CPSC 466-59

Dr. Chang-Hyun Jo

us

HOMEWORK 2 REPORT

Team 7 **June 19, 2023**

David Fazio	**davidfazio1101@csu.fullerton.edu**
Bryan Garcia	**pieismanly@csu.fullerton.edu**
Bryan Rivas	**brivas100@csu.fullerton.edu**
Pratyush Sahu	**pratyush.sahu@csu.fullerton.edu**
Juan Uriarte	**uriarte.juan@csu.fullerton.edu**

Table of Contents

Revision History:

Date	Author(s)	Tasks
7/7	Bryan Garcia	Project description and vision
7/7	Team	Created FRs and NFRs
7/8	David Fazio Bryan Rivas	Create Use Case Diagram
7/8	Juan Uriarte	Project plan
7/9	Juan Uriarte	Technology Preparation
7/9	Bryan Garcia	Traceability Matrix
7/11	David Fazio	UML, Detailed Class Diagram, Interaction Diagram
7/12	Bryan Garcia Juan Uriarte	Sequence Diagram, Architecturally Influential Factors
7/13	Pratyush Sahu	Architectural Design Document
7/14	Bryan Rivas	Transition Plan
7/16	Pratyush Sahu	Optional CMMI Section

7/23	**Juan Uriarte**	**Part B**
7/24	**David Fazio** **Bryan Garcia**	**Refine Diagrams from HW1**
7/24	**David Fazio** **Juan Uriarte**	**Interaction Diagram's for Functional Requirements**
7/25	**Bryan Garcia** **Bryan Rivas** **Pratyush Sahu**	**Game Design Refine**
7/27	**Pratyush Sahu**	**Part C**
7/31	**Juan Uriarte** **Bryan Rivas**	**Team Charter**

Project Plan:

1. What: The project aims to create a Roblox game that provides a virtual tour of the Computer Science Building and the surrounded areas at California State University Fullerton. It will simulate the interior of the Computer Science Building, classrooms, hallways, offices, and common areas. There will be a incorporation of interactive elements such as non playable characters (NPC's), obstacle courses, spawn points, and information displays.
2. Why: The virtual tour will serve as an informative experience for new students and their families, offering a great and interactive experience to explore the campus and what we learned so far by developing this game. It focuses on showcasing the faculty, resources, and facilities which will allow the users to acquire a comprehensive understanding of what the Computer Science Program at California State Fullerton University has to offer.
3. Who:

    Project Manager: Responsible for coordination, planning, and ensuring objectives are completed

    Development Team: Designers, developers, and content creators will help with the virtual environment in Roblox.

    CSUF Computer Science Department: Department layout and information.

    Students and their audience: Targets those who seek information about the Computer Science Department and those who want to play the game.

4. When: The project timeline will span one month starting from July 3, 2023 to August 4, 2023.

    1st Week: Requirements, brainstorming, preparations, asset creation, white board sketches, Stud Implementation

2nd Week: Diagrams, Campus Research, implementation of interactive features, Obstacle Course

3rd Week: Playtesting, feedback, NPC Scripts, Code Development

4th Week: Testing, bug fixes, finalization of virtual tour, Post Evaluation

5. Scope:

a. Depiction of CSUF Computer Science Building interior, classrooms, offices, and common areas.

b. Creation of NPC's representing the students and Professor Goffman with dialogue and scripts related to Computer Science.

c. The Professor's Office will be located on the 4th level with the involvement of Computer Science material on the walls.

d. Obstacle Course will start from the Parking Structure to the Computer Science Building. There will be a harder Obstacle Course shown in the same location for the player to pass through. These integrations of obstacle courses are required to pass in order to talk to Professor Goffman while there are other students trying to distract the player.

6. Estimation: The project is estimated to take one month with the help of five students. Playtesting and bug fixes will account for 25% of the total effort.

7. Resources: Roblox Studio will be utilized as a development tool for building the virtual environment. Other aspects of Roblox Studio that will be utilized is the 3D Modeling of Buildings, the Stud Implementation on buildings and playtesting for potential bugs and errors. Collaboration from Google Earth will be utilized to depict the Computer Science Building and surrounding areas to our virtual environment.

8. Logistics: Communication and feedback from individuals in our team. Project Managing tools for task tracking and a road map of our game development.

Our objective is to successfully develop and deliver an immersive virtual environment tour of the Computer Science Building in Roblox. The virtual experience is created to have the user engaged with the obstacle course, NPC's, the design features, and Professor Goffman's word of advice on Computer Science material. We are confident in creating an enjoyable exploration of the Computer Science Building, providing valuable insights of our Computer Science Program.

Inception

Project Plan

Our team will utilize Roblox Studio to create a virtual environment of the Computer Science Building located in California State University Fullerton. The surrounding areas will also be created with obstacle courses being showcased as a route that ends at the Computer Science Building. There will be Non-Playable Characters whose whole purpose is to distract the player from entering Professor's Goffman's Office for help on related Computer Science material. The purpose of this project is for students and their families to experience the campus by the tools we learned so far from the Computer Science Program in Cal State Fullerton.

Creating a fun game that involves tools we learned from our previous classes is a perfect way to show our skill sets. Bryan Garcia and David Fazio will be the developers for the Roblox Game. Juan Uriarte, Bryan Rivas, and Pratyush Sahu will work on the documentation and the planning of creating an immersive game. For functional requirements, the game will consist of Interaction with NPC's, Time Scaling, Information Displays, Obstacle Courses, Social Interactions, and Professor Goffman's After Hours Meeting. Nonfunctional Requirements will consist of the game running without any errors on Windows and macOS. The project will be built with Roblox Studio and will require minimal memory to run.

Project Description

We are going to create a Roblox game depicting the interior of the Computer Science Building and the Computer Science program at Cal State Fullerton. We are going to create a time scale depicting the amount of people inside the building depending on the time of the day. We will also create Non Playable Characters depicting students and professors that will be asking questions on Computer Science related material. Professors' offices are going to be shown on the 4th level and information material on the walls that can be useful to students such as a road map. Facts from the Computer Science department will also be illustrated on the walls inside the building such as when the building was created, the faculty in charge of the department and more related material.

Vision:

Our project goal is for new students, specifically computer science majors and their families to virtually tour CSUF campus(Computer science and Engineering buildings). We want the virtual tour to not only be fun and enjoyable, but to make it fully interactable. Giving students and families the chance to interact with the faculty, sit in on virtual class and too see exactly what the CSUF computer science department has to offer. Users will also have the chance to interact with other students on campus and experience what their journey at CSUF will look like as a computer science major.

UP Development Case

Discipline	Techniques	Artifact	Incep.	Elab.	Elab	Construction	Construction	Transition
		Iteration	I1	E1	E2	C1	C2	T1
Requirements	*Requirements workshops, preparations, brainstorming, prototypes,*	*Vision* *Requirements(FR and NFR)* *Usecase* *Feature List* *Prioritization on Objectives* *Technology Preparation*	S S S S S S					
Design	*Diagrams, campus research, whiteboard sketches*	*Architectural Design Document* *Detailed Design Diagrams* *Campus Layout* *Asset Creation* *Building Layout* *Classroom / Office Layout* *NPC Layout* *Obstacle Course*	S S	R R S S S S S	R R R R S	R R R R R R	R R R	
Project Mgmt	*Unified process practices, scrum, agile*	*Team Charter* *Plan scope of requirements/timebox* *Team Scrum Meetings* *Task Planning* *Team Roles* *Post Evaluation*	S S S S S	R R R R R	R R	R R	R R	R R S
Implementation	*Pair programming, continuous integration*	*Stud Implementation on Buildings* *Asset Integration* *UI Implement* *Coding* *Playtesting and Feedback* *Debugging* *NPC Scripts*		S S S S S	R R R R S R S	R R R R R R R	R R R R R R	
			R = Refine					
			S = Start					

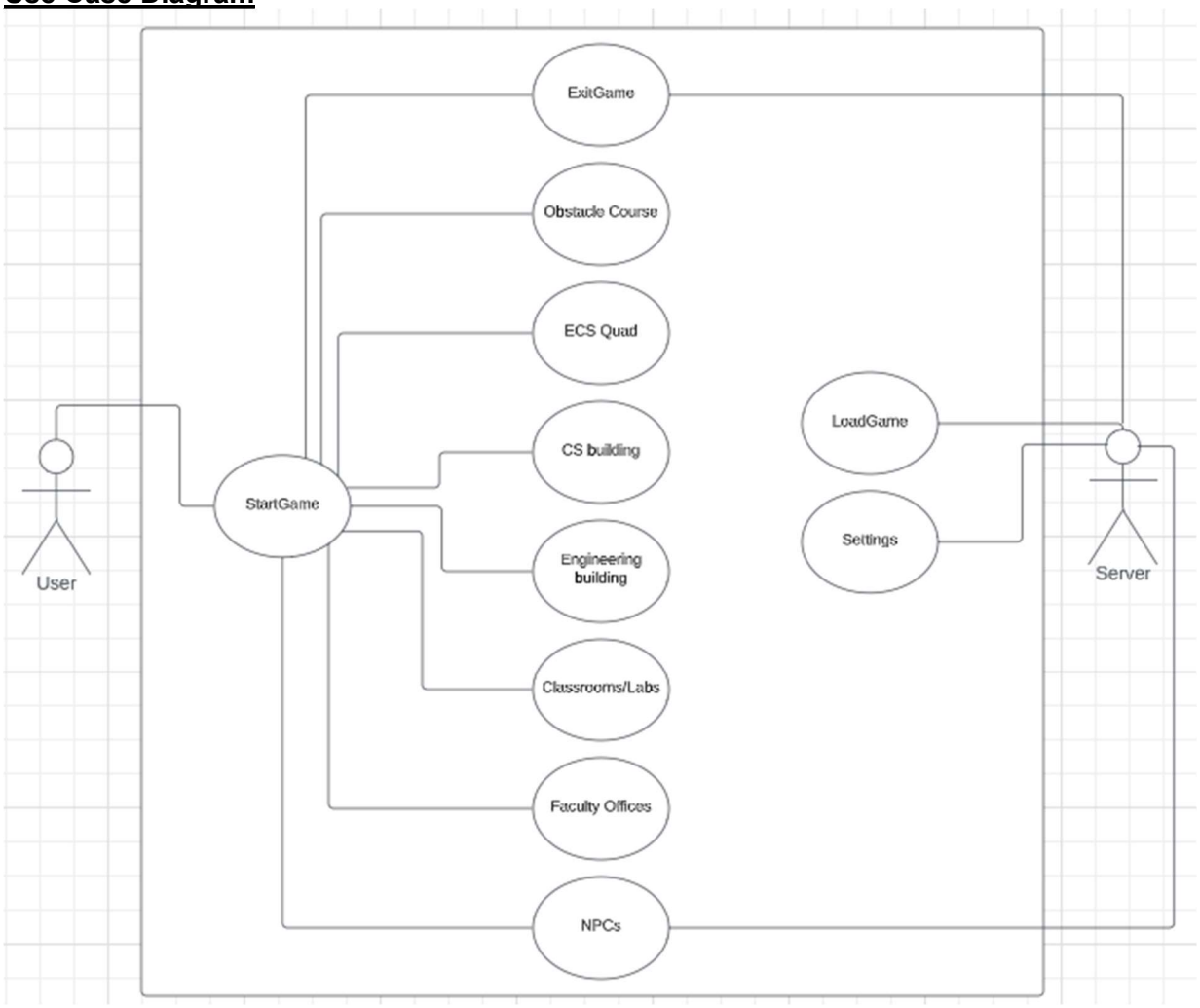
Functional Requirements:

1. Create CSUF Computer Science Building
2. Create CSUF Engineering building
3. Create CSUF Computer Science and Engineering Quad
4. Construct CS Classrooms/Labs
5. Construct CS Faculty Offices
6. Construct Engineering Classrooms/Labs
7. Construct Engineering Faculty Offices
8. Allow users to interact with faculty and other students
9. Program a short class taught by Gaufman that people on tour can view
10. Construct a obby (obstacle course)

Non-functional requirements

11. Game runnable on Windows, macOS, iOS, and Android
12. Create surrounding buildings of ECS buildings
13. Create landscaping
14. Create characters(students and faculty)
15. CS parking lot

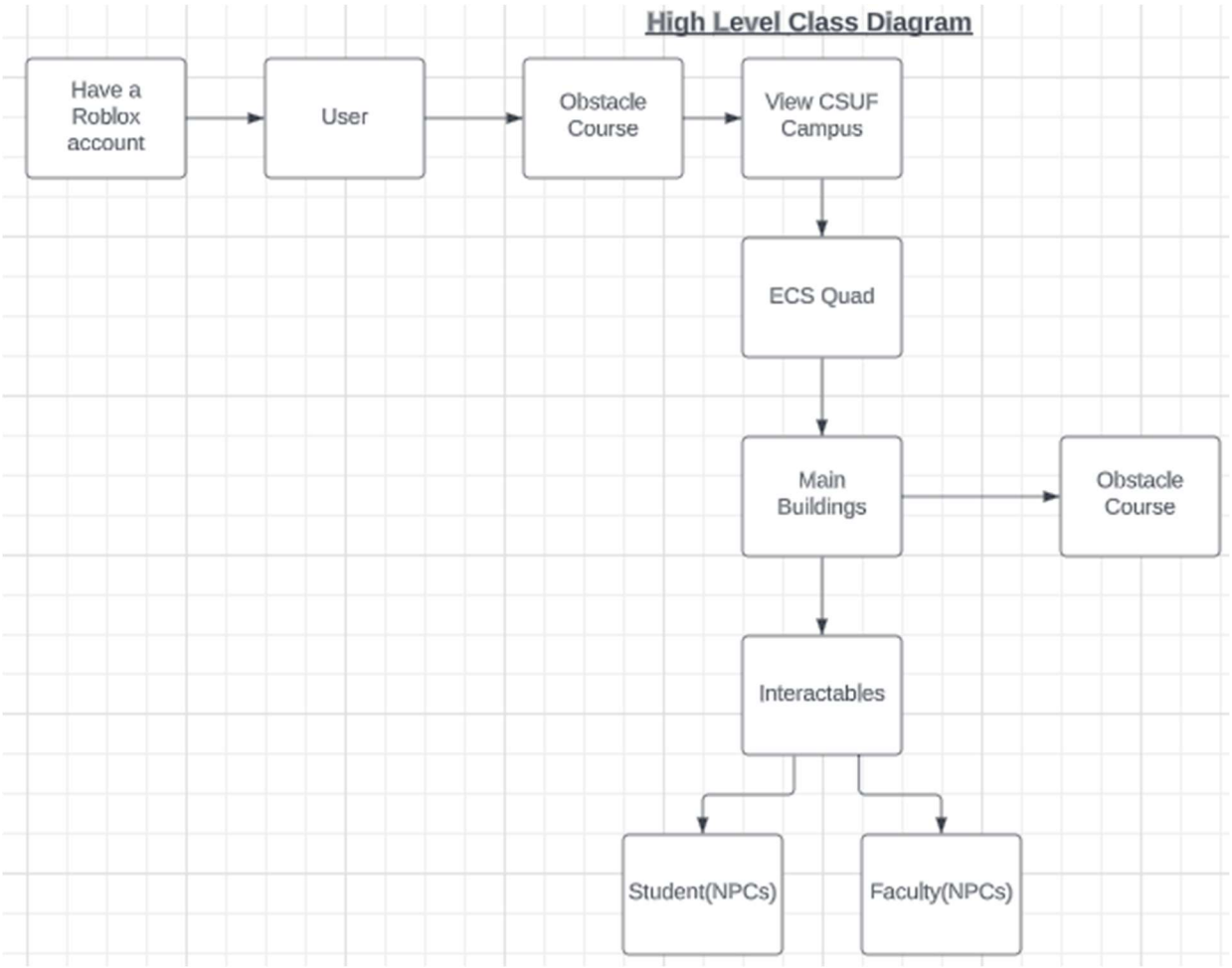
Use Case Diagram



Use Cases

	Use Cases	*Actor*	*Description*
1	Start Game	Player	The player will be able to view the campus, open doors, go thru obstacles, and interact with students and faculty.
2	Exit Game	Player	The Player will be able to exit the game at any time.
3	Obstacle Course	Player	The player will go thru a obstacle course located in the Parking Lot, the obstacle course will lead to the CSUF Campus.
4	CSUF Campus	Player	The map overlay of the game consists the Computer Science / Engineering area and the East Side Parking Lot.
5	Engineering Building	Player	There will be a obstacle course connecting the Engineering Building. Doors will have the function of opening.
6	CS Building	Player	There will be a obstacle course connecting the CS Building. Doors will have the function of opening.
7	ECS Quad	Player	NPC's walking around with interaction towards other NPC's.
8	Faculty Offices	Player	Gaufman's office with Faculty Interaction towards the Player.
9	Classrooms/Labs	Player	NPC's Students will be inside interacting with FAC NPC.
10	NPCs	Server	The server will spawn NPCs and Assign them roles of either Student NPCs or Faculty NPCs
11	Load Game	Server	The Server will load the CSUF campus tour game.
12	Settings	Server	The Server will have settings for the whole game.

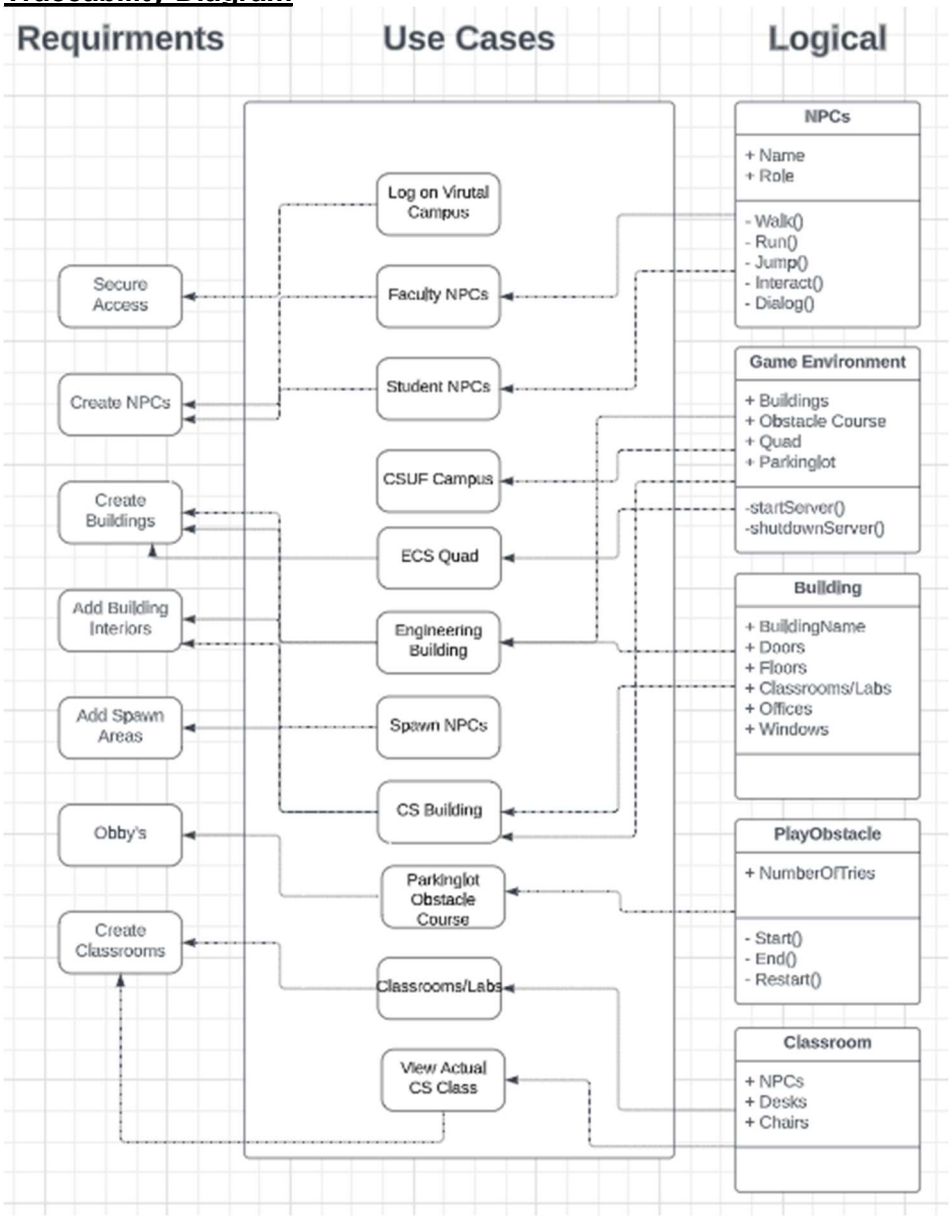
High Level Class Diagram



Traceability Matrix

	UC = Use Case											
Requirment ID	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
1						☑			☑	☑		
2					☑							
3							☑					
4									☑			
5								☑				
6									☑			
7								☑				
8							☑				☑	
9						☑			☑		☑	
10		☑	☑	☑								
11	☑											☑
12				☑								
13			☑	☑			☑					
14	☑										☑	☑
15			☑	☑								

Traceability Diagram



Elaboration

Technology preparation:

The members with experience of Roblox started on the creation of the game through Roblox Studio. Other members that were not familiar with Roblox went through the tutorials on the homework. Once through the tutorial videos we made a sample game that we built an obstacle course and played around in Roblox studio as a team. We arranged a group call so that we were able to help each other with the fundamentals in Roblox Studio. The other members focused on the planning and the requirements of the game. Including documentation which resulted in the team to pre planned accordingly to what objectives needed to be done before the start of a new week. We all participated in the linked tutorials of Roblox in order to get a better understanding on how to use Roblox Studio. We also used outside sources such as Youtube to figure out how to implement the open and close of a door while playing the game. This resulted in coding and learning how to implement these code changes to various other doors and types of doors. The Stud Implementation through the help of Google Earth utilized during the creation of our buildings helped us provide a realistic view of the Computer Science Building and its surrounding areas by using a 3d modeling and texturing point of view. We added NPC's in our game, which also resulted in having to do some further research to add dialog so that users can talk and interact with other characters. We were able to learn how to write scripts for the NPCs to move and talk. We learned about obby's in Roblox and how to construct them and make them challenging to the user.

Architecturally influential factors:

1. **Interactivity and Engagement**

   Incorporating interactive lectures and classrooms allow players to get a feel for the learning environment, attending classes, and participating in educational activities. Also interacting with the students and engaging in conversations make the experience more immersive.

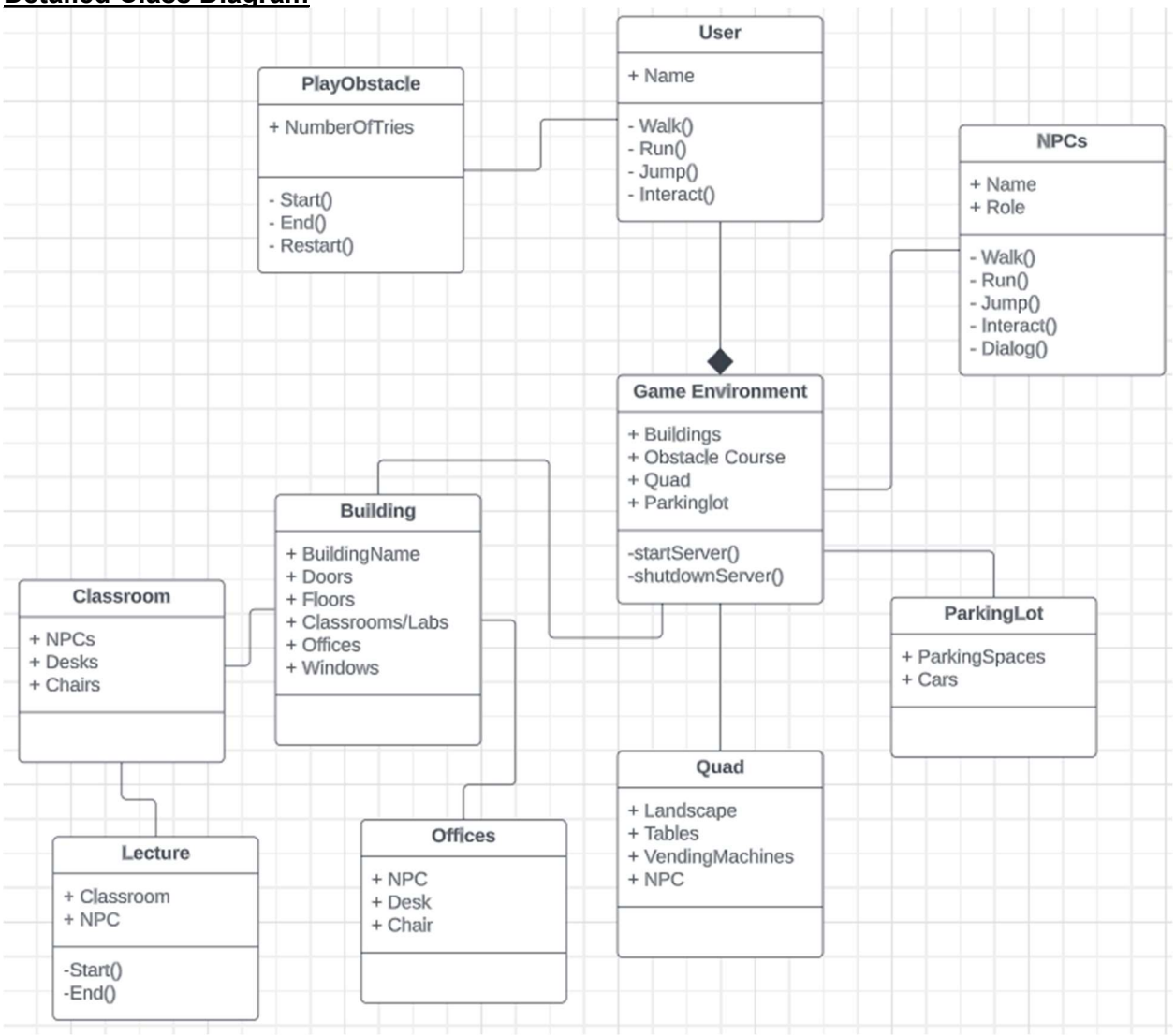
2. **Performance and Scalability**

   It's essential for our game to run smoothly on various devices and platforms, the game's optimization will help prevent lag, reduce loading times etc. Managing in-game resources and performance is essential for delivering a high-quality experience.

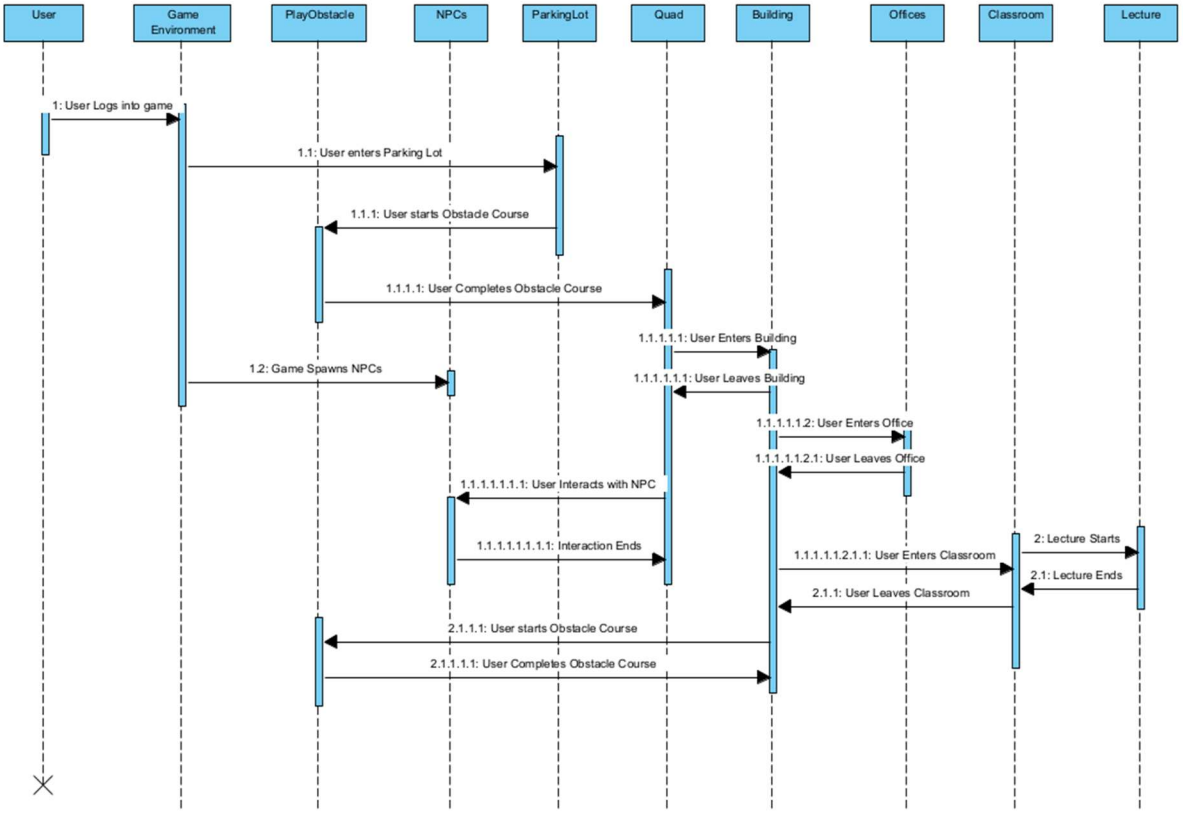
3. **Game Boundaries**

   Every game has its own borders and boundaries, especially with restricted areas and out of bounds areas. These boundaries are placed so that players do not go out of bounds and play within the environment that is developed. As the environment of the game focuses on the CS building and a couple of the surrounding buildings as well.

