Requirements

Group Number: **Cohort 1, Group 11**
Group Name: **Y111 Studios**

Group Members:

1. James Hutchinson
2. Somto Igweonu
3. Robert Kisloski
4. Sam Knight
5. Kenneth Kok
6. Ashish Kumar

This document outlines the procedure followed in acquiring and presenting the user and system requirements for the development of our project, UniSim, a single-player simulation game that allows the player to build their own university campus.

Requirements Elicitation & Negotiation

The requirements gathering process began with brainstorming sessions among team members, with the aim of aligning our creative visions with the goals outlined in the product brief. To facilitate this, we conducted an interview with our client, which was crucial for understanding the overall vision for the game, its key features, and user interactions.

We began the interview by asking the client for their **Single Statement of Need (SSON)** for the product, which helped us become acquainted with the overall goal of the project.

SSON: "I'm looking for a short, fun, jovial game for young adults to play, to enjoy seeing the positive aspects of working in computer science."

With this, we were able to note the intended audience and use of the game:
Target Audience: Young adults (16-21).
Intended Use: A fun, entertaining game with a focus on making computer science appealing to the player in a more casual setting.

The interview then involved our client providing valuable input on the game's style and perspective, the platform and performance, and the gameplay elements, allowing us to ensure that our requirements align with the expectations of our stakeholder.

The negotiation phase involved us prioritising the requirements based on feasibility and importance, with the time constraint in mind. Our client's input was significant in shifting our focus towards creating a Minimum Viable Product (MVP) that delivered a simple yet engaging gameplay experience. For instance, while we were tempted by the idea of building multiple maps with varying difficulty levels, our client stressed the importance of starting with a single, well-designed map with a static geography, to allow us to collect feedback before exploring more complex systems.

Thus, this arrangement allowed us to effectively obtain and prioritise our requirements in a way that will guide us to build a game that will both meet the client's vision and be enjoyable for the players.

Requirements Presentation

We are presenting our user and system requirements in the form of structured tables that include the ID and description of each requirement. With this approach, we aim to create a clear roadmap for developing the game, allowing us to set goals and track our progress using this referencing system. The next steps will involve translating these requirements into the architecture and implementation phases.

<u>User Requirements:</u>

These requirements reflect the needs or expectations of the game's end-user. They focus on user interactions, outlining specific tasks that the players want to perform. The table below contains the IDs, descriptions and prioritisation of the user requirements of our game.

ID	Description	Priority
UR_STATIC_MAP	The game should have a single, static map with predefined characteristics.	High
UR_GAME_PERSPECTIVE	The game should have a top-down perspective for easy control and visibility.	High
UR_GAME_ENJOYABILITY	The game should have a lighthearted and positive atmosphere.	High
UR_BUILDING_PLACEMENT	The player must be able to place at least one of each type of building: teaching, accommodation, recreational and catering.	High
UR_BUILDING_MANAGER	The player should have access to a simple interface for managing buildings.	High
UR_GAME_CLOCK	The game should have a countdown timer displaying the remaining game duration.	High
UR_PAUSE_GAME	The player should be able to pause the game at any time during the 5-minute session.	Medium
UR_BUILDING_COUNTER	The game should display a simple building counter showing how many of each building have been placed so far.	High
UR_GAME_DIFFICULTY	The game should be simple enough for young adults to enjoy without needing advanced skills.	Medium
UR_MINIMUM_HARDWARE	The game should run on standard laptops and desktops without requiring high-end specifications.	High

Functional Requirements:

These requirements describe what the game must do, detailing the functions and features that it must support. The table below contains the IDs and descriptions of the functional requirements of our game, as well as their link to the user requirements.

ID	Description	User Requirements
FR_STATIC_MAP	The game shall provide a static map for the player to traverse	UR_STATIC_MAP
FR_GAME_CONTROL	The game shall provide controls and camera for the top down perspective	UR_GAME_PERSPECTIVE
FR_PLACE_BUILDING	The game shall check the collision map and determine if its valid to place a building	UR_PLACE_BUILDING
FR_BUILDING_MANAGER	The game shall provide UI and controls to manage buildings	UR_BUILDING_MANAGER
FR_GAME_PAUSE	The game shall pause the game on the player’s request	UR_GAME_PAUSE
FR_GAME_END	The game shall terminate and end the game when the 5 minute timer is up	UR_GAME_CLOCK
FR_BUILDING_COUNTER	Display a counter to show the player the number of each type of building placed	UR_BUILDING_COUNTER
FR_MINIMUM_HARDWARE	Optimise the game for standard laptops or desktops	UR_MINIMUM_HARDWARE

Non-Functional Requirements:

These requirements describe how the game performs a task rather than what it should do. They ensure that the game meets certain standards of performance, usability, and reliability. The table below contains the IDs, descriptions, links to the user requirements and the fit criteria of the non-functional requirements of our game.

ID	Description	User Requirements	Fit Criteria
NFR_GAME_ENJOYABILITY	The game should be maintain a positive, lighthearted atmosphere	UR_GAME_ENJOYABILITY	Response to the game atmosphere should be 70% positive
NFR_GAME_DIFFICULTY	The game should be accessible and straight forward, without requiring advanced skills	UR_GAME_DIFFICULTY	Player completion rate > 50%
NFR_MINIMUM_HARDWARE	Confirm the game can run smoothly on standard hardware configurations without performance issues	UR_MINIMUM_HARDWARE	Performance testing on average lab hardware, FPS > 30