Product Brief: Escape from Uni

You wake up surrounded by maze-like walls. Have you woken up? You are pretty sure that university was only meant to be an unnavigable maze, metaphorically speaking… only one way to find out!

You are to build a single-player game that allows the player to escape from a maze that represents university life, its complexity and annoyances. Specific features include:

- You will be constructing a university-like maze from scratch, placing various visible and hidden obstacles along the way.
- The game will last 5 minutes (you need to escape in that timeframe), but will start paused and will allow you to pause at any time.
- Your primary objectives are:
    - Ensure you escape in time! No one wants to be trapped in uni forever (or do they?)
    - Ensure you take an effective path to escape, many challenges and rewards are found ahead, some unseen.
    - Run! The dean is coming!
- Your game must contain:
    - A map containing the maze that the player is escaping from. This map should evoke university-like qualities/places/scenery but does not necessarily need to be realistic, it should make sense and have clear limits to where the player can go (or at least, attempt to go!).
    - At least five visible events that will hinder the player from progressing. For example this can be a door needing a key, a ~~monster~~ professor blocking the way, etc. An event does not necessarily have to only appear in one location, for example a door with a key will clearly need to have at least two, one for the door and one for its corresponding key.
    - At least three visible events that will benefit the player if they pass through them. For example this can be a temporary boost to speed, or an immunity to the next encounter etc.
    - At least three hidden events (invisible until triggered) with some fun interactions.
- You ‘win’ the game by achieving the best score:
    - Escaping as fast as possible will give you a better base score.
    - Solving events may increase/reduce the score. Some events may have clear indications that they will reduce the score (but may offer a more direct route to the exit). Some events may offer an increase in score, for some complication (i.e. moving the exit to another location).
- While the game lasts up to 5 minutes, it is aimed at a casual audience so can be paused at any time (at least in the base/easy difficulty).

Constraints

You are building a game that should be playable and enjoyable by your cohort. However, there is a stakeholder that you must also accommodate.

The customer: one of your lecturers will play the role of a customer who is interested in eventually trying to market and sell your game. Ultimately the customer is the person you must convince of the validity of your assumptions and decisions. This stakeholder can be contacted as often as you need and at any time (but do not expect an instant reply!).

Use of 3rd-Party Libraries, Tools and Assets

You are encouraged to use appropriately-licenced (see the Intellectual Property lecture for details) 3rd-party libraries (e.g. a game engine like libGDX or jMonkeyEngine, a JSON/object mapper library if you need to store structured data), tools (e.g. graphical level/map editors) and assets (e.g. images, textures) where possible, instead of implementing everything from scratch.

Assessment 1

For Assessment 1, you are only required to implement one of each type of event (one negative, one positive and one hidden), a tracker of the time the game lasts (up to 5 real-world minutes), and a simple counter denoting how many of each event have been interacted with.

Assessment 2

For Assessment 2, you are required to implement the full product brief.