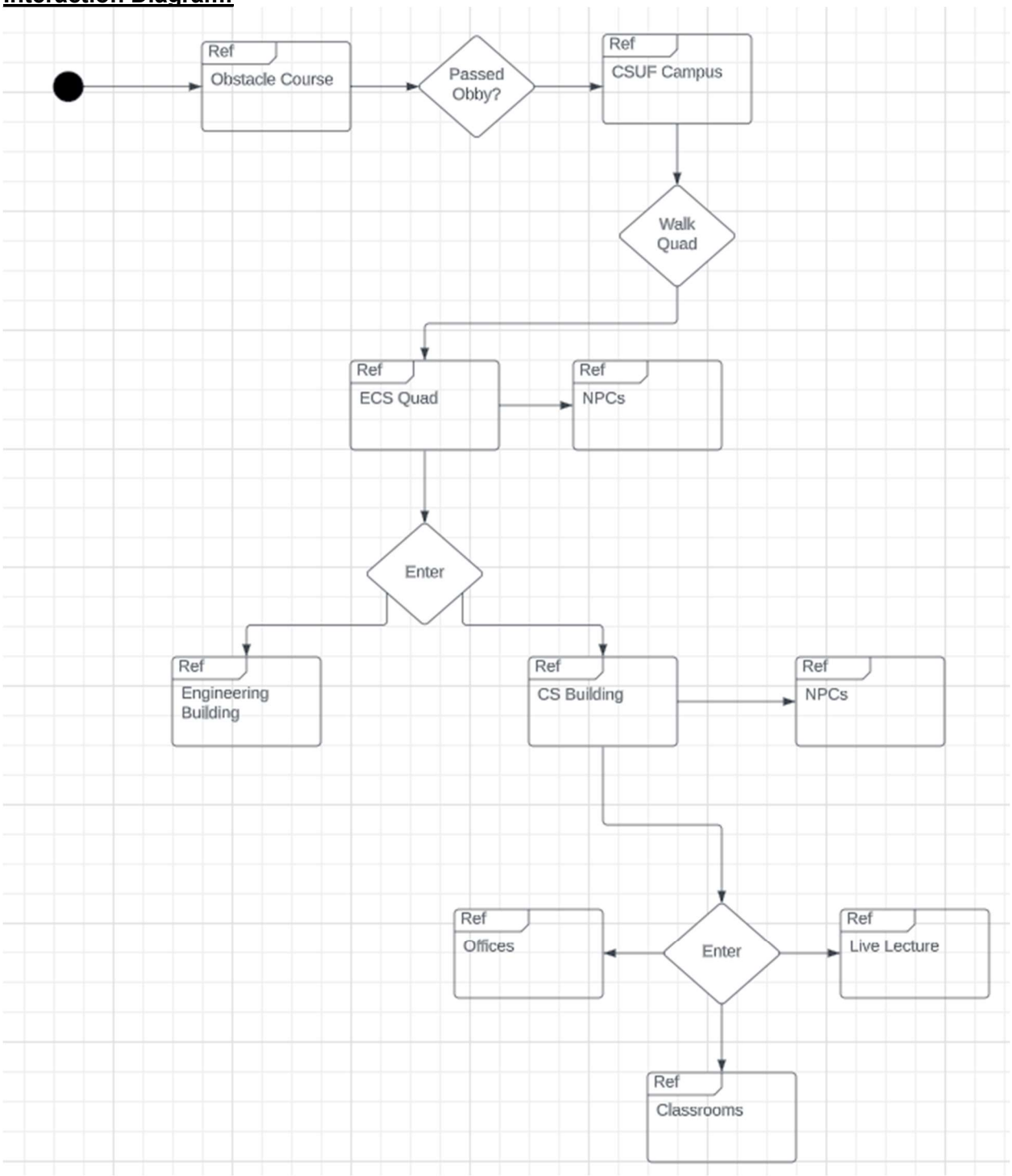
Detailed Class Diagram



Sequence Diagram



Interaction Diagram:



Construction

Architectural Design:

Software architectural preparation for the virtual tour of the Computer Science Department at California State University Fullerton involves a systematic software process to ensure the successful development of the Roblox game. The process can be outlined as follows:

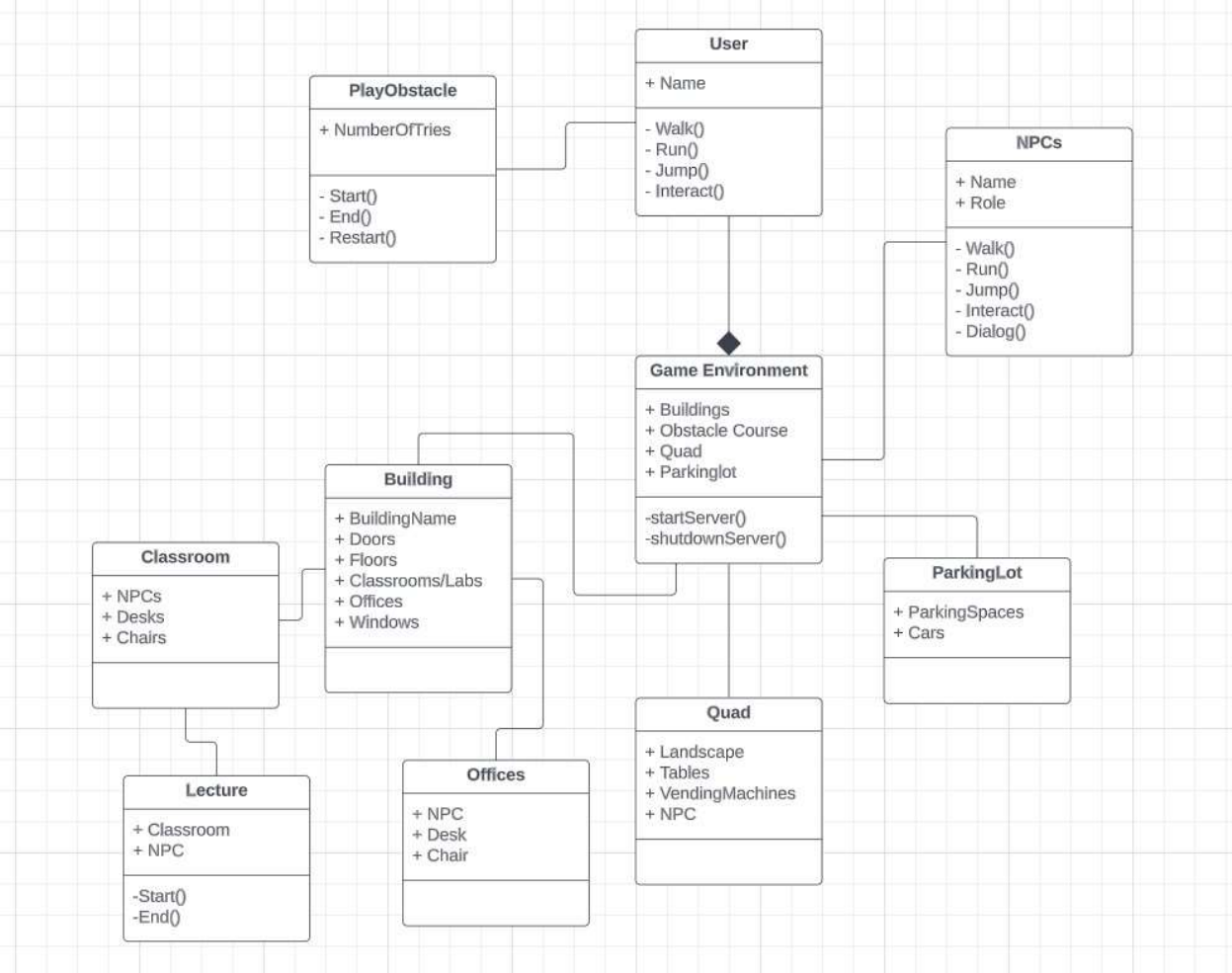
- Requirement Gathering: We gathered requirements for the virtual tour of our computer science department buildings, which includes simulating the building's interior, classrooms, live classrooms, hallways, offices, and common areas. Identify the need for interactive elements such as non-playable characters (NPCs), obstacle courses, spawn points, vehicles and information displays.
- Conceptual Design: Bryan Garcia and David Fazio developed the initial design concepts for the virtual tour game, considering the layout, aesthetics, and user experience concerning students/parents. To visualize the proposed design all members created the sketches, diagrams and 3D models.
- Technology Selection: Roblox Studio is the primary tool utilized for building the environment for the virtual tour.
- Architectural Design: All members contributed to designing the software architecture for the virtual tour game. We defined classes and their attributes for every detail in our game like the door, door handle, the NPCs, building parts, cars, spawn areas, Cal State Fullerton Logo and walking areas. We also defined the interactions between various elements such as NPCs, obstacle courses, and information displays.
- User Interface Design: Bryan Garcia and David Fazio along with the rest of the members designed the user interface for the virtual tour game, ensuring it is intuitive, visually appealing, and responsive. A lot of factors have been considered while designing like

information displays, navigation controls, and interaction mechanisms to enhance the user experience.

- Development and Testing: After implementing the architectural design and developing the game features using Roblox Studio all the members conducted regular testing to identify and fix any bugs or issues or missing information. We also considered the perspective of new students and their families. We shared the continuous feedback and noted it down on our discord group, for future reference and improvements.
- Simulation: We prepared the virtual tour game for deployment on the Roblox platform, ensuring all dependencies and requirements are met. We have focused more on creating a real-time simulation for our computer science department, playing and moving around the buildings, iterating with other real-time players, the player can enter the building, can enter a live classroom, where we have created a real-time classroom with the professor and students, components like chair, board and writings have also been displayed in our real-time virtual tour game.

By following this architectural design process, the team aims to create an immersive and interactive virtual tour game that effectively showcases the Computer Science Building and its surroundings. The game will serve as an informative experience for new students and their families, providing them with valuable insights into the faculty, resources, and facilities offered by the Computer Science Program at California State University Fullerton.

Class Diagram



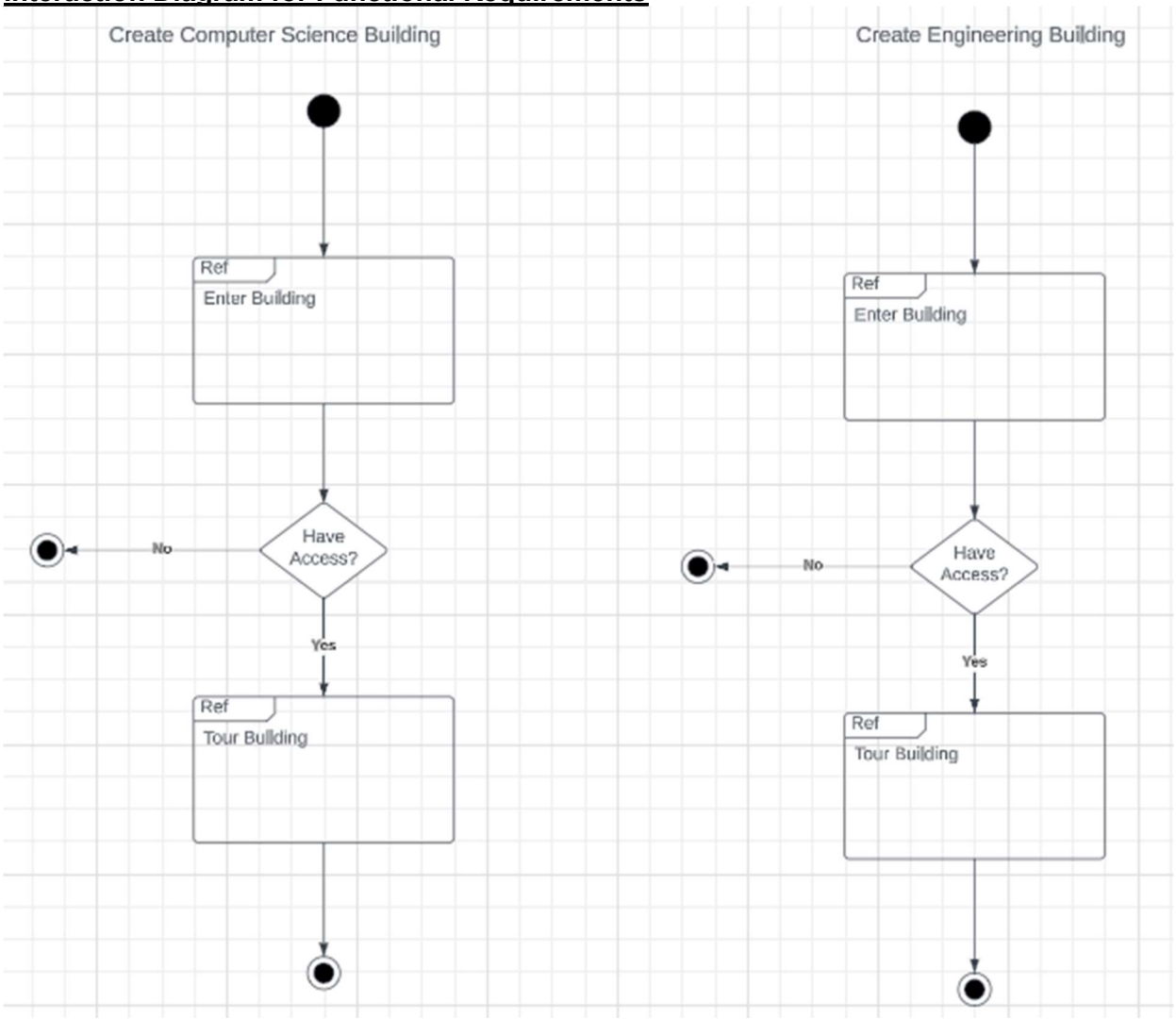
Transition:

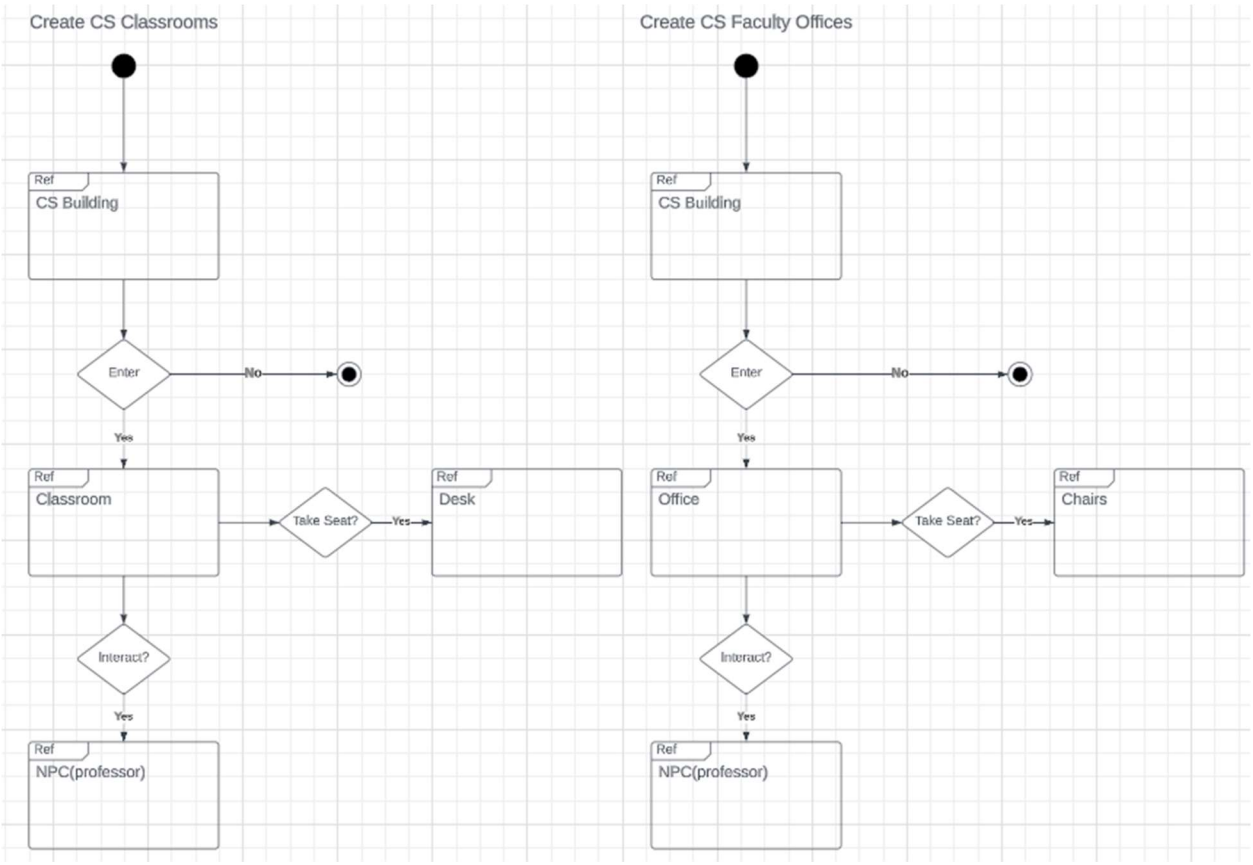
Transition Plan

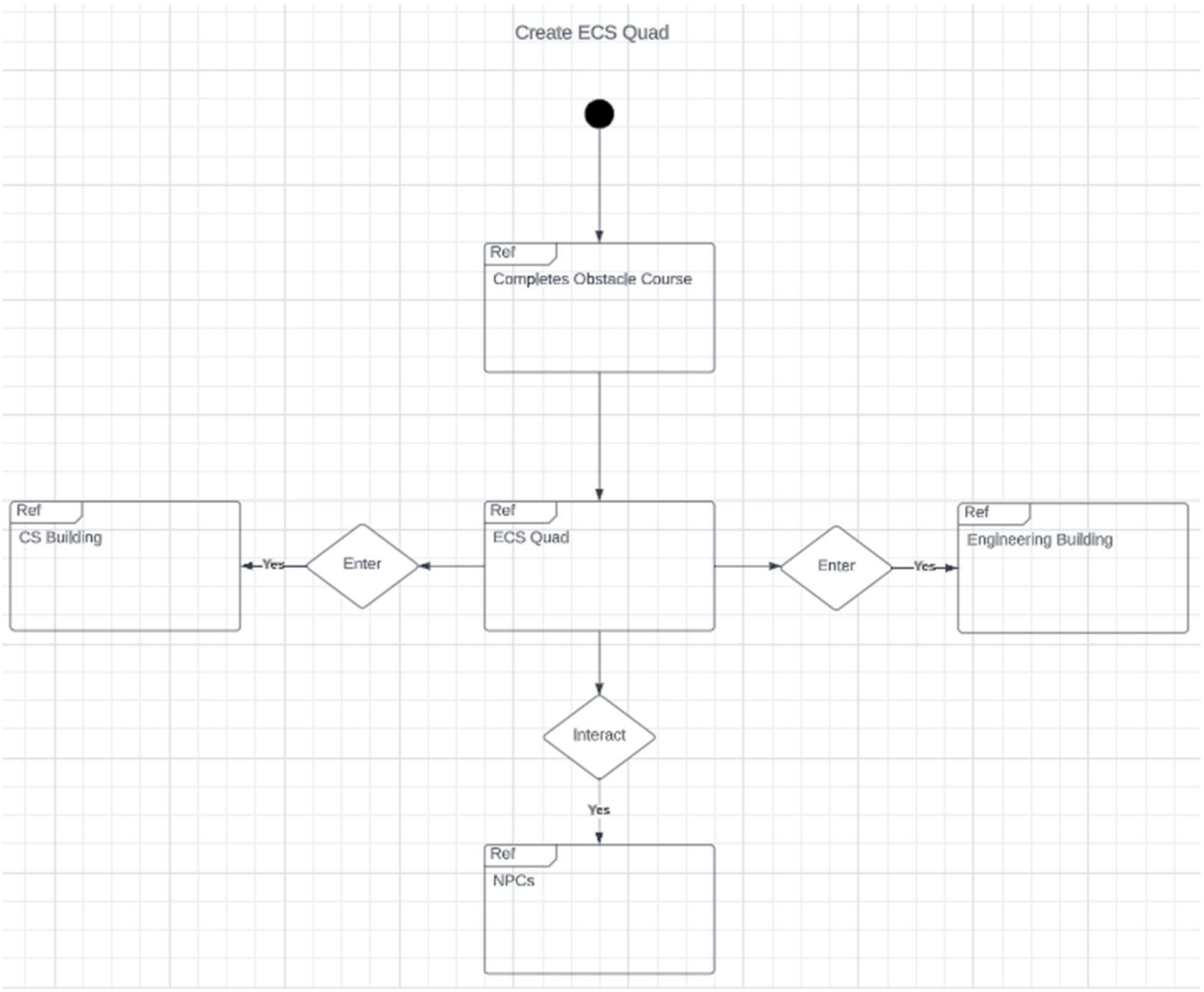
During the transition phase, our team will begin by exploring the campus environment we have created. The purpose is to compare it to the experience of being on an actual campus and identify any similarities or differences. We will then assess the performance of the program, particularly focusing on the geographically demanding areas of the campus. If any issues arise, we will plan further refinements for future iterations after deployment.

