CS 4621 Project Proposal

September 10, 2025

Andrew Arena (aja298), Sterling Chargois (smc453), Nolan Lizmi (nbl25), Luke Tao (lyt5), Ethan Zhang (epz6)

Description

Our vision is to create a four-dimensional (4D) “dungeon crawler” game—i.e., a game where the goal is to explore a procedurally-generated dungeon, fight monsters, collect items, and ultimately escape.

The 4D aspect of our game is what makes it unique. We use four spatial dimensions, and allow the player to change their location in the added dimension. The player can, of course, only perceive three dimensions at a time, so the idea is that the player experiences a 3D “slice” of the 4D world at any given moment. The “slice” they are in is controlled by their fourth coordinate, or “w” coordinate, which they control as they would any other coordinate (i.e., using movement keys).

We would like 3D spaces that are nearby in w-space to have some logical consistency. This means that moving in the fourth dimension would have an effect that is, while at first mysterious, ultimately predictable to the user. For example, new doors and staircases may appear, or walls may move to change the sizes and shapes of the rooms the player can explore. We also would like to include enemies for the player to fight or evade, and boss battles to challenge the player and check their understanding of the world. Combat with these enemies would either be direct (using weapons) or indirect (using trickery and the player’s knowledge of the 4D environment).

We envision that the players will initially traverse the dungeon slowly, using trial and error to navigate the extra dimension. However, as they progress through the game, our goal is that they develop an intuitive understanding of how movement in the fourth dimension changes their environment, and eventually master the skills they need to engage in combat with increasingly difficult enemies and solve increasingly difficult puzzle rooms.

The second defining aspect of our game is the procedural generation of the dungeon. We currently plan to do this using the wave function collapse algorithm in four dimensions, though this may change during development due to (for example) considerations of practicality or performance. The use of wave function collapse entails that we would put together a set of 4D “tiles” for the algorithm to pull from (for example, a few different staircase tiles, room tiles, and hallway tiles). We then provide the algorithm with a set of rules for how these tiles can fit together, and then it assembles a dungeon for the player to explore. Creating these tiles and having the algorithm output cohesive worