Comp220 Proposal

Matthew Shaw
ms228668

February 1, 2021

1 Outline of intended artifact

1.1 Area for improvement

Too many games these days are focused around the boring 3D Euclidean space, simply because it's the one we happen to live in. What I propose is a puzzle game, similar to Portal, where the player will navigate a 4 dimensional space, aiming to reach the end of each level, then proceed on to the next one. This world would have an extra spatial dimension that the world we currently exist in. Doing so allows the puzzles to involve hiding things in different 4D spaces, which are impossible to see from the player's normal perspective.

1.2 Approaches to the problem

There are two main schools of thought for showing 4D hypershapes in 3D space: Projection and Slicing.

Projection In the projection method, the entire shape is crushed down into 3D space, allowing all parts of it to be seen at once. This gives rise to the classic tesseract shape of a hypercube, which apears to be a small cube inside a larger cube.

In Figure 2, both the inner and outer cubes are the same size, one is just farther away along the 4th axis. It is analogous to how in Figure 1, one square is smaller than the other, even though they would be the same size in

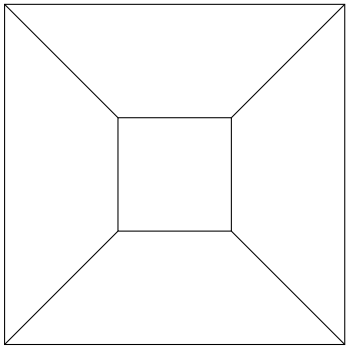

Figure 1: Shadow of a wireframe cube

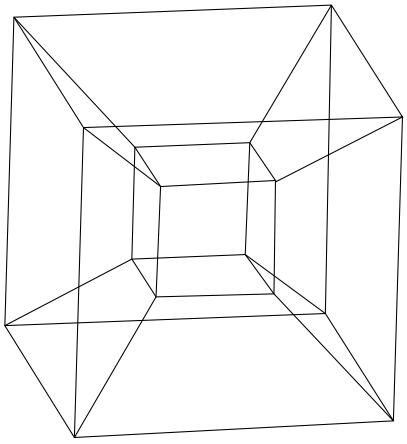

Figure 2: Shadow of a wireframe tesseract

the real cube.
In addition to this, all the pyramid-shaped areas in Figure 2 are cubes, like how the trapezoids in Figure 1 are actually squares.

Slicing The other method is simply taking a 3D cross-section of the hypershape. This is like cutting a 3D shape with a knife, and looking at the shape of the cut. If a cube is cut in line with one of the sides, the cross section will be a square. Similarly, if a tesseract is cut in line with one of it's faces, the cross-section will be a cube. This would show a hypercube just as a normal cube, and would not allow the viewing of all parts of the hypershape at the same time, which is a lot easier for a person to see and understand, but an additional control is needed to move through the 4th dimension.

1.3 Benefits of suggested approach

When I do this, I would prefer the slicing method over projection. This view is more intuitive from a 3D perspective, as looking at it would be akin to looking at a 3D scene. However, moving along the 4th dimension would make the shapes appear to morph and shift, when in reality they are simply being moved.

1.4 Potential challenges

4D maths is very hard, and I have been working on and off on it for over a year now, without significant headway being made. However, since this is actually being done for the university now, I should be able to get more support from university staff with this matter, which should help me along massively.
The player might also have trouble understanding what is going on when they move through the 4th dimension. I could try to mitigate this through storytelling or tutorials, and through the visuals. I could try fading the geometry along the 4th dimension, so that the player could see it fading in and out, however that may make it hard to discern what is passable and not.
Another issue is trying to implement a collision system. Although modular collisions systems exist for 2D and 3D, I am not aware of any for 4D. It might be possible to use a 3D system, but I think that the best way would be to create my own solution. The easiest way would be to use axis-aligned bounding boxes, and keep everything cubic (hypercubic?). I could also add

