

# Requirements

Group: Group 10

Names:

Haaris Altaf

Casper Co

Will Garavelli

Kamso Osuji

Krishna Rajamannar

Diyar Savda

Sam Storrs

## Single Statement of Need

The system shall be a single-player game that allows the player to escape from a maze representing university life, its complexity, and annoyances.

## Requirements Introduction

For the project, the requirements were primarily elicited and negotiated through our meeting with the customer, who is interested in marketing and eventually selling the game. The customer himself has stated that this will be the first version of the game, just to gauge interest. The purpose of the project is to design a maze game to meet the specifications of the customer involved. Aside from the meeting itself, the requirements were also formed through the product brief given. During the actual meeting, the customer informed us that we have creative control over the specifics for the project. Despite that, the customer did outline some requirements of his own, which included the target audience, the operating system and device on which the game can be played as examples. The customer also mentioned how we should prioritise the requirements that are vital to the project over others and complete other requirements only if we have time. From this, we were able to create a succinct set of achievable requirements that are set out in a table for each requirement to ensure that they are all met in their entirety, with identifying numbers that will be referred to in our documentation. Some requirements consist of multiple sub-requirements, which are indicated with decimal points in their reference number. We've also included an importance rating (Must, Should, May) for each user requirement, thus allowing us to prioritise which requirement we should focus on completing first, as mentioned by the customer.

The requirements indicated in the table were taken from the customer, but were also taken from the requirements research that we conducted, mainly by looking at lectures. In particular, we looked at the Requirements Engineering lecture from Week 2 for the formatting of the requirements. From that lecture, we replicated the structure of the user requirements table and the system requirements table (divided into functional and non-functional requirements), which can be seen below in the listed user and system requirements for the project. We used this approach to dictate our requirements as it means that we can clearly see the requirements themselves, but we can also see the effect that these requirements have on the system. We can easily use these requirements to differentiate between what the system should do and how the system should do it (i.e., its behaviour).

## User requirements table

<b>ID:</b>	<b>Description:</b>	<b>Priority:</b>
University_Maze_Game_	The game must be in a maze format where the goal is to escape from university; this is to be a single-player game.	Must
Target_Demographic_	The game is targeted towards members of the university, whether that be students (including past students) or members of the faculty, which means there should be recognisable parts of campus and can be specific references such as animals or people on campus. One of these must be the dean who is chasing the player around the maze. The aesthetic can be serious or jovial in the player's interaction with the maze. The game must be family-friendly.	Must
Time_	The game must have a 5-minute timer, but the aim is to complete the game in the fastest time possible. Thus, less than 5 minutes should be spent.	Must
Effects_	There will be items throughout the maze that can be found to hinder or help the player; at least 3 must be hidden and at least 3 must be visible.	Must
Winning_and_Losing_	The game is won by escaping the maze in under 5 minutes. It can be lost by taking over 5 minutes or by being "caught by the dean", which entails being at a certain part of the maze slower than is allowed.	Must
Pausing_	The game is aimed at a casual audience, which means that it should be able to be paused at any time, stopping the timer from continuing later.	Should
Sound_	There should be a sound/music track to go alongside the play to keep the user immersed. This must follow copyright and licensing laws and regulations.	May

## Functional requirements table

ID:	Description:	User Requirement:
Game Engine	A game engine will be used with Java to allow for a playable game without excessive redundant code. This means that the game will take up less storage space, which is better for the consumer.	University_Maze Game_
Powerups	There will be 5 power-ups throughout the map to help the player continue or increase their score. These must match the seriousness of the game, such that a game which has a silly aesthetic mustn't have power-ups that are too serious and vice versa.	Effects_
Debuffs	There will be 3 debuffs throughout the map to hinder the action of the player or lower their score. It follows the same rules as powerups, as referenced in 3.1 - it must be in line with the style.	Effects_
Score	The faster you complete the game, the higher your base score. This base score can then be influenced by the events in the game, some of which will clearly affect the score, e.g., a door that lowers the score but provides a more efficient route. But others can be less obvious e.g., pop-ups that cause challenges which, upon completion, will increase the plate's score.	Winning_and_Losing_
Score Display	The score will be visible to the player at the end of the game. Because the longer you spend, the worse your score, the score will gradually tick down as you play. It must also be clear that their score has been changed when it occurs because of an in-game event rather than the ticking down from time, so that the player can learn which actions have positive and negative impacts on their score should be displayed through the increase and decrease of the time remaining.	Winning_and_Losing_
Play Again	To encourage user retention in the game, it should be easy for the user to play the game again once they have either won or lost. This can be done in the form of a button, which will appear upon completion of an attempt at escaping the maze.	Winning_and_Losing_

## Non-functional requirements table

<b>ID:</b>	<b>Description:</b>	<b>User Requirement:</b>	<b>Fit Criteria:</b>
Map	The game's playing area will be split into sections, each representing a level and containing different, hidden obstacles.	University_Maze_Game_	At least 3 sections of the map can be distinguished from one another. E.g., the different colleges on campus.
Specifications	The game must be playable on a standard desktop Windows computer so that the majority of the target demographic can play without any specialised hardware. It should be flexible enough to be rolled out across any major operating system.	Target_Demographic_	A player on a Windows desktop should be able to download and play the game without specialised hardware.
Storyline	The game must be relatable to a student's interaction with uni-life, but it can do so in a variety of ways. As referenced in 1.4, the game can be jovial or serious, as different people will have had other experiences at university, but some struggles will be similar or the same.	University_Maze_Game_	All events in the game must be recognisable by students of the university.
Difficulty	The game must be accessible to the greatest number of people possible, which means that, for its first instance, there must be a baseline/easy difficulty that doesn't require the user to have years of experience gaming to pick up and try to escape the maze. If the game were to continue, there would be a more difficult edition.	Winning_and_Losing_	A student should be able to escape the maze successfully within 10 attempts, having never played the game.
Controls	The control must be whatever is easiest and intuitive for the user. This could be through mouse movement or keyboard and should be determined through testing.	Winning_and_Losing_	A student should be able to move around the map on their first attempt at the game.