the System Analysis report we plan to create a professional agreeable interaction between the client and the software team. We will be mainly focuses on advantages for the business for the system and controlling any risk that occurs in the project.

This will help the team to communicate with the client and testers for the detail design so that all codes are functioning in the right order so that the system will be finalized and established. The main aim is creating a breakdown of the official plan agreed on the team and client in order to find all problems that Clara faces in which they are properly fix with the help of the right tools and codes. The main purpose is to create a Clara 3D game that is dedicated to the needs of the client and supervisor and outline any essential fix, so that as a team we can improve the functionality of Clara in the project.

The main factors that are considered in the report are mainly based on the time to create the project and the quality of the scope so that all the elements are finalized in the right order within the project. Our aim task is to explore new tools and features in order to improve our main objective in creating new models an exporting them in the game.

Table of Contents

Introduction	4
Problem Statement	5
Client Statement	6
System Requirements	7
Functional Requirements	7
Non-Functional Requirements	9
Risks and Constraints	10
Detailed System Design	11
Use Case Diagram	11
Expanded Use Cases	12
Sequence Diagrams	14
Entity Relationship Diagrams	16
Screen Designs	17
Test Plan	20
Features / Use Cases to be Tested	20
Conclusion	21

Introduction

The main intention for this document is to present any risk or issues that the project faces and try to identify test plan so that they can be fixed and properly maintained with the right codes and control. The goal is to provide new ideas and solution that can be incorporated to Clara 3D with the help of new tools and features that will utilise in new technology so that all problems are controlled, and all risk are eliminated. The main idea of creating the SAD report is to enable that the breakdown of the task and help find better solutions by identify and risk associated in the project outlined by the team and Client.

As a team we can provide solutions that we can take in consideration of internal aspects that can adopt the project development in the time frame given which will help in us to analysis and present the project which will benefit us to move and excel forward on creating a likeable Project. In the document we will outline the constraints, Problem statements and risk associated when developing Clara 2d into 3D environment .

Problem Statement

Clara's world is a game designed around learning how to write code. It was designed and developed at the University of Western Sydney so that its ICT and computer science related students would have an easy platform to develop and practice their coding skills without having to dive into something too unfamiliar.

Problems we have identified:

- The current state in which Clara's world has been depicted is only two dimensional, the main issue this 2D depiction promotes is that it is quite far behind the modern expectations of how games should look and feel, this may cause some students to not feel as immersed in the experience as they should be.

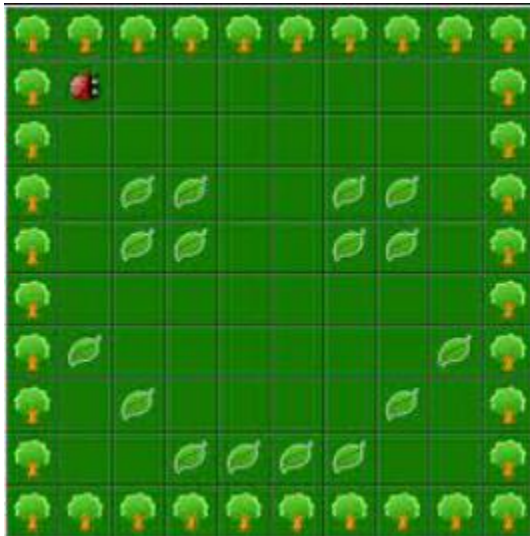


Figure depicting current state of Clara's world

Our goal is to transform the current two-dimensional depiction of Clara's world into something more in line with modern norms, namely making the game three dimensional. We believe that this portrayal of Clara's world is the next logical step in the evolution of this game.

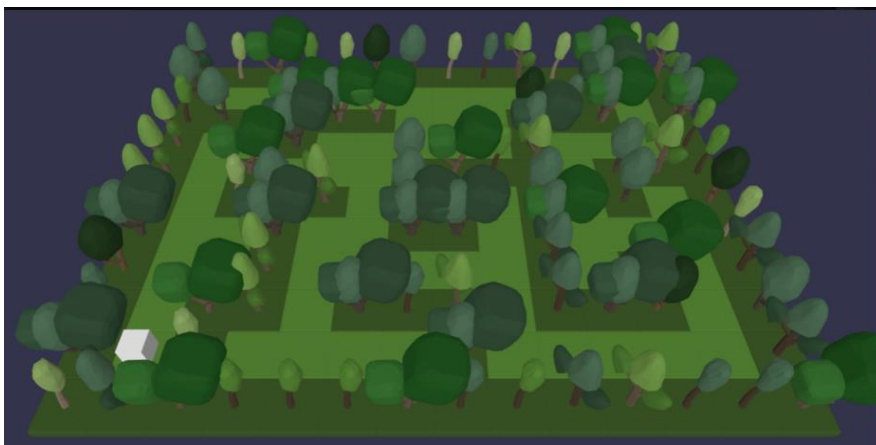


Figure depicting our vision of Clara's world

The main benefit from achieving our goal is to have students feel more immersed during their experience within Clara's world which may allow for their generation of coding skills to be greatly enhance

Client Statement

Tomas Trescak has previously attempted to deliver modernised adaptations of Clara's World using three-dimensional game development technologies such as Unity. However, now with more advanced and web-based three-dimensional game engines, the visualisations imagined, are able to become a reality.

