ROTAT-A-BOT

Game Design Document

Daniel Cutajar & Dante Camenzuli

6/7/2019

ROTAT-A-BOT

The scope of the project is to produce a 3-level prototype conveying an easy level, a medium level and a hard level. The game will be akin to the likes of Topsy Turvy and Rotate in its puzzle style. The game will appear similar in functionality but in terms of design and layout will be original.

The game is aimed to be published for desktop computers. With the access to a PC and Mac both will be catered for in release. Resolution will be 16x10 across all devices.

Input methods will come in the form of the keys entered via the keyboard along with the mouse. The mouse will be used to navigate the UI in the start menu and the level select screen. When a level is loaded up control is switched to the keyboard.

CONTROLS

The game's main mechanic is the ability to rotate the entire level by 90 degrees, allowing the player to reach difficult locations as well as manipulate enemies into trapping themselves or even open doors for the player.

Movement is controlled by the arrow keys and SPACE to jump. The C key rotates the level to the left whilst the V key rotates the level to the right.
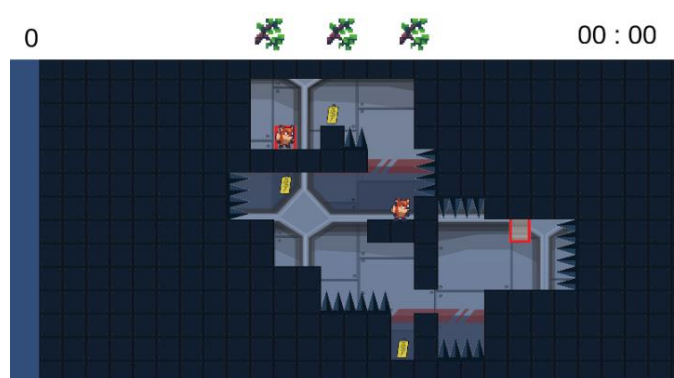
Several pickups are scattered around the level for the player to retrieve for extra points but are not necessary to complete the level. The faster the player reaches the EXIT door, the more points they'll achieve.
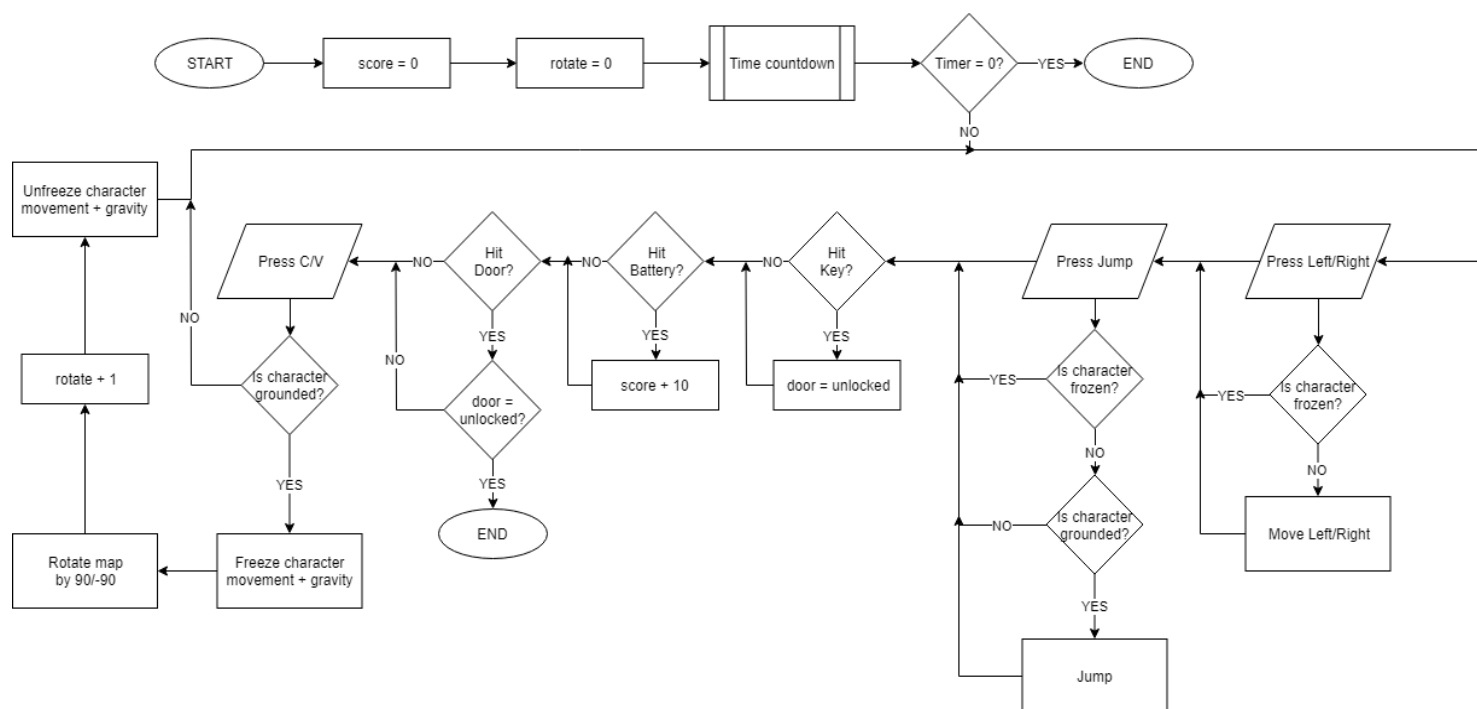
GAME SCREENS