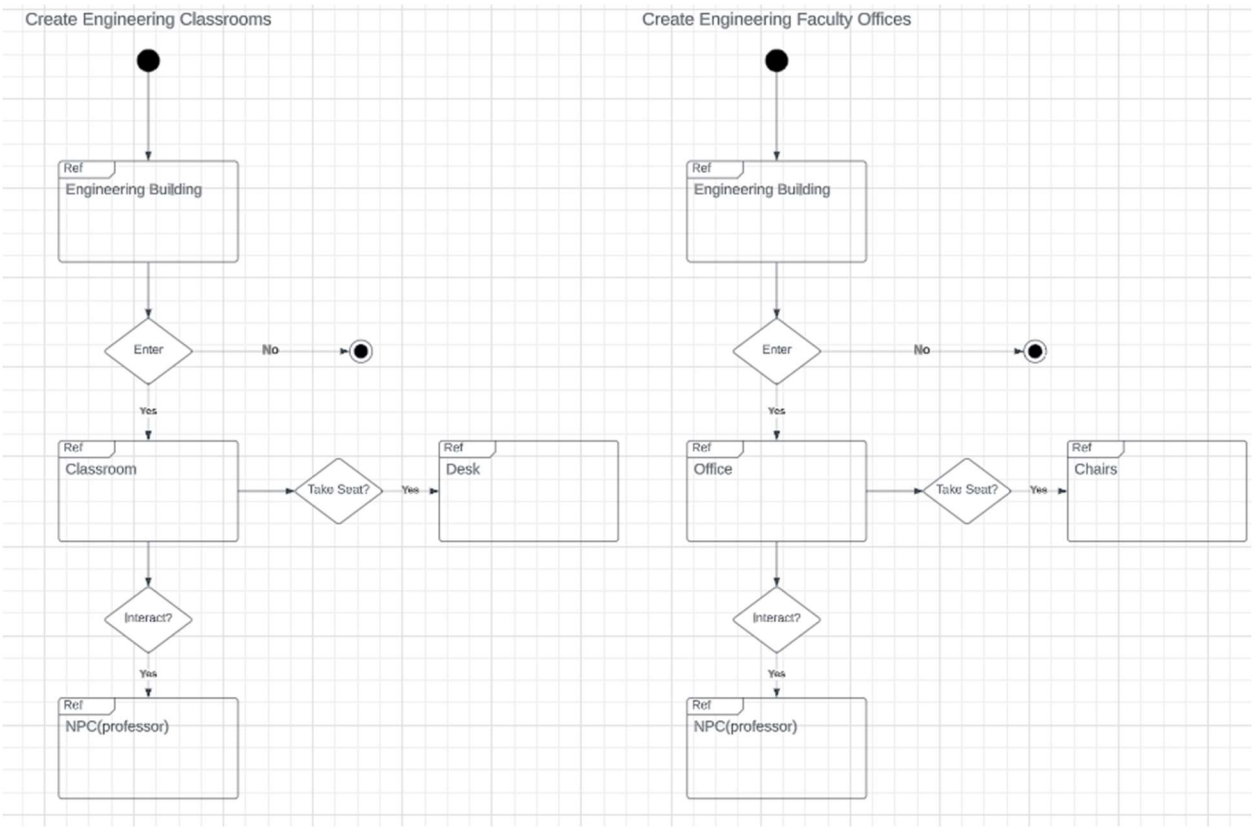
Additionally, we will evaluate the program's performance on mobile devices, although it is not our primary focus. We aim to make it accessible for mobile users, with plans for future iterations to improve performance on these devices. Our intention is to deploy the program to users who are not part of the development team and have them simulate their experience being in the Computer Science Department. This will provide us with valuable feedback to determine the future requirement and enhancements for our program.

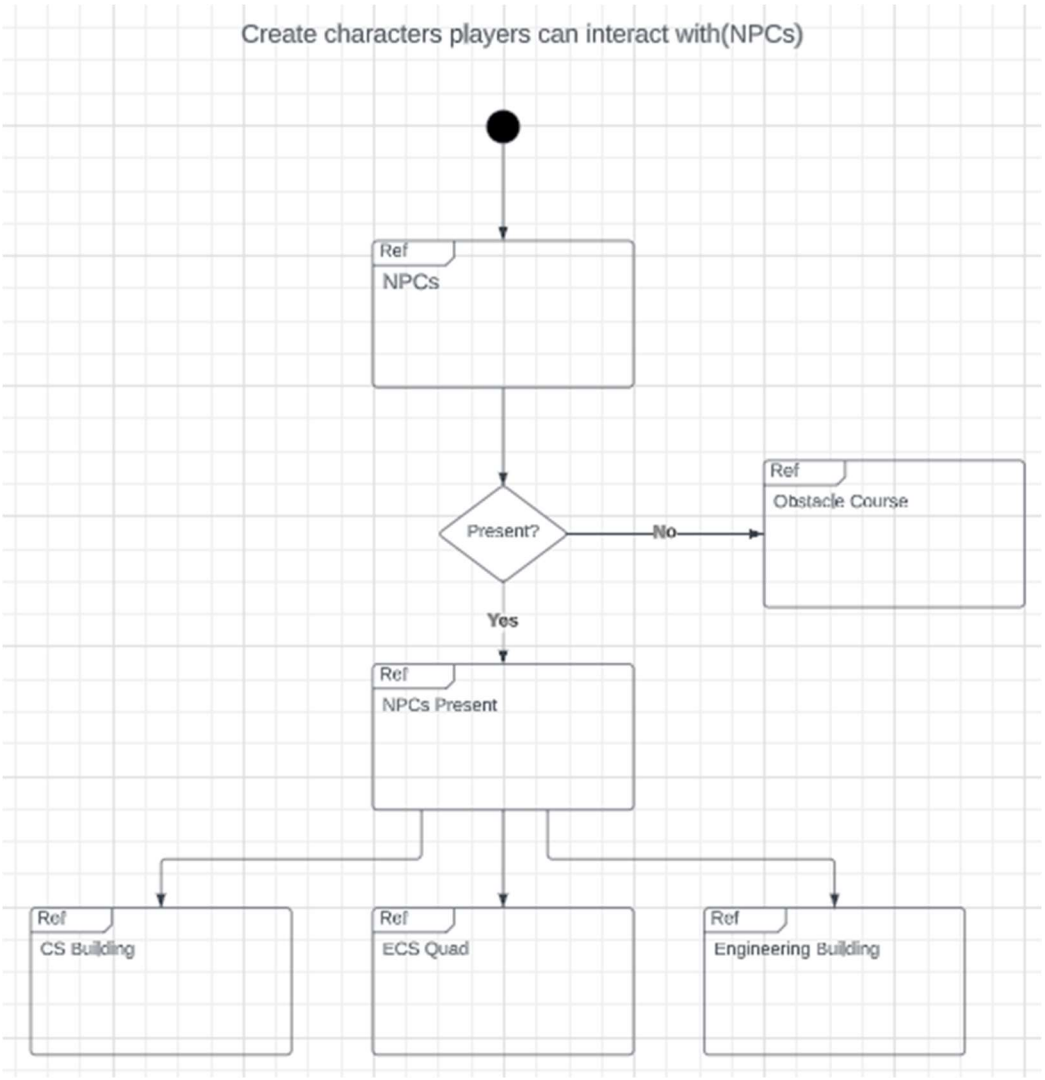
Interaction Diagram for Functional Requirements

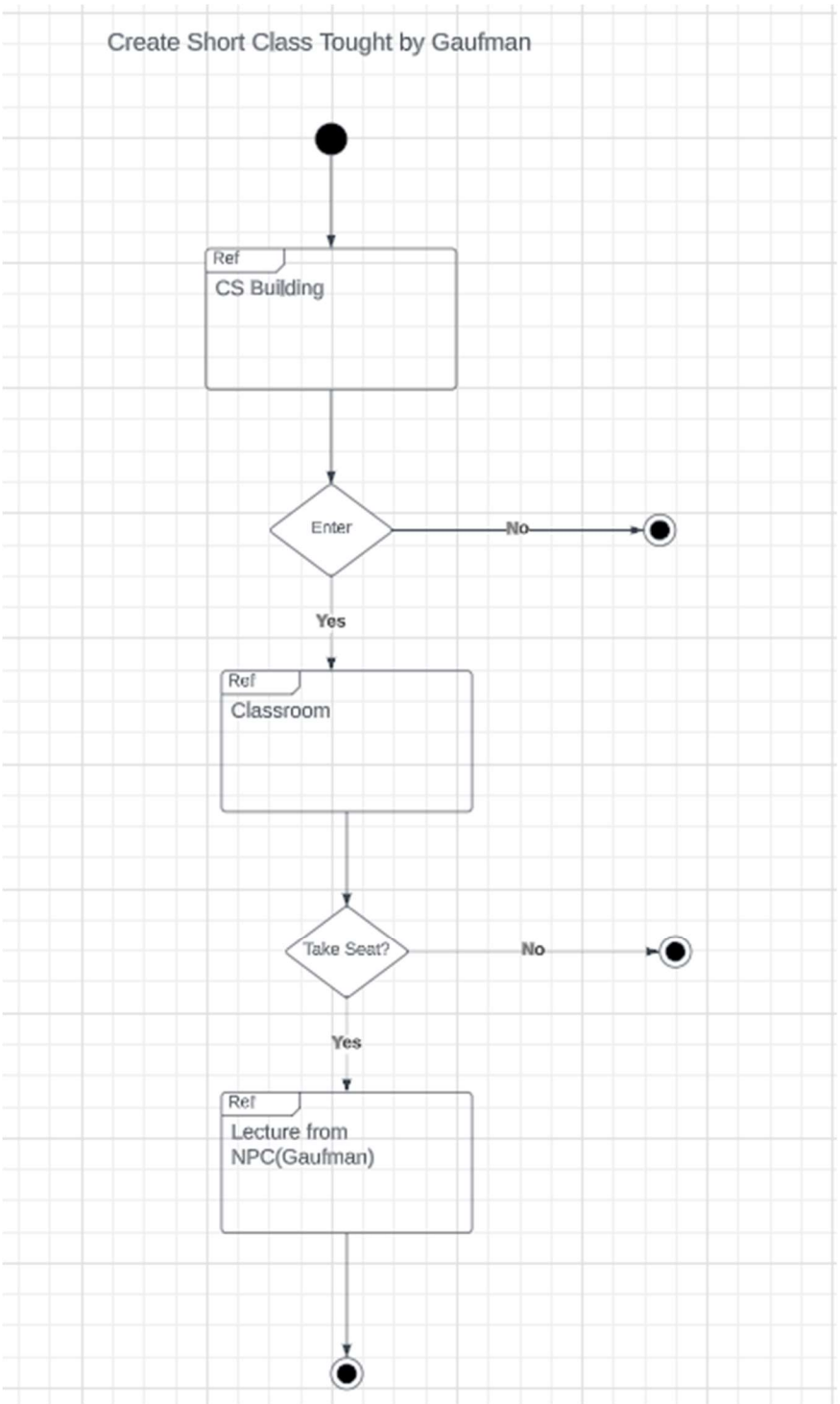


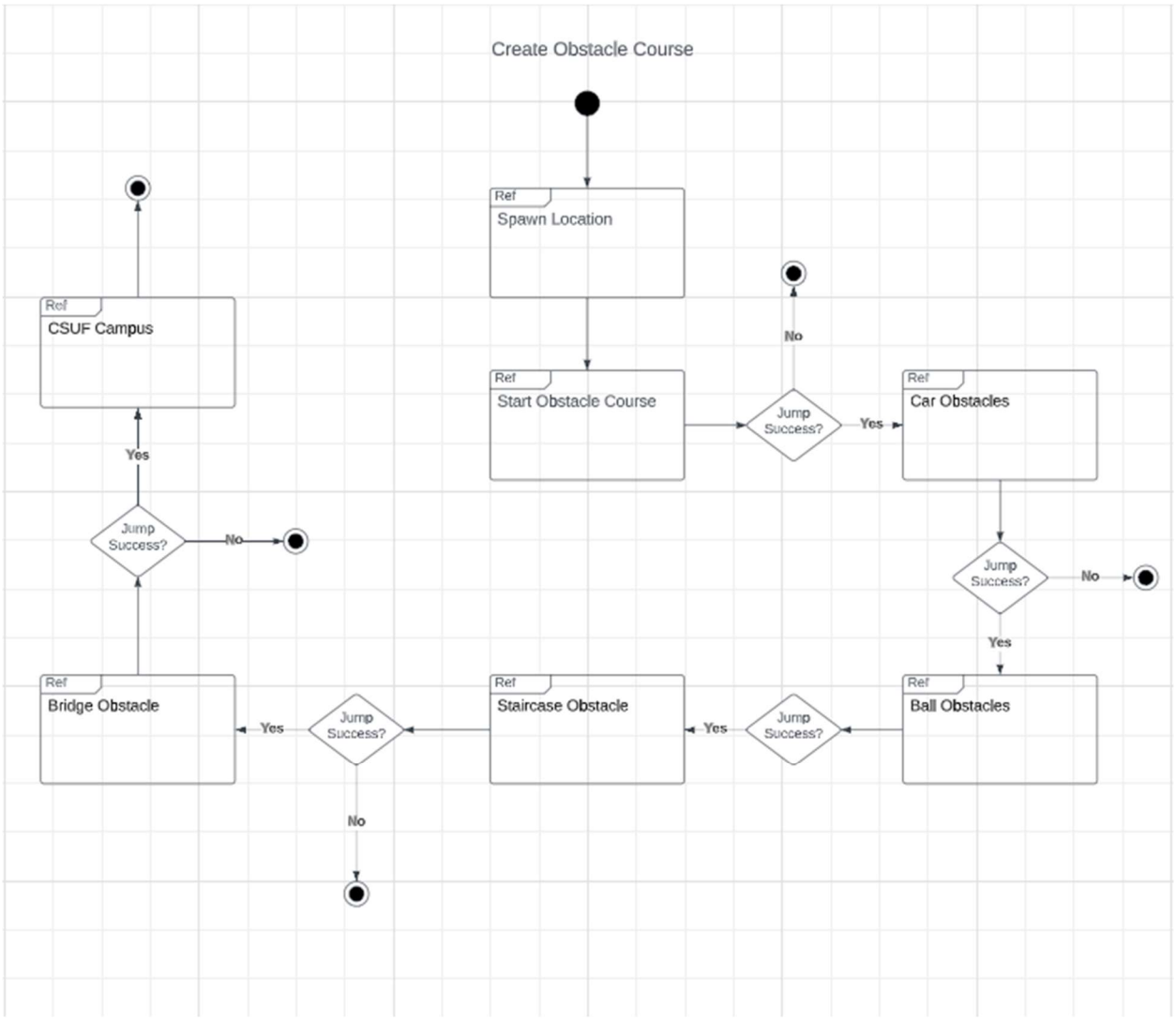

Create CS Classrooms
Ref
CS Building
Enter
No
Yes
Ref
Classroom
Take Seat?
Yes
Ref
Desk
Interact?
Yes
Ref
NPC(professor)
Create CS Faculty Offices
Ref
CS Building
Enter
No
Yes
Ref
Office
Take Seat?
Yes
Ref
Chairs
Interact?
Yes
Ref
NPC(professor)


Create ECS Quad
Ref
Completes Obstacle Course
Ref
CS Building
Yes
Enter
Ref
ECS Quad
Enter
Yes
Ref
Engineering Building
Interact
Yes
Ref
NPCs


Create Engineering Classrooms
Ref
Engineering Building
Enter
No
Yes
Ref
Classroom
Take Seat?
Yes
Ref
Desk
Interact?
Yes
Ref
NPC(professor)
Create Engineering Faculty Offices
Ref
Engineering Building
Enter
No
Yes
Ref
Office
Take Seat?
Yes
Ref
Chairs
Interact?
Yes
Ref
NPC(professor)


Create characters players can interact with(NPCs)
Ref
NPCs
Present?
No
Ref
Obstacle Course
Yes
Ref
NPCs Present
Ref
CS Building
Ref
ECS Quad
Ref
Engineering Building


Create Short Class Tought by Gaufman
Ref
CS Building
Enter
No
Yes
Ref
Classroom
Take Seat?
No
Yes
Ref
Lecture from
NPC(Gaufman)


Create Obstacle Course
Ref
Spawn Location
Ref
Start Obstacle Course
Jump Success?
No
Yes
Ref
Car Obstacles
Jump Success?
No
Yes
Ref
Ball Obstacles
Jump Success?
Yes
Ref
Staircase Obstacle
Jump Success?
Yes
No
Ref
Bridge Obstacle
Jump Success?
No
Yes
Ref
CSUF Campus
Yes

Part B

Scrum and Agile-UP have distinct characteristics and practices that can lead to different outcomes and project management. If we used Scrum, there would be different teams with each team representing a set of skills in order to complete the project. The teams would consist of at least 2 to 3 members with defined roles and objectives such as a Product Owner whose interest is to bring value to the customers. There are other roles such as Scrum Master and Development Team whose objective is to help with the Scrum process and building increments respectively. The main difference between Agile-UP and Scrum is that Agile-UP does not require structure based on the team members skills. It instead provides instructions and guidance on how to create a project based on the team.

Scrum involves improvement based on looking at what went wrong, feedback and what can be improved through retrospectives. Scrum would also involve a product backlog to keep track of the functionality and the features of the game based on the importance of the feature. As a result, the most important features would be worked on and be developed first in order to work with the Product Owner's objective on keeping the customers content on our work progress. The future of our project in Scrum is more in depth by providing us tools such as Daily Scrum Meetings. Scrum Meetings would enable more flexibility and easy to make changes with correlation of the Product Owner. To conclude, if we used Scrum instead of Agile-UP, our project would be more structured with specific objectives relating to the roles assigned to the member.

Part C
(for graduate student - Pratyush Sahu)

The development approach and results would probably be very different if the virtual tour Roblox metaverse project had been built using CMMI (Capability Maturity Model Integration) as compared to Agile. The goal of the CMMI approach to process improvement is to define and standardize processes in order to enhance both project management and product quality. The lightweight Agile methodology, on the other hand, encourages adaptability, flexibility, and customer collaboration throughout the development process.

Here are some possible changes and effects if the project utilized CMMI rather than Agile:

- ***Development Approach***: CMMI generally utilizes a plan-driven and more structured approach, in which development activities are carefully defined, scheduled, and documented before implementation. On the other hand, Agile adopts an incremental and iterative development strategy, enabling constant adjustment and flexibility throughout the project.
- ***Requirements management***: In terms of CMMI, requirements are explicitly stated, and any changes may call for formal change management procedures. In contrast, Agile emphasizes strong coordination with stakeholders and accepts evolving requirements in order to prioritize and modify requirements as development progresses.
- ***Iterations and Prototyping***: Agile supports iterative development and prototyping because they provide early feedback and continual improvement. With CMMI, there might be less of an emphasis on prototyping and more of a focus on adhering to established processes, which could cause feedback and corrections to be delayed.
- ***Customer Involvement***: Agile-UP places a strong emphasis on ongoing client input and feedback in order to guide the project in the proper direction. While taking customer needs into account, CMMI may not have the same frequency of contacts and feedback loops with stakeholders.
- ***Project documentation***: comprehensive documentation of processes and artifacts is extremely important in the CMMI process. While documentation is still provided in Agile , it tends to be

lighter and less formal with an emphasis on delivering functional software rather than in-depth documentation.

Agile would probably be a better strategy given the project's nature, where the objective is to provide an interactive and educational experience. It enables quicker development iterations, ongoing product improvement based on user feedback, and a stronger emphasis on client happiness. However, a hybrid approach that combines components of both techniques could also be taken into consideration, depending on the specific environment and needs of the project.

Team Charter:

Course Title	CPSC 466 (58/59) Software Process	*All team members participated in the creation of this charter and agree with its content.* ***Date*** *07/07/2023*
Instructor	Dr. Chang-Hyun Jo	
Course Dates	07/03/2023 – 08/04/2023	

Team Members (Contact Information)

Name	Address (city, state, country)	Phone	Cell	Email
David Fazio	Irvine, CA	xxx-xxx-xxxx	949 278 0045	davidfazio1102@csu.fullerton.edu
Juan Uriarte	Fontana, CA	xxx-xxx-xxxx	xxx-xxx-xxxx	uriarte.juan@csu.fullerton.com
Bryan Rivas	Norwalk, CA	xxx-xxx-xxxx	xxx-xxx-xxxx	brivas100@csu.fullerton.edu
Bryan Garcia	Garden Grove, CA	xxx-xxx-xxxx	714 603 5059	pieismanly@csu.fullerton.edu
Pratyush Sahu	Fullerton, CA	xxx-xxx-xxxx	657 514 3077	pratyush.sahu@csu.fullerton.edu

Team Member Skill Inventory

David Fazio	§ MS Word, Excel, PowerPoint § Project Management § Web programming, Database design and development § C++, C#, Python, SQL, HTML, CSS, Java

Juan Uriarte	§ MS Word Knowledge, Excel, PowerPoint § Project Management § Database Design and Development § Java, C++, Python, SQL
Bryan Rivas	§ MS Office Suite (Word, Excel, PowerPoint, Project) § Database § Programming Languages (C#, Java, C/C++, Assembly, python) § Linux § Mobile App Dev
Bryan Garcia	§ Database Design and Development § Systems Architecture, DODAF Specialist § Project Management § Numerous Computer Software and Hardware skills including Oracle, Peoplesoft, SQL, C, Microstrategy, Actuate, Informatica, Microsoft Window, Word, Excel, PowerPoint
Pratyush Sahu	§ 3 years of work experience in an MNC in India § Skills: NodeJS, ExpressJS, javascript, and java

Team Goals (Project goals, team process goals, quality goals, etc.)

§ Learn about Roblox, Use of Roblox Studio and Lua programing language

§ Learn about software processes and software design and architecture.

§ Learn Unified Process and be able to work as a team and develop a successful Roblox game

§ Acquire practical software engineering (people, process, tools, and methods) knowledge from teammates.

§ Maintain great relationships between teammates.

§ Produce and deliver Software Process detailed documentation for project

§ Produce Software Process diagrams (UML, Use Case, Class ect…)

§ Develop skills to facilitate future career goals by delegating team members into completing objectives.

Team Roles (Define roles of each member to achieve goals)

David Fazio Team Lead	§ Evaluating performance of team members, providing feedback, and identifying areas for improvement § Record tasks, including desired completion dates and responsible team members. § Document ideas and key points for future references. § Planning and organizing team activities, setting goals, and establishing timelines. § Able to provide guidance and expertise on related project material.

Juan Uriarte Team Assistant	§ Responsible for having objectives completed in a timely manner. § Ensure equal participation of all team members. § Distribute agendas to the team well in advance to provide input. § Maintain open communication with the professor/class. § Address and remove any barriers that could hinder teams progress.
Bryan Rivas Facilitator	§ Support the lead in preparing the agenda for collaboration sessions. § Ensure that collaboration sessions adhere to the agenda and stay within the allocated time frame. § Promote equal participation among teammates during sessions, ensuring that everyone has an opportunity to contribute. § Provide reminders of the team's progress and actively seek input from team members.
Bryan Garcia System Architect	§ Maintaining up to date overview of the project's technical aspects. § Comprehensive understanding of how different processes are interconnected. § Stay informed about the projects technical landscape. § Maintain a high level perspective of the project's technical components.
Pratyush Timekeeper	§ Communicate any necessary changes to scheduled collaboration sessions. § Ensure that the team stays on track and meets scheduled deadlines for tasks. § Review and assess the progress of tasks to identify any areas that require more attention § Monitor and track team's progress to ensure tasks are completed in a timely manner.

Ground Rules (Meeting schedule/locations, attendance expectations, agenda, assignment completion, communication methods, etc.)

§ All team members must communicate the progress or tasks they complete on the project.

§ All members are expected to work as team and to respect each other.

§ The team will discuss when they are able to meet throughout the week to discuss tasks that need to be completed for that week .

§ Team members will let the group know if they are not going to be able to attend a scheduled meeting or if any emergency has happened that will not let to be able to work on the project.

§ Team members should check the group chat at least once a day to stay on top of things.

§ Team members will give updates of the progress they are making on their particular task.

§ All team members must carry their weight and complete their portion of the work within set timeblocks.

§ Team members must participate on the group discord server and follow along with any updates or changes to project or team meetings. Try to respond within 24 hours at most.

§ All team members are expected to help one another.

§ All team members are expected to subscribe to a work culture that makes everyone feel accepted, valued, and a sense of belonging. No negativity or toxicity.

§ All team members are expected to participate in weekly team building exercises.

Time Commitments/Availability (Pacific Time)

David Fazio	§ Monday-Tuesday 5pm-7pm § W-F 5pm-2am § Weekends 11am-5pm

Juan Uriarte	§ M-W, 12pm-10pm § TH-F 8am-12pm § Sunday, 8am-10pm
Bryan Rivas	§ M-F 4 pm - 8 pm and 10 pm - Midnight § Saturday & Sunday 10 pm - Midnight (can make arrangements if meeting earlier is necessary)
Bryan Garcia	§ M & T 5:00pm - Midnight § W & TH All day § F - SUN 9:00pm - Midnight
Pratyush Sahu	§ M&W from 5:00 pm - Midnight § T&Th from 8:30 pm - Midnight § Between 6:00 am - 2:30 pm

Conflict Management (What are potential conflicts that might arise among or between team members during this course? How will team members deal with these and other conflicts?)

§ Assigning clear roles and responsibilities to each team member prevents confusion and minimizes conflicts.

§ If a team member is not performing up to expectations, the team can have a one on one conversation to identify the root cause of the problem.

§ Encourage team members to openly communicate their concerns or conflicts

§ Encourage team members to address conflicts within the group to prevent negatively impacting the team's project progress.

§ If conflicts persists, a neutral third party will be involved such as a Professor to help facilitate and find resolutions that satisfy both parties.

Risk Management (What are potential barriers to the achievement of these goals?)

§ List risks that are chances or possibilities of suffering loss or danger in the project.

§ Lose of team member - be able to divide up work if we lose a team member.

§ Team members not completing their portion of work - communicate with the rest of the team and be able to pick up the slack where needed.

§ Communication - have regular team meetings. Set clear goals, requirements, expectations and work expected from everyone.

§ Determine the likelihood or probability of a risk occurring based on available information.

§ Conduct a thorough analysis of potential risks by involving team members using techniques such as brainstorming and lessons learned from previous classes.

§ Develop a risk mitigation plan that outlines specific actions, responsibilities, and timelines for addressing certain risks.

§ Continuity monitor identified risks throughout the projects lifecycle

§ Identify the risks that have the potential to significantly impact the project's success if they occur such as focusing on risks that have a higher likelihood of happening or those with severe consequences.

Team Evaluation Criteria (List evaluation criteria that will be used to evaluate team members objectively.)

§ Conduct periodic evaluations at predefined intervals throughout the course or project duration.

§ Base evaluations on objective evidence such as team meeting logs, project documentation, and game progress.

§ Provide timely and constructive feedback to team members throughout the course or project

§ Highlight areas of strength and areas that need improvement

Team Evaluation:

Members / Evaluators	David Fazio	Bryan Garcia	Bryan Rivas	Juan Uriate	Pratyush Sahu	Total	Comments on Your Evaluation on Team
David Fazio	100	100	100	100	100	500	All members did well.
Bryan Garcia	100	100	100	100	100	500	All members did well.
Bryan Rivas	100	100	100	100	100	500	All members did well.
Juan Uriate	100	100	100	100	100	500	All members did well.
Pratyush Sahu	100	100	100	100	100	500	All members did well.
Total	500	500	500	500	500	2500	
Max	500	500	500	500	500	500	
Average	100.00	100.00	100.00	100.00	100.00	500.00	
Percent	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
Signature	DF	BG	BR	JS	PS		
Comments on Your Score Earned from Team							

References:

https://www.geeksforgeeks.org/capability-maturity-model-integration-cmmi/

https://www.geeksforgeeks.org/software-testing-requirement-traceability-matrix/

https://www.lucidchart.com/pages/uml-interaction-diagram

https://www.lucidchart.com/pages/uml-class-diagram

https://www.lucidchart.com/pages/uml-sequence-diagram