The final system will include two themed version of the three-dimensional world, including the default theme of "Clara in the Woods" and the alternative them of "Clara in Space". It will also include an updated administrative environment, allowing for Clara's World boards to be created easier.

System Requirements

Functional Requirements

FR01 – BabylonJS Library

Features: The BabylonJS library provides the engine and workspace required to develop and run the web-based three-dimensional adaptation of Clara's World.

Description: In order to run the game well, and in any common web browser that supports it, the BabylonJS library has been used.

Justification: Due to the BabylonJS libraries' allowance of quick load times and minimal performance issues, the BabylonJS library is the required web-based game engine to use for this project.

FR02 – 'SceneLoader' Class

Features: Class of the project that defines the scene (game) and runs the BabylonJS engine.

Description: The SceneLoader class is the parent class that generate the engine and actual game itself, all classes and algorithms are either in the SceneLoader class or children of it. The class defines the scene, loads the engine, and generates the board.

Justification: Without the SceneLoader class, the three-dimensional world would not be generated at all, and thus it is required.

FR03 – 'MeshLoader' Class

Features: Class of the project that loads the three-dimensional models into the game from the provided web server.

Description: The MeshLoader class is a child of the SceneLoader class, its only function is to pull all the three-dimensional models from the provided web server into the game. Whenever a mesh/model is placed in the game, it's instanced from the MeshLoader class's list of models.

Justification: The MeshLoader class is required in order to load all models for the game, this includes the 'tiles', trees, and miscellaneous filler items.

FR04 – Board Generation Algorithm

Features: Algorithm written in TypeScript which utilises the BabylonJS library in order to generate the desired game board.

Description: Once a map is selected by a user, the board generation algorithm will translate the respective JSON file into a three-dimensional board comprised of GL Transmission Format (GLTF/GLB) models pulled from the provided web server.

Justification: The board generation algorithm is needed in order to produce a viewable and playable board for the user.

F05 – Movement Class/Functions

Features: The movement functions allow Clara to actually move and interact within the world.

Description: Once the user has run their code, Clara moves by using the movement functions that are triggered. These functions utilise the BabylonJS library's mesh movement functionality, which allows Clara to move around the board.

Justification: In order to have Clara move respective to the code written by a user, the movement functions are required.

FR06 – Procedural Board Beautification Algorithm (PBBA)

Features: The PBBA has been written in TypeScript and also utilises the BabylonJS library to beautify the selected game board.

Description: After the board generation algorithm has run, the map must now be beautified, this is done through the PBBA. The algorithm will place an item (grass, rocks, mushrooms) roughly every one in six tiles, and will randomise the placement of the item on each 'tile' as to give the board a less uniform look. This algorithm also pulls the GLTF/GLB models from the provided web server.

Justification: In order to give the board more character, the PBBA is used. The amount of items placed on the board can also be changed within the code, allowing for anything between one in one to ten tiles

FR07 – Scaling and Positioning Functions

Features: Used to scale and position models once instantiated into the scene.

Description: Once the MeshLoader class has been called and a GLB/GLTF model has been instantiated, a scaling function will run to determine the sizing of the model, followed by a positioning function, which will place the instantiated model into the correct area on the board.

Justification: To have the models placed in the correct areas and of the correct sizes, these functions are required. If they were not used, all models would be out of place and on incorrect axes.

Non-Functional Requirements

NFR01 – Usability

Description: The final system must be usable. The game must function correctly, it must operate based on the code the user inputs, and the Clara model must follow the rules of the board. That is, not falling off the map, walking through solid objects, and avoiding collisions.

Measure: Rigorous testing has been conducted throughout development, and further testing will continue during the testing phase of the systems development.

NFR02 – Quality

Description: Quality is identified in the systems board design and beautification. In order to have the games layout be desirable, a high standard of quality must be followed.

Measure: The quality of the system is measured by the overall design and look, the game must be able to run well, but also have a high-quality design.

NFR03 – Reliability

Description: Reliability in this system is defined by the outcome of the generation of each board.

Measure: The generation of boards must be reliable; this has been ensured through thorough testing with many boards.

NFR04 – Scalability

Description: The scalability of the system is defined by how well it can expand and how easily new boards can be implemented into the game.

Measure: The final product will support all the currently defined boards, however, due to the board generation of the new system, any board can be added through the same means used for generating a two-dimensional world.

NFR05 – Flexibility

Description: Flexibility in this system is required for the addition or replacement of three-dimensional models used in the game. The system must be flexible in order to allow for models to be changed and also for all kinds of board definitions to be generated.

Measure: The two functions of the system that require flexibility (generation and model loading) are built in a way that incorporate adaptability. The MeshLoader class covers the model loading flexibility, and the board generation algorithm covers the generation flexibility.

NFR06 – Maintainability

Description: Maintainability is identified in the server-side of the system, more specifically, how the game is maintained. It's important that the source code is updated if any future changes are made to the BabylonJS library that have an effect on the game.

Measure: The maintainability of the system is measured by regular checks of the source code to see if any in-built functions or commands have been depreciated.

Risks and Constraints

Risks are commonplace for any project regardless the scope, the only thing that changes is the amount of risk and this correlates with the size of the project i.e., the larger the project, the more risks are associated with that project.

In terms of risks involved utilising our current development strategy:

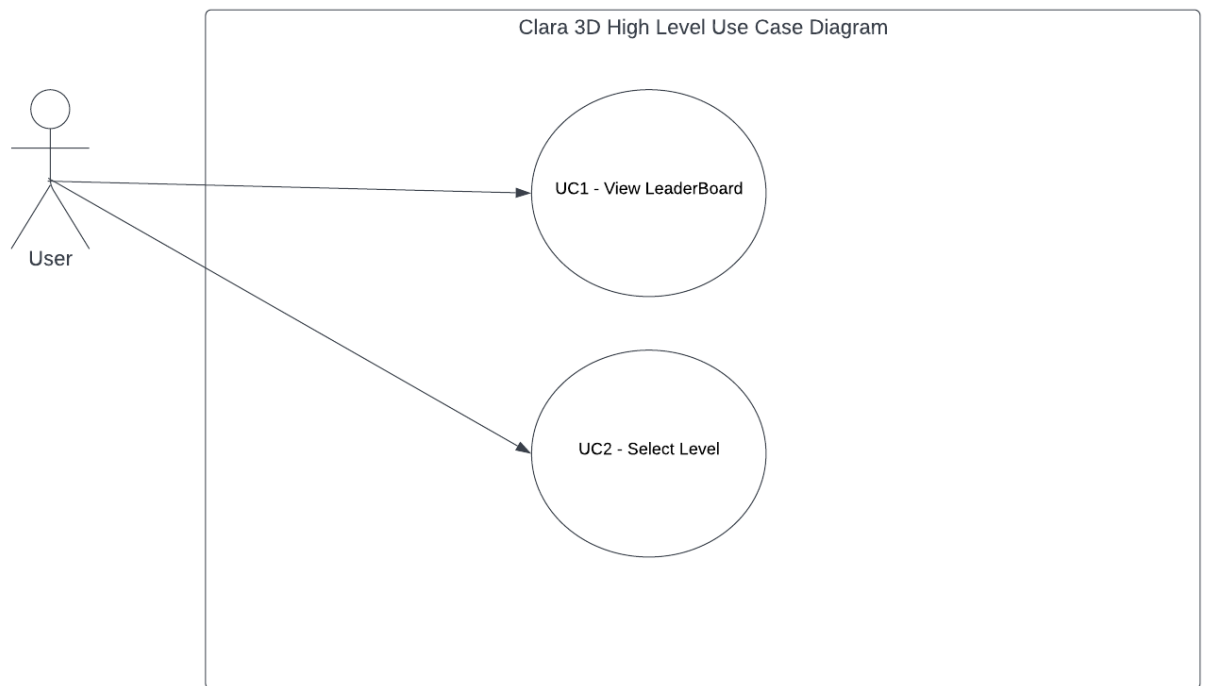
- It would be on par with any other form of web development. It is more likely for developers to embed risks into code during the development phase unknowingly, these forms of risks can be eliminated through thorough quality assurance.
- One potential risk that comes directly from the library itself is that an external module is needed to ensure that strict content security policy (CSP) is enabled, without this external module a door to cross site scripting (XSS) can be opened. This problem can be eliminated by using the aforementioned module ("PIXI requires unsafe CSP - Issue #7324 · pixijs/pixijs", 2022).

Constraints are prevalent in all systems regardless of their use and how they have been developed. For our chosen system, the following constraints are:

<i>Constraint Type</i>	<i>Explanation</i>
Hardware constraints	This is in relation to our intended user's hardware i.e., their own devices they will use to connect to Clara's world. Due to the nature of the system which requires the rendering of different 3D objects, this will tax the users processing performance a fair bit. We have however, implemented a feature that will allow users to select the level of detail they desire, this will allow for users with less powerful devices to still have a decent experience with Clara's world.
Internet speed	The speed of the user's connection will also affect how quickly the user can be loaded into Clara's world, this may in turn also effect their general experience with Clara's world
Browser compatibility	The users chosen browser and version may affect their experience in Clara's world. This is because different browsers can interpret scripts differently, this could result in system crashes or pages being loaded incorrectly or in an undesired way.

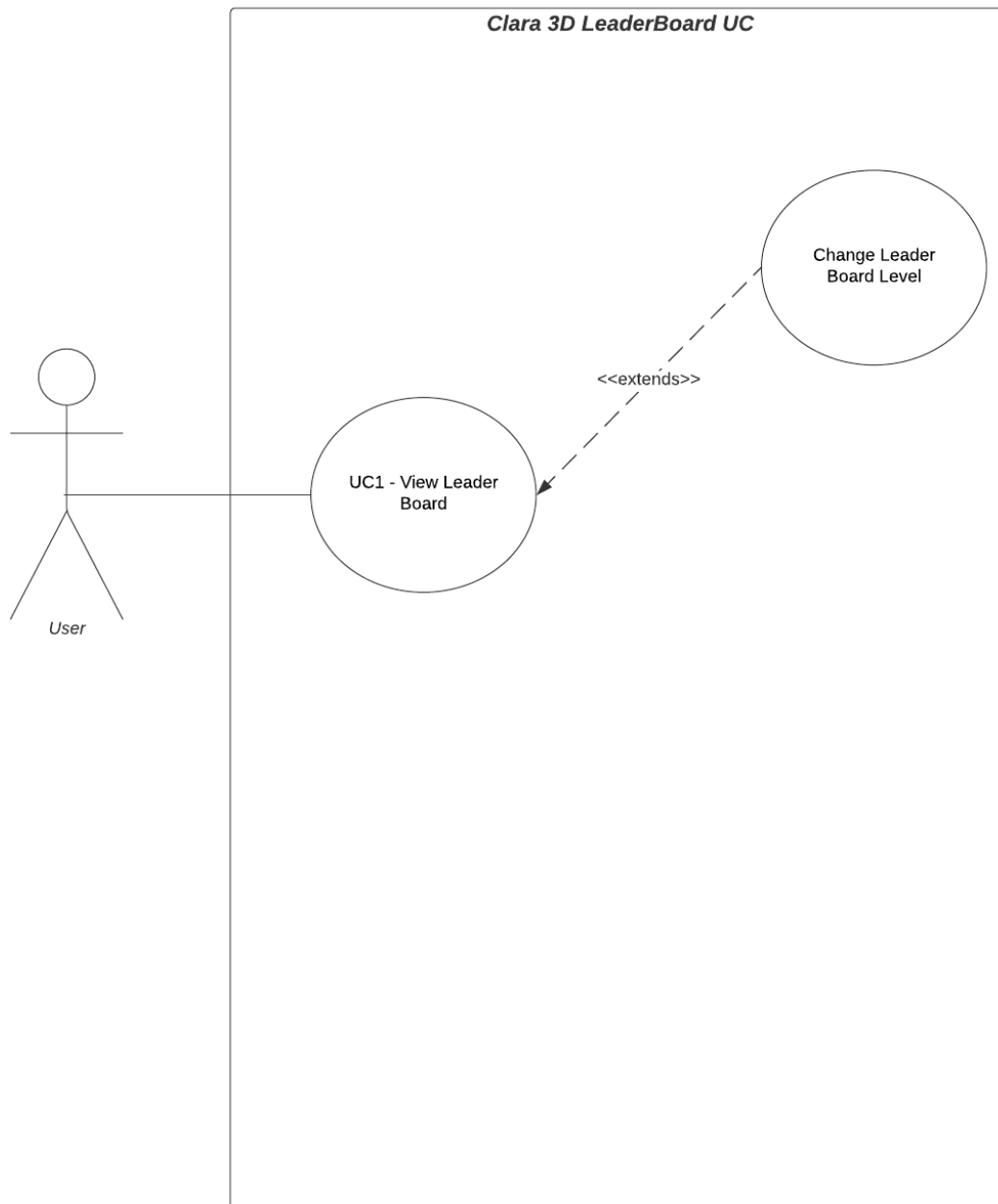
Detailed System Design

Use Case Diagram

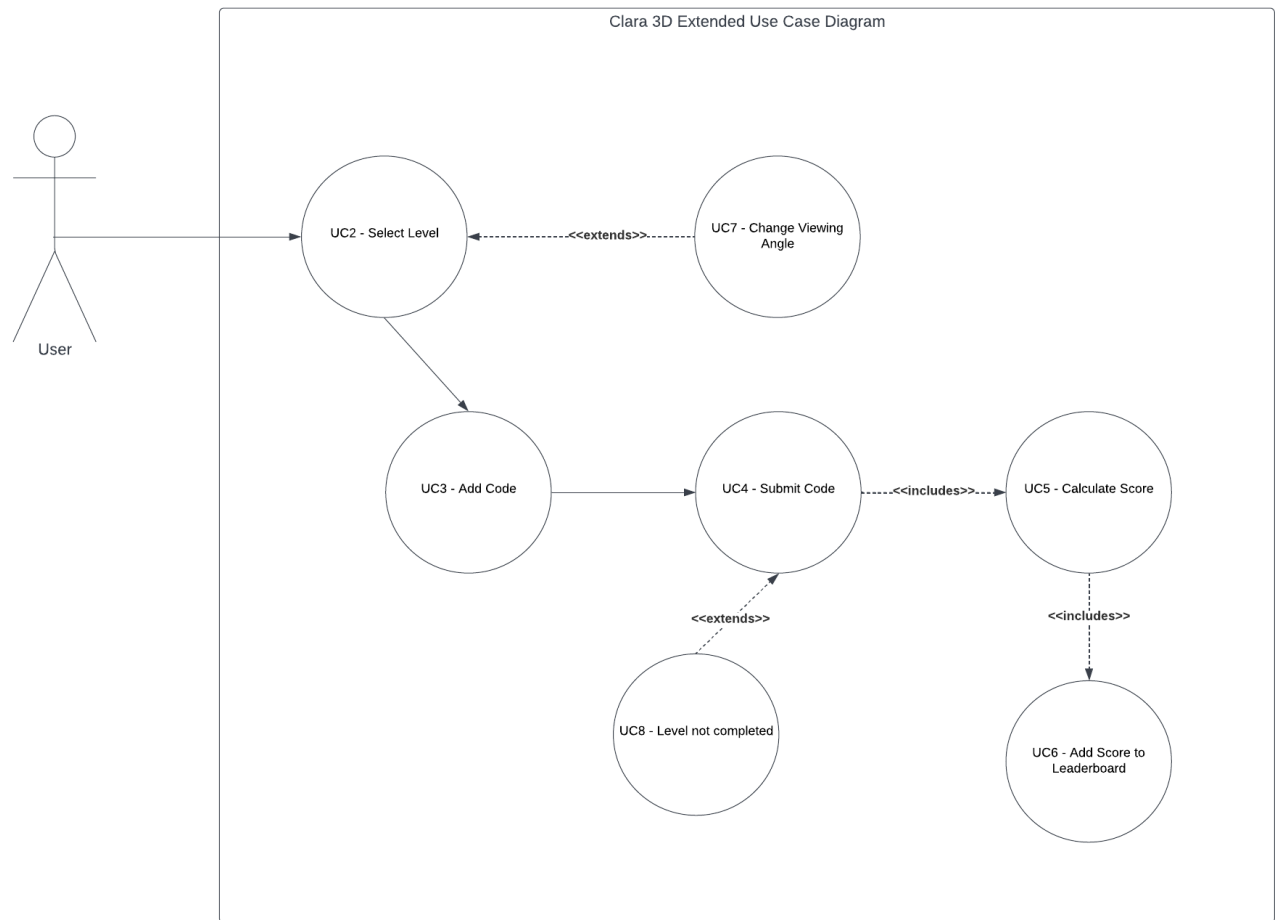


Expanded Use Cases

Leader Board Use Case Diagram

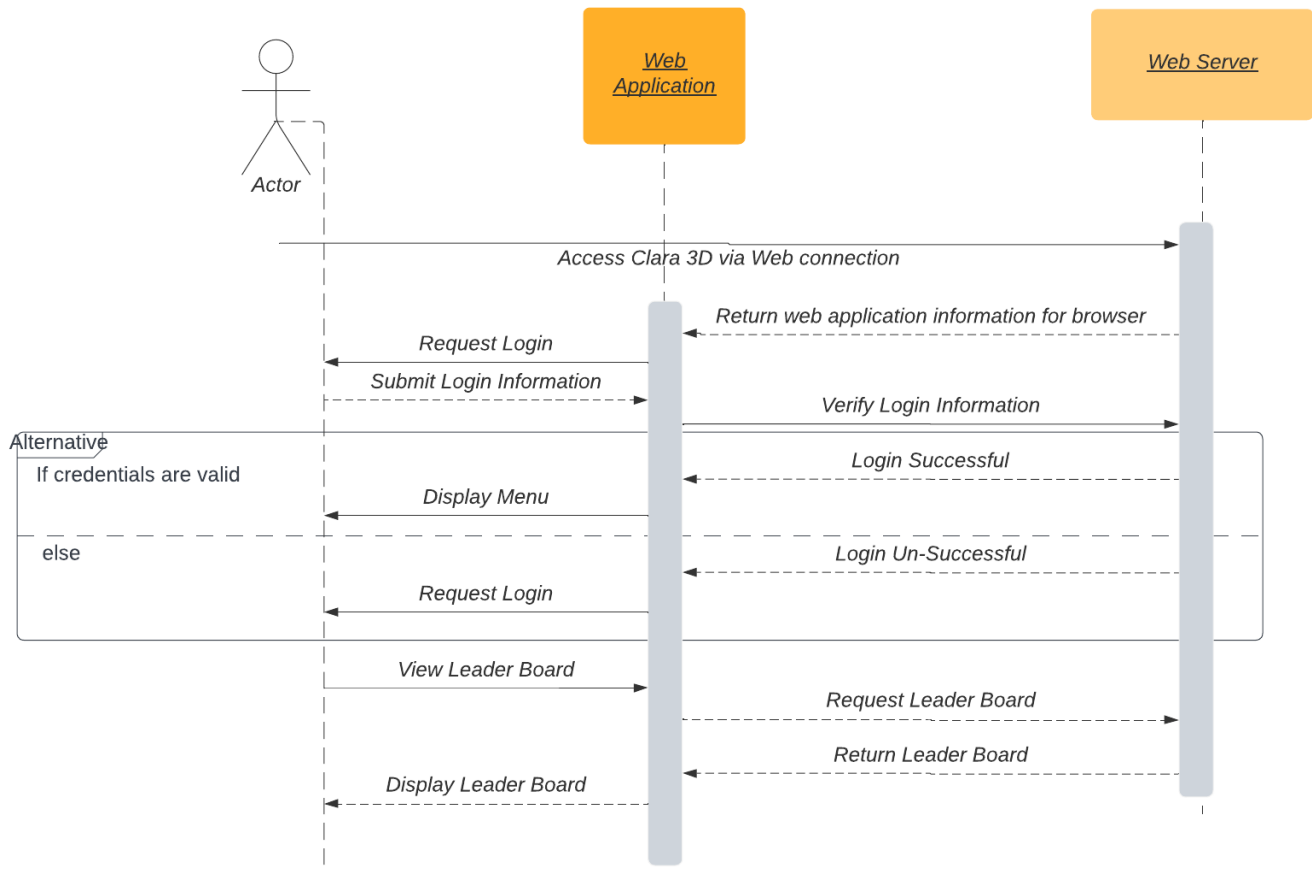


Level Use Case Diagram

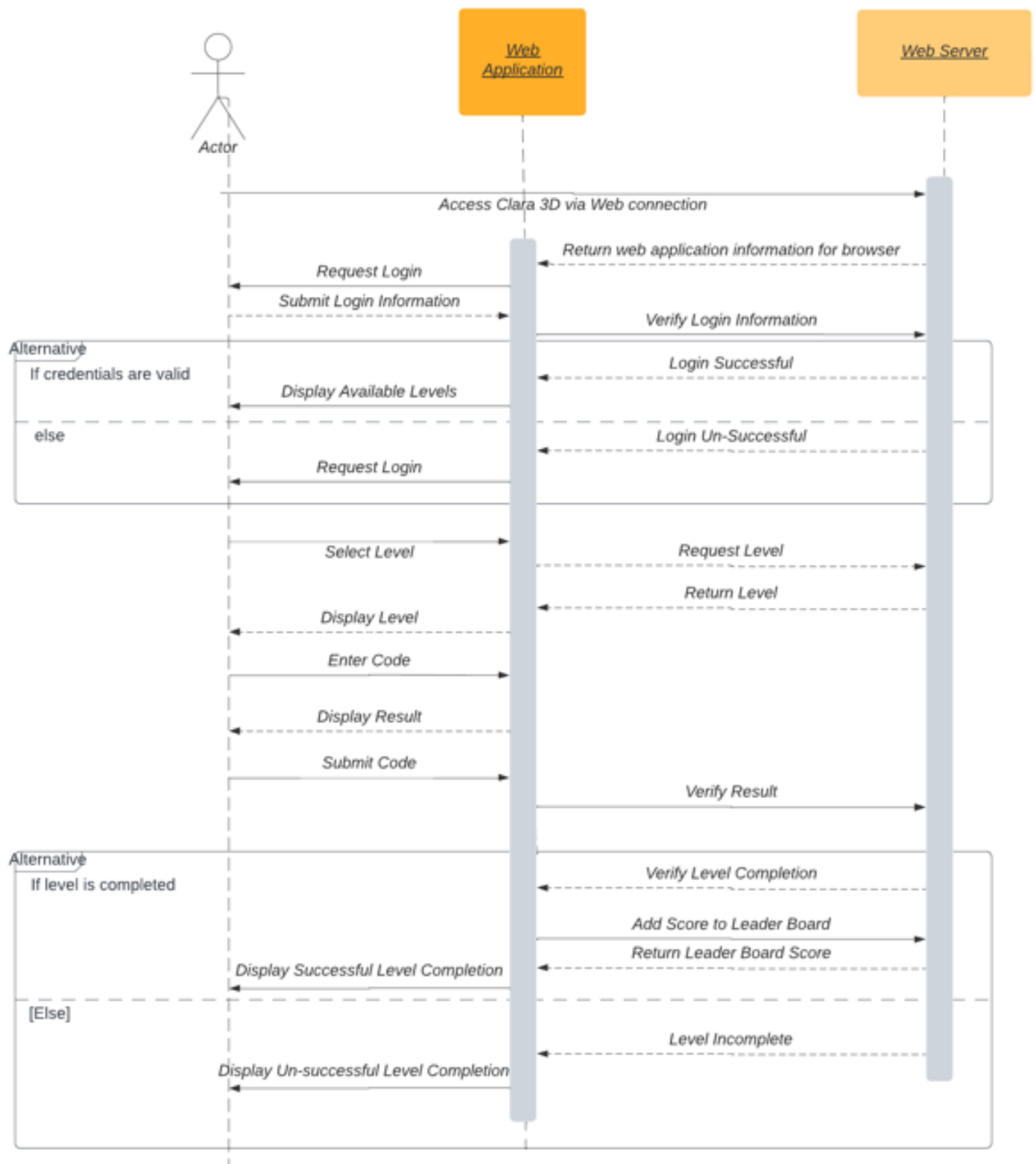


Sequence Diagrams

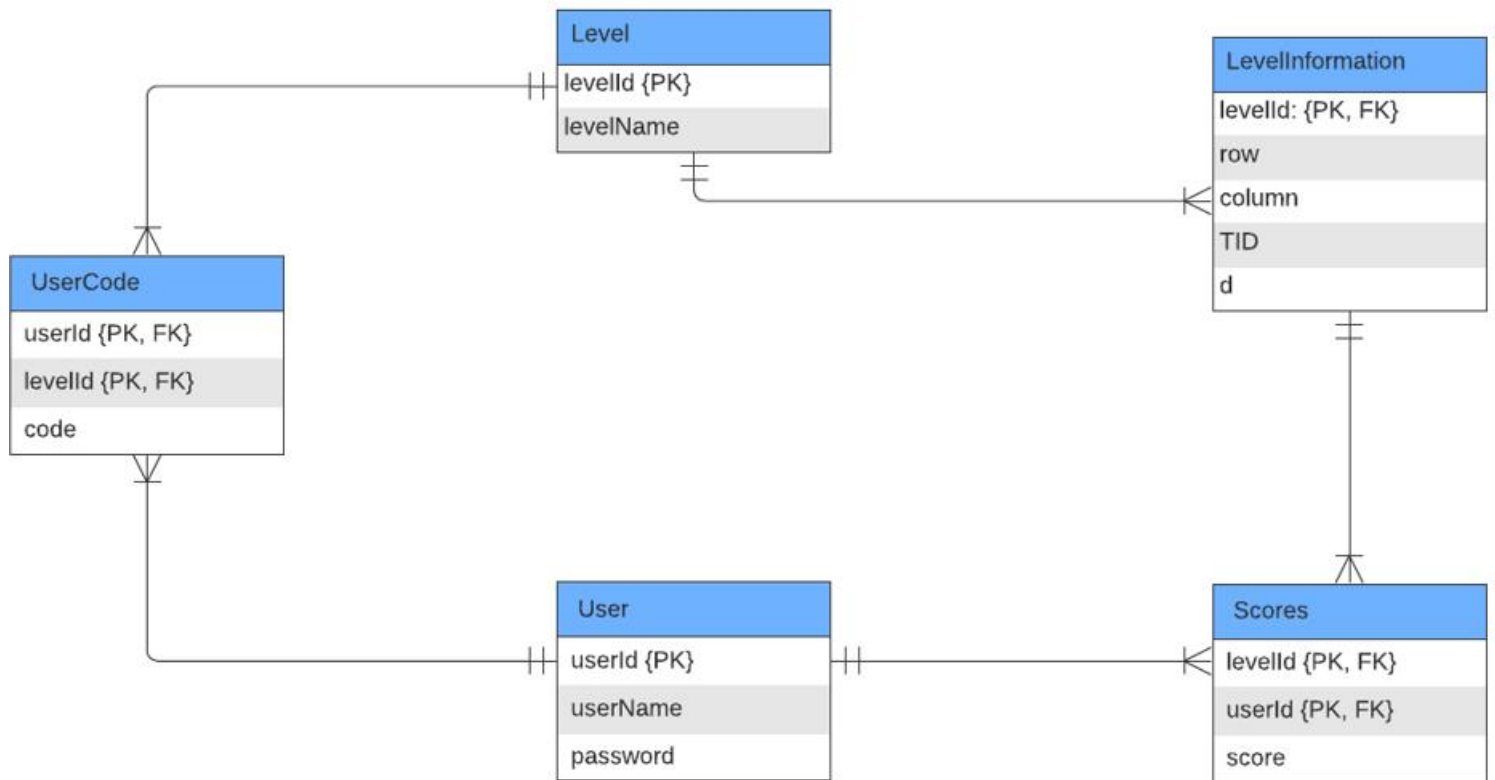
Leader Board Sequence Diagram



Level Sequence Diagram



Entity Relationship Diagrams



Screen Designs

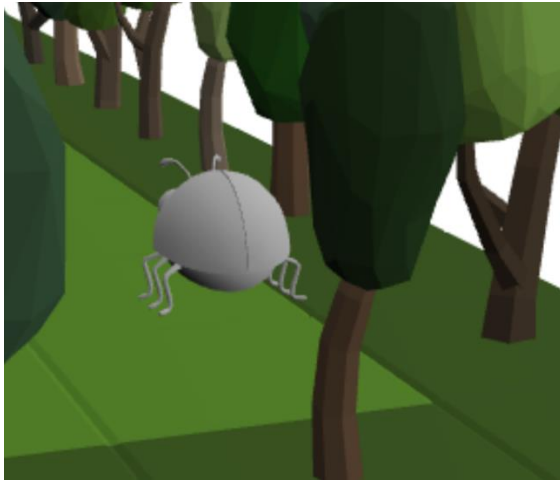
First Working Design



Original GUI Overlay Design (Sketch)



