Date: 28/10/24

UniSim

REQUIREMENTS

TEAM6 Game Studios

Work Undertaken by:

Thomas Koukouris
Adam Khan
Oliver Herron
Sam Jordan
Nathan Hopper
Fergus Irvine

SINGLE STATEMENT OF NEED

UniSim shall be a fun and simple to play game while offering enough depth to challenge experienced players. The game should be intuitive to learn, immersive and engaging so that it encourages repeat play.

DOCUMENT PURPOSE

This document provides a detailed description of the game UniSim. It outlines user needs and system requirements, which are split under functional and non-functional requirements. The document will serve as guide throughout the development stages of the game and shall be regularly revised to meet updated client requirements.

REQUIREMENTS ELICITATION & NEGOTIATION & PRESENTATION

The initial phase of requirements elicitation for the UniSim game began with the client's **product brief**, which outlined the core elements of the game. Following this, Team6 had a group meeting where the project's requirements were discussed and a list of questions was developed. A subsequent **meeting with the client** took place a few days later and further details and clarifications were acquired. Afterwards, Team6 had a second meeting to finalise the complete set of requirements. This document is the consolidation of these meetings and will serve as the foundation of the game development. During a **second client meeting**, the client confirmed these requirements, some were revised/removed. We decided to present our set of requirements in a **tabular format** to allow for quick traceability and referencing but also to be able to update them easily as needed. By presenting them clearly and concisely, we are aiming to avoid misunderstandings of the requirements by developers.

GAME OVERVIEW

UniSim is a **single player simulation game** where players are tasked with building and managing a university campus. The goal of the game is to achieve maximum **student satisfaction score** by the end of the time.

The player begins with an empty map and are offered options to construct several buildings which cannot be altered after. Once the game begins, the player must use their resources available to them wisely to achieve the best student satisfaction possible by striking a balance between different factors.

The player is presented with special events that occur randomly. The events may be positive, negative or have no impact based on the players reaction (or non-reaction). The player must decide how to allocate and acquire resources during the game to achieve the best outcome.

The game has a time limit of **5 minutes**, then the game stops and the score is displayed as well as a complementary message reflecting performance.

It is assumed that players will have basic familiarity with simulation games and how game interfaces work. A short tutorial or interface explanations should be provided highlighting certain aspects of the game.

GAME PURPOSE & TARGET AUDIENCE

The aim is to provide an engaging game that will attract players primarily in the ages of **16-20+**. The game should encourage players to engage with Computer Science by highlighting that it was developed by CS students. This serves as a motivational element to inspire interest in the field. This game will serve as the initial minimal viable product (MVP). Depending on its success, the client is interested in further development of the game.

USER REQUIREMENTS (URs)		
ID	Description	Priority
UR_START_GAME	The player shall be able to start the game by pressing a button. They shall be offered options such as difficulty options and settings.	Shall
UR_PAUSE_GAME	The player shall be able to pause the game.	Shall
UR_MAIN_MENU	When the game is paused, a main menu shall appear with key functions, settings, and an exit button. Important game information shall be easily accessible. Buttons shall be visible, with text and interactive elements at least 13pt for readability.	Shall
UR_MAP	The player shall be able to view a 2D map representing the university campus and the buildings placed on it.	Shall
UR_BUILDINGS	The player shall be offered a list of placeable buildings in the main interface to choose from.	Shall
UR_BUILDINGS_LEARNING	The player shall be able to place at least one building for students to attend courses.	Shall
UR_BUILDINGS_LIVING	The player shall be able to place at least one building that students can use to live and sleep.	Shall
UR_BUILDINGS_RECREATION	The player shall be able to place at least two buildings that students can use for recreation.	Shall
UR_BUILDINGS_FOOD	The player shall be able to place at least one building that students can use to eat.	Shall
UR_BUILDINGS_PREVIEW	The player may be offered a building preview on the map before placing a building.	May
UR_GAME_INTUITIVENESS	The game shall be intuitive to play, ensuring that the player can easily understand and navigate the interface with minimal explanations.	Shall
UR_EVENTS_RANDOM	The player shall be offered actions to handle events as they occur randomly.	Shall
UR_EVENTS_PLANNED	The player shall be offered options to organise planned events.	Shall
UR_EVENTS	The player shall be able to respond to at least 3 events during the course of one game.	Shall
UR_EVENT_ALERTS	The player shall be alerted of events at predetermined times throughout the game and may be prompted to make a decision that could impact the game in some way.	Shall
UR_EVENT_MAP_DISPLAY	The player should be able to view events taking place on the game map. Each event's impact on the student satisfaction score should be clearly displayed to the player.	Should
UR_TUTORIAL	The player should be provided with a short tutorial that explains the map's features and game mechanics. This can be in the form of "hover-over" explanations of interface elements.	Should
UR_TIMER	The game shall display a countdown timer from 5 to 0 minutes.	Shall
UR_COUNTER	The player shall be able to track how many buildings are placed on the map at any time.	Shall