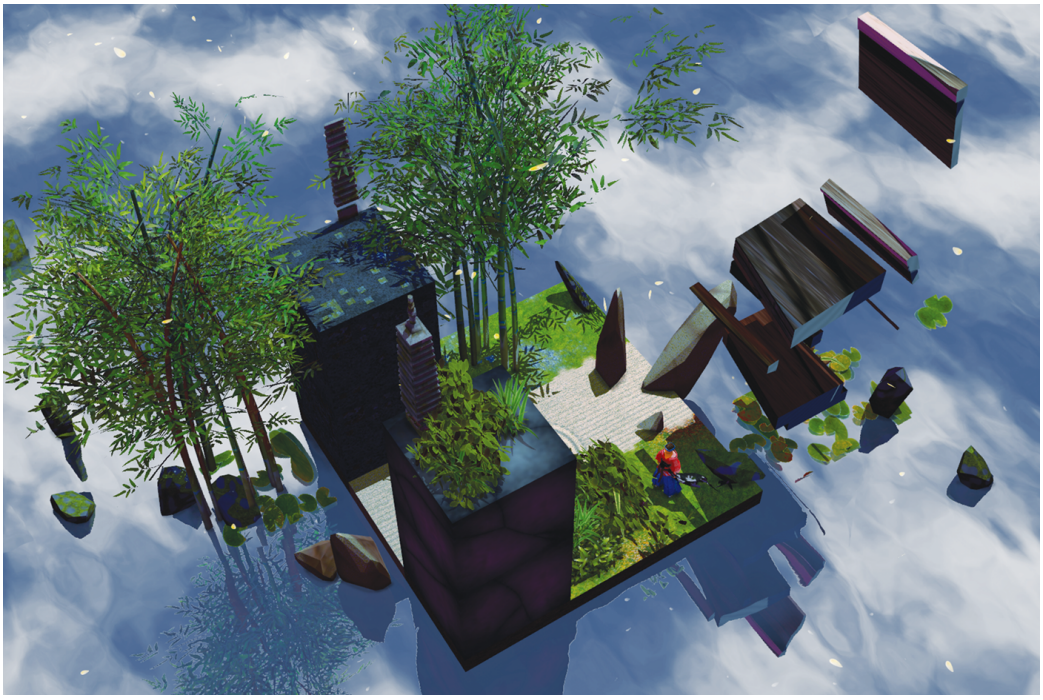

Figure 3: Miegakure

in spherical shapes, since their collision bounds are just all points that are a certain distance away from the center. Adding in spheres would also be a good way to show the 4th dimension, as if you aren't in exactly the right position, it will appear to float off the ground. This is like viewing a sphere in a 2D world, where if it is not aligned with the cutting plane, the cross section will appear to float.

2 Broader context

In my opinion, I feel like there aren't enough games set in the 4th dimension like this. I believe that the largest one is Miegakure, however that has been in development since 2009, with no final release date in sight.

There are also not a huge amount of games that defy normal space in general. The most popular are probably the Portal series, which link two normally separate spaces together via a portal. There are also games like Manifold Garden or Antichamber, which mess with space and reality in their

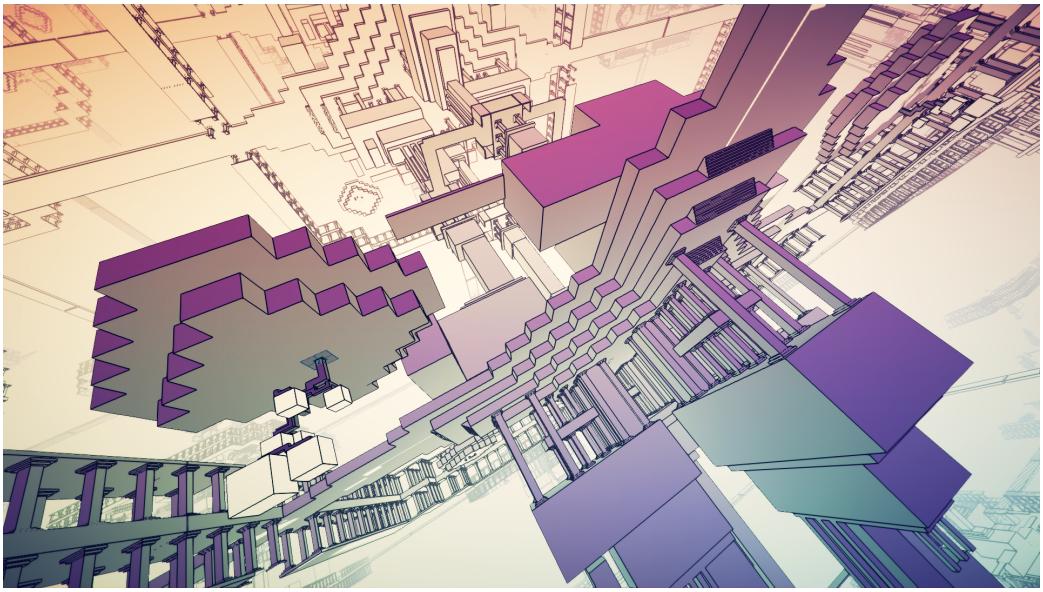

Figure 4: Manifold Garden

own way, but they aren't set in the 4th dimension.

