

- **Brief explanation of the game**

The game is a 2.5D platformer. The objective of each room (level) is to survive the enemy cubes and escape via a portal. Players collect abilities that accumulate throughout the duration of the game. Once the player escapes the three rooms consecutively, the game is finished.

- **How to use it (especially the user interface aspects)**

The main controls are listed in the controls panel accessed via the main menu. The controls for different abilities will be shown in-game once they are unlocked.

- **How you modelled objects and entities**

All objects and entities are modelled with primitive 3D objects provided by Unity3D. The Player is a sphere, and Platforms are created by scaling native unit cubes.

- **How you handled the graphics pipeline and camera motion**

The camera is set at a fixed distance from the player along the Z-axis, and tracks the player's movement exactly (ie keeps in sync with the Player's coordinates in the game/XY plane)

- **Descriptions of how the shaders work**

There are four custom shaders implemented in this project: HaloShader, PortalShader, LovelyShader, and InvisibleShader.

HaloShader is an extraordinarily simple shader that does nothing but write a value of 1 to the stencil buffer. No pixels are rendered and nothing is written to the z-buffer. This shader is only applied to the “halo”, an invisible sphere centred on the player that only comes into existence when the player turns invisible.

PortalShader and LovelyShader both check the stencil buffer to see if HaloShader has written to it. If it has, they do not render, creating the effect that objects within a certain radius of the player turn invisible when the player does.

PortalShader achieves an animated effect of rings of alternating colour drawing inwards towards a uv coordinate of (0.5,0.5). This obviously works best on flat shapes, and in the game is only applied to the portals between levels. On a per-pixel basis the result of the function $\sin(kx + wt)$ is calculated, where k is a parameter controlling the density of waves, x is the uv distance from (0.5,0.5), w is a parameter controlling speed (negative values can be used to reverse the direction of the effect) and t is absolute time. The product of wt is calculated at the vertex stage and passed to the fragment shader for efficiency. This result is taken to the fourth power, both in order to restrain it in $[0,1]$ and to narrow the ring width of the “Highlight Colour”. The rendered pixel is the colour indicated by looking up this value on a colour ramp between two user defined colours.

LovelyShader is a simple cel shader with three cuts. Implemented as a surface shader, this shader takes the dot product of N (the normalised surface normal) and L (normalised light direction), transforms them into one of four discrete values, and attenuates the colour of the rendered pixel using that value, the colour of the light, and the intensity of the light. The player (when visible), and the platforms are rendered with this shader.

InvisibleShader is a transparent shader that highlights the edges of an object. Applied to the player when invisible, this shader uses the per-pixel result of the inverse of the dot product of the normalised camera direction and normalised surface normal taken to the 1.5th power. If less than the original opacity, the resulting number is used for the opacity of the rendered pixel, leading to an effect where surfaces more perpendicular to the camera are more transparent and surfaces more parallel to the camera are more opaque.

- **Description of the querying and observational methods used, including: description of the participants (how many, demographics), description of the methodology (which techniques did you use, what did you have participants do, how did you record the data), and feedback gathered.**

We used the Questionnaire query technique and Post-task walkthrough observational method on 5 people each.

- **Questionnaire Questions (Rated: 1 (bad) - 5 (good))**

1. How appropriate was the difficulty?
2. How easy to use was the control scheme?
3. How clear were the objectives?

Questionnaire Feedback

Questionnaire No	Player 1	Player 2	Player 3	Player 4	Player 5
1. Difficulty	4	3	4	5	2
2. Ease of controls	2	2	5	5	3
3. Clarity of Objective	3	3	1	3	2

Post-task Walkthrough Questions

1. How was the difficulty?
2. How was the control scheme?
3. How clear were the objectives?
4. Any other suggestions for improvement?

Post-task Walkthrough Feedback

- Player 1 (20-25 yrs)
 - Game was of moderate difficulty, slightly difficult to coordinate controls
 - Controls are smooth and easy to understand
 - Did not realise at first shooting direction was controlled by mouse
 - Recommended making enemies “scarier” to suggest the player should shoot them (eg with a glow)
 - Recommended making projectiles more pointed for a greater sense of direction
- Player 2 (20-25 yrs)
 - Would prefer customisable controls and clearer control instructions
 - Found coordination between moving, jumping and shooting difficult
 - Assumed game would be an infinite staircase with a scoring system
- Player 3 (20-25 yrs)
 - The game difficulty was okay, it was not that difficult
 - The controls were very clear from the get go
 - If there's no people to explain, need to try 2-3 times to know what to do
 - Explain the objectives better, as I wasn't really sure what to do
- Player 4 (20-25 yrs)
 - I found the first level to be quite difficult, the second level was really easy, and the third level was okay too
 - The controls were good as there were only a few of them
 - It was not really clear that the mission of the game was to escape via the portals
 - Make it clearer that the objective was to escape the levels
- Player 5 (20-25 yrs)
 - Found jumping was disproportionately weak compared to movement speed
 - Found second level to be too easy, while the first level was too hard
 - Suggested making “W” the jump key, using “AS” instead of “AD” for movement, and allowing the player to use arrow keys
 - Thought general idea of heading upwards in first level to be obvious
 - Believed there were infinite stairs in first level with a scoring system

Document the changes made to your game based on the information collected during the evaluation.

We unfortunately had no time to implement changes to the game to reflect player feedback. However, it would be simple (just time-consuming) to implement these changes using our current game framework.

A statement about any code/APIs you have sourced/used from the internet that is not your own

- PauseManager Code - <https://forum.unity.com/threads/how-to-pause-a-game-x-seconds.433446/>
- Basis for LovelyShader - <http://xdavidleon.tumblr.com/post/122950440695/next-gen-cel-shading-in-unity-5>
- Basis for InvisibleShader - https://en.wikibooks.org/wiki/Cg_Programming/Unity/Silhouette_Enhancement
- Basis for audio classes + scripts - <http://brackeys.com/wp-content/FilesForDownload/AudioManager.zip>

A description of the contributions made by each member of the group

- Adam - Shaders, basic invisibility logic
- Emmanuel - User experience design, scene transitions, audio
- Ryan - Game mechanics, level design