UR_LANGUAGE	The language used in the game shall be easily understandable and not technical.	Shall
UR_COUNTER_LIMIT	The player shall be made aware when they have reached the limit for the number of each building type.	Shall
UR_RESOURCES	The player may be able to spend their resources such as game currency on game elements, such as buildings, event handling and more.	May
UR_RESOURCE_ALLOCATION	The player may be shown the resource cost (resources to be decided) for each game element, such as building construction, building upgrades and maintenance, and event handling.	May
UR_PERFORMANCE	The player should be given feedback based on the appropriateness of their response to events. Also, they should be made aware when they have achieved some key milestones such as highest student satisfaction score or maximum number of students achieved.	Should
UR_BACKGROUND_MUSIC	The player may have the option to listen to background music while playing the game.	May
UR_GAME_END	An ending screen shall be displayed to the player when the timer is over. The player shall be offered the option to replay the game once finished	Shall

SYSTEM REQUIREMENTS

❖ FUNCTIONAL REQUIREMENTS (FRs)

ID	Description	User Requirements
FR_SOUND_VISUAL_EFFECTS	Map interaction, button presses, choosing an event action should give visual and audio feedback.	UR_INTERACTION
FR_NOTIFICATIONS	The system should provide notifications for events taking place, student satisfaction being too low and time remaining.	UR_EVENT_ALERTS
FR_DIFFICULTY	The game should have distinct, multiple difficulty levels. Harder mode shall deduct more resources for buildings and events shall occur more frequently.	UR_START_GAME
FR_MAP_MOVEMENT	The system shall allow movement in all directions within map limits.	UR_MAP
FR_ZOOMING	The system should allow for zooming in and out of the map.	UR_MAP
FR_SATISFACTION_SCORE	The student satisfaction score shall update in real time based on building types, events, building proximity and nature placement.	UR_PERFORMANCE
FR_EVENT_IMPACT	The system shall respond to the player's actions on events by adjusting the satisfaction score.	UR_EVENTS
FR_BUILDING_RESOURCE	The system shall deduct money from the player when a building is placed.	UR_BUILDINGS

FR_MAINTENANCE	The system shall display a countdown timer for each building type to notify the player when maintenance is due.	UR_RESOURCE_ALLOCATION
FR_BUILDING_MAINTENANCE	The system shall reduce money from the player's resources when building maintenance is done.	UR_RESOURCE_ALLOCATION
FR_TIMER	The system shall initiate a 5-minute timer when a game starts. It should also pause when the game is paused.	UR_TIMER
FR_NATURE	Nature elements of the game (placeable and not) shall have an impact (positive/negative) to the student satisfaction score.	UR_MAP
FR_NATURE_TREES	Trees may be removable for building pavement.	UR_MAP

❖ **NON-FUNCTIONAL REQUIREMENTS (NFRs)**			
ID	Description	User Requirements	Fit Criteria
NFR_EVENT_IMPACT	The system shall respond to player actions on events with initiating related events modifying resources.	UR_EVENTS	System shall respond within 1 second.
NFR_MAP_GENERATION	The system shall generate the map within a few seconds.	UR_MAP	Within 5 seconds
NFR_RESPONSIVENESS	The system shall provide instant feedback on button presses	UR_INTERACTION	Within 1 second
NFR_GAME_END_TIMER	The system must terminate the game as soon as the timer reaches 0.	UR_GAME_END	Within 1 second
NFR_SCALABILITY	The game code shall be modularised with clear documentation provided to support future expansions.	UR_START_GAME	Clear, concise docstrings and comments
NFR_RELIABILITY	The game shall not crash.	UR_START_GAME	If so, it should do it gracefully and provide options.
NFR_SCORE	The system shall compute the score based on buildings (proximity, type, maintenance), nature placement and event handling.	UR_PERFORMANCE	Must test edge cases to ensure proper score calculations.
NFR_COMPATIBILITY	The game shall be compatible with the latest version of Windows and macOS	UR_START_GAME	Windows 10+ and macOS Ventura+

REFERENCES

Guides on information to include and format:

https://www.designrush.com/agency/software-development/trends/software-requirements-specification

https://asana.com/resources/software-requirement-document-template

Guide on distinction between functional/non-functional requirements:

https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/