There is a total of three main game screens:

- Title screen – the first screen the player sees and thus introduced to the game
- Game screen – iterative code is used to update this screen, in which the player actually plays the game
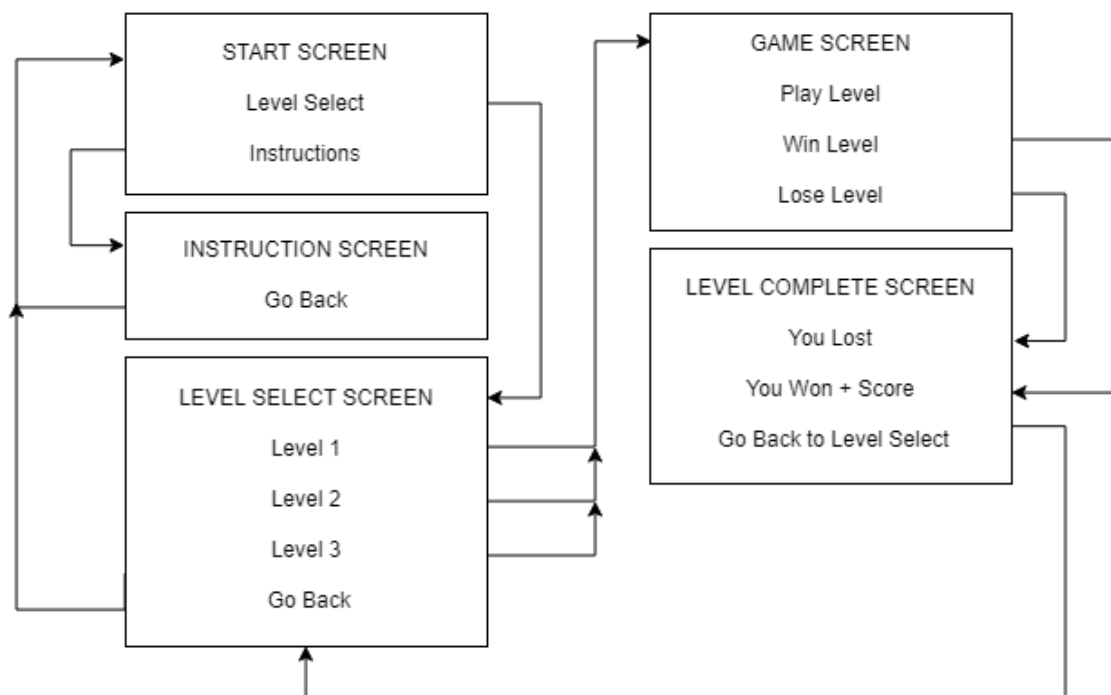- Game over screen – final screen if the player runs out of lives.

