CS7038 Group B

Hao Guan, Yaffy Koyo, Saloni Sharma & Jeremiah Dunn

1 Theme

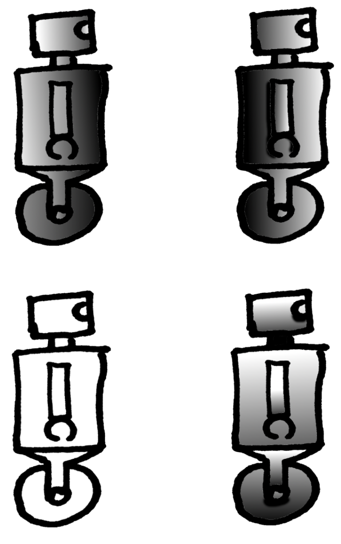
In our game the player controls a maintenance robot on board a spaceship on a long journey. While on the journey the ship malfunctions and begins shutting down, the whole crew are in stasis and it is up to the robot to restore power.

The player must traverse the ship, reactivating components and lighting the ship up in the process. The gameplay takes inspiration from exploration based platformer games such as *Spelunker*.

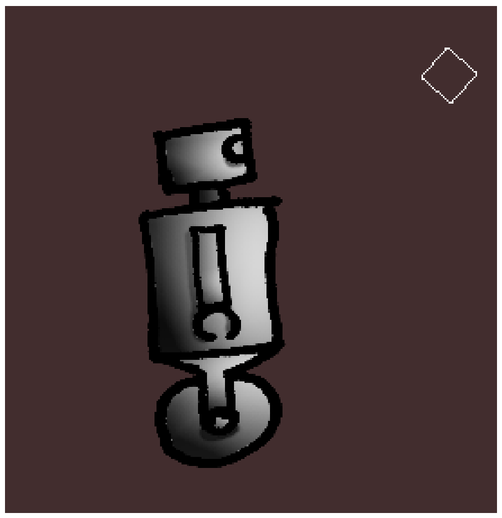
2 Gameplay

Movement and jumping will be implemented using Unity's 2D physics engine to handle.

Re-activating components of the ship has a probability of activating some of the ship's security measures. These will include activating traps or spawning hostile defence drones. On reaching the next component these security systems can be deactivated.



(a) input files including a base file and the same file drawn as if it was lit from a handful of different directions



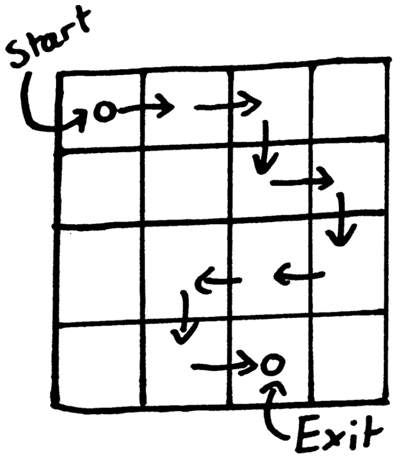
(b) output file reacting to a dynamic light shown by the white cube

Figure 1: Player character mock-up

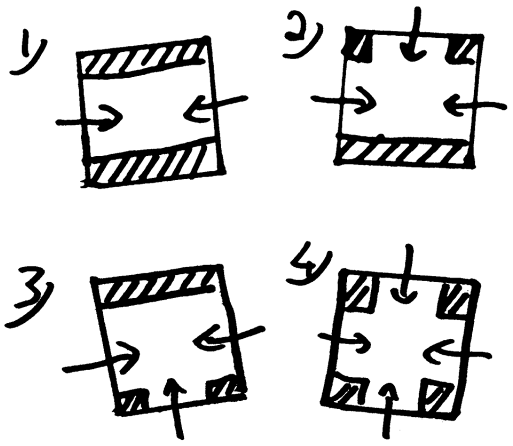
We plan to use 2d sprites, with normal maps created using *SpriteLamp* for the game's lighting effects. Figure 1 shows an example of it's use.

3 Level Generation

The game will use elements of procedural generation to create levels. To do this, first a grid of *rooms* is created as shown in figure 2a. A random position on the top of the grid is chosen to be the starting room. From here there is a probability of stepping left/right or down. If the stepping algorithm reaches a wall it will automatically step down. At the bottom of the grid the probability of stepping down is replaced with the probability of creating a level end.



(a) Room grid



(b) Different rooms, hashed areas represent floor/roof

Figure 2: Level Generation

After the room grid is created the individual rooms are populated with pre-existing room tiles. To get a basic version of the game running only the four room tiles show in figure 2b would be needed with solid rooms for the rooms off the solution path. Individual rooms will be tile based and will be stored in text files which the game will read in and then populate.

4 Admin/Planning

sprints and stuff