Clara Implementation Design Test vs Clara Correctly Implemented



Current Testing Screen Design

Detail Level: High



Ideal Screen Design

Detail Level: High



Test Plan

Features / Use Cases to be Tested

The below table details the features and use cases to be tested by our development team.

<i>Feature/Use case to be tested</i>	<i>Types of testing</i>	<i>Pass fail criteria</i>	<i>Personnel</i>	<i>When and where</i>	<i>Training</i>	<i>Risks</i>	<i>Contingencies</i>
Process user code	Acceptance testing	User code is able to manipulate Clara as per written code	Client Tomas	Week 10 at developer site or remote	Not needed	System crashes during testing	Re-develop features in an environment that will be more suitable to the back-end system used by the client
View user leader board	Acceptance testing	Upon selecting the leader board option, users should be displayed from highest to lowest scores	Team member Djed	Week 10 at developer site or remote	Not needed	System crashes during testing	Arrange for an environment separate from production and development to allow for further testing
Level selection	Acceptance testing	The selected level needs to be loaded in correctly as per client provided data	Team member Jarrod	Week 10 at developer site or remote	Not needed	System crashes during testing	Arrange for an environment separate from production and development to allow for further testing
Graphics detail section	Performance testing	Dramatic drops in frame rate will be considered a fail	Team member Toufic	Week 10 at developer site or remote	Not needed	System crashes during testing	Arrange for an environment separate from production and development to allow for further testing
Move camera	Acceptance testing	The camera needs to be able to move as per the selected view method/option	Team member Hasan	Week 10 at developer site or remote	Not needed	System crashes during testing	Arrange for an environment separate from production and development to allow for further testing

Candidate Test Cases and Test Data

The tables below depict the test cases and their respective test data, each including if the testing for the feature was successful or not.

Use Case / Feature:	Process User Code
Test Purpose:	To ensure that user written code is correctly processed within the three-dimensional visual
Expected Results:	See Clara move respective of the user code
Success/Failure:	Success
Test Data	
<i>Movement Function</i>	<i>Outcome</i>
Move	Clara moves forward one 'tile'
Turn Right	Clara turns 90 degrees to the right
Invalid Move	Game stops an error is shown to user
Place Leaf	A leaf is placed underneath Clara
Grab Leaf	The leaf underneath Clara is removed

Use Case / Feature:	View Leader Board
Test Purpose:	Confirms that the player can view the scoreboard for Clara's World
Expected Results:	User is able to view their place on the leader board as well as their peers
Success/Failure:	N/A

Use Case / Feature:	Graphical Detail Selection
Test Purpose:	Verifies that the graphical slider implemented in the game functions correctly
Expected Results:	Changing the position of the slider alters the detail of the game
Success/Failure:	Failure
Test Data	
<i>Options</i>	<i>Expected Outcome</i>
High Level Graphics (3)	All models will be used in board generation, no limit on file sizes
Medium Level Graphics (2)	A limited amount of models used in generation, some limit on file sizes
Low Level Graphics (1)	Very limited selection of models used and harsh limits on file sizes

Use Case / Feature:	Camera Movement	
Test Purpose:	Ensure that cameras in the three-dimensional environment have limitations as to restrict the user from breaking their view angle	
Expected Results:	Scene camera should behave correctly and not go into out of bounds areas	
Success/Failure:	Failure	
Test Data		
Camera Function	Description	Result
Angle #1	Top-down angle	Success
Angle #2	Side view angle	Success
Angle #3	Restricted free cam	Failure

Conclusion

In our final conclusion, we as a team can finalize that we have identified all risk and problems within the system design and understood the related setback in great detail so that we can fix our mistakes with better solution and answers. We have Establish a great connection with the client within the development process and also identify all risk and issues and resolve them in the right given amount of time and as well prepare final budget which estimates all the cost and time of the project. We identified advantages for the system and designed upcoming test plans on controlling it within the system.

We have created test plan to test our coding and functions so that as a team we can find any issues or risk within the project so that we can apply fixes and updates to them so that our final system will be ready to go without any problems at all. The benefits for our system are that it will be accurate and correct which will help users to access Clara 3D without no problems which will help improve communication within the system. Finally, as a team we can determine that the software we used for creating Clara 3D save us cost and time in developing the system.