Part of what I want to do with games is create novel experiences, and while 4D is not an completely new concept in games, its still relatively unexplored. Maybe after creating it, I'll find there is a very good reason why nobody wants games like this, but I can;t really know until I try it.

3 Project feasability

I believe that this is a project that it is possible for me to complete. While in the past I have struggled with this kind of maths, I am confident that I have improved in my abilities.

3.1 Breakdown

The hardest part of this will be to compute the 3D intersection of the 4D shapes. However, as long as the cutting plane is not rotated, and remains orthogonal to the 4th dimension, each edge can be tested to find the intersection points.

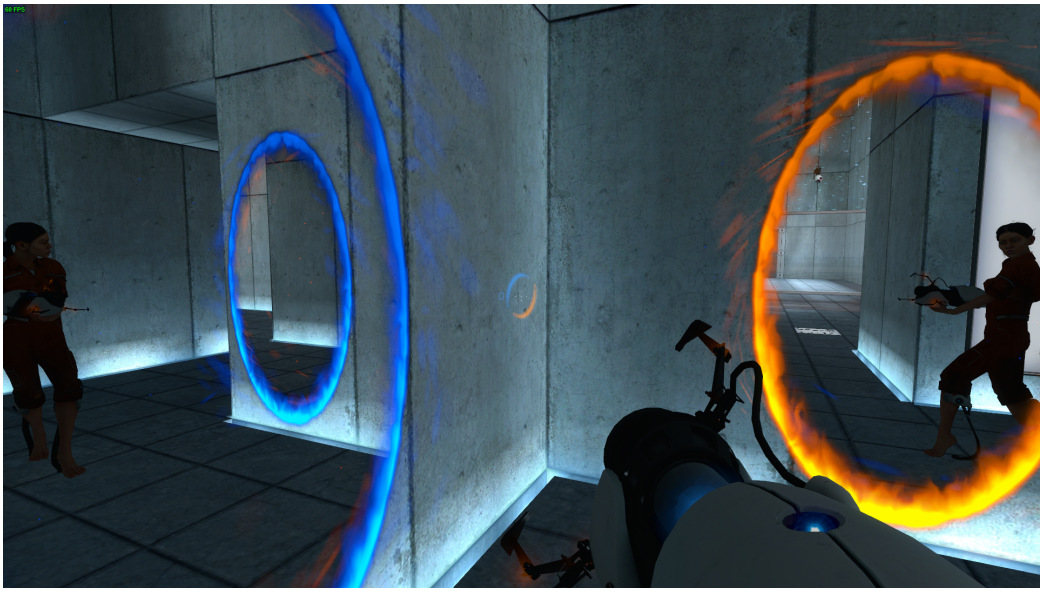

Figure 5: Portal

Each 4D object can have either a 4D bounding box, or more simply a maximum and minimm W coordinate stored, which will help with optimisation. Each edge also has both ends stored, so that it can be expressed in a vector form.

$$\left\{\begin{array}{c} x = x_0 + t(x_1 - x_0) \\ y = y_0 + t(y_1 - y_0) \\ z = z_0 + t(z_1 - z_0) \\ w = w_0 + t(w_1 - w_0) \end{array}\right\} \text{where t is between 0 and 1}$$

When the W coordinate is changed, each object can be checked to see if the W coordinate is within its bounds. If it is, then each edge will be checked. It will use the equation

$$\frac{W - w_0}{w_1} = t$$

to calculate t, then if it is between 0 and 1 the edge has an intersection. If it does have an intersection, the intersection point in 3d space can be calculated with the first set of equations. If 2 edges share a face, and are also both intersected, then the two intersection points should be joined with an edge.

Everything else can be handled by normal OpenGL rendering and inputs.

3.2 Contract requirements

In order for this proposal to be accepted, it must meet the requirements in the appendix of the assignment brief.

- "A Scene containing at least 1 textured mesh and at least 1 light source" - The game will have textured walls and objects, and ceiling lights.
- "Standard first person controls" - All controls will be as normal for a first person game, with the addition of a pair of controls to move through the 4th dimension, which will likely be Q and E.
- "At least 2 of the following graphics and simualtion techniques..." - I will be implementing collision detection, and the 4th dimension aspect could be considered "non-realistic rendering", or the catch-all "Other advanced rendering or simulation techniques of your choice".
- "Some aspect intended to create fun and/or engagement" - This will be completing a simple gameplay objective, as the game is a puzzle game.