Below are some previews of the screens still in development

FLOWCHARTS

Flowchart

START → score = 0 → rotate = 0 → Time countdown → Timer = 0? —YES→ END

Timer = 0? —NO→ (to main loop)

Press Left/Right → Is character frozen? —YES→ (loop)
Is character frozen? —NO→ Move Left/Right

Press Jump → Is character frozen? —YES→ (loop)
Is character frozen? —NO→ Is character grounded? —NO→ (loop)
Is character grounded? —YES→ Jump

Hit Key? —YES→ door = unlocked
Hit Key? —NO→ (to Press Jump)

Hit Battery? —YES→ score + 10
Hit Battery? —NO→ (to Hit Key?)

Hit Door? —YES→ door = unlocked? —YES→ END
Hit Door? —NO→ (to Press C/V)

Press C/V —NO→ (loop)
Press C/V → Is character grounded? —YES→ Freeze character movement + gravity → Rotate map by 90/-90 → rotate + 1 → Unfreeze character movement + gravity

---

NAVIGATION MAP

START SCREEN
Level Select
Instructions

INSTRUCTION SCREEN
Go Back

LEVEL SELECT SCREEN
Level 1
Level 2
Level 3
Go Back

GAME SCREEN
Play Level
Win Level
Lose Level

LEVEL COMPLETE SCREEN
You Lost
You Won + Score
Go Back to Level Select

GAME OBJECTIVES

The objective of the game is to manoeuvre through the map to the exit. The exit will be disabled unless a key is retrieved prior to entering. The story behind the game is that the main character is an AI Robot and is being tested by scientists in terms of puzzle solving, one aspect of its decision making. Therefore, each level is a test more complex than the last attempting to train the AI in progressive puzzle solving.

VISUAL ASSETS

All visuals of the game are handled by the graphics artists. The game will be set in a robot universe in which our metallic hero must escape a faulty factory. They would need to pick up batteries for points, and keycodes to open the EXIT door. Each level is progressively harder as platforms move or outright disappear, the number of enemies and turrets increase, and the amount of time the player has to complete the challenge consistently decreases. Below are the assets provided to us for use in the game.

UI
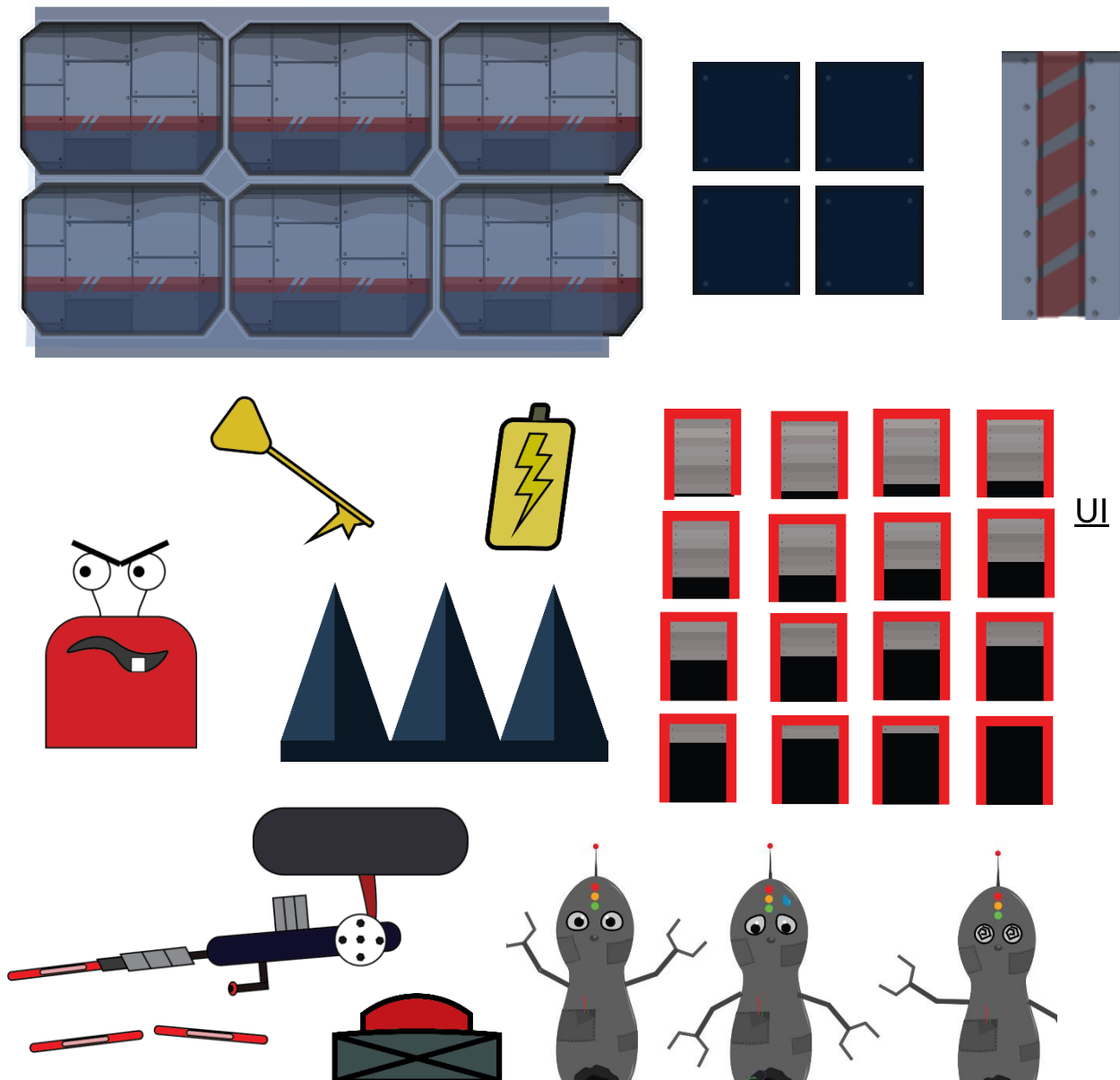
ELEMENTS
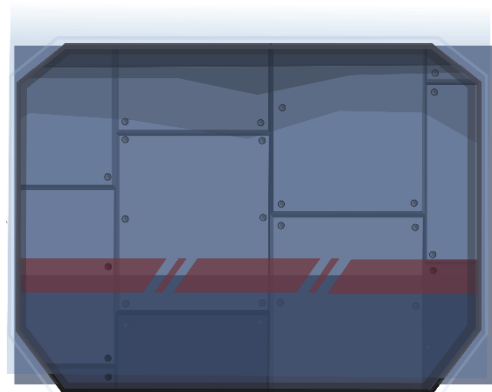
AI

The AI in the game comes in the form of the enemies preventing the player from reaching the end. The level needs to be manipulated via rotating to kill the enemy or by using the enemy's mechanics to aid you.

The AI is present in levels 2 and 3. The AI has an enemy follow function which takes the position of the player and compares it to the position of the AI. Should this difference be within its following range it shall start attempting to reach the player. Considering it has the same movement script as the player with 0 jumping power it will only be able to move left and right.

To make the AI more "intelligent" it will also assess whether it will kill itself by approaching the player in some instances. It will find the difference between itself and the nearest danger object being the spikes. If that distance is shorter than the player, it won't move. This is to prevent instances where the player is next to the AI but separated by a floor of spikes. Without the implementation the AI would surge forward and kill itself posing no threat to the player. The idea is that the player has to rotate the level in some method to kill the AI and get it out of the way.

Apart from that in the 3rd level we are implementing a turret AI. This AI's purpose is to create another small puzzle for the player to overcome. The turret will shoot in the players direction when within range. This would make the game very complicated if it could actually kill the player and poses some challenges to level design which we have no experience in. Therefore, the idea was it would be in a confined space with a button of sorts. The player needs to manoeuvre the level attempting to get the AI to shoot the button whilst attempting to hit the player which it will never manage. The hitting of the button would unlock somewhere the player needs to get to continue on in the level. This makes the AI more of a tool for puzzle solving rather than an adversary which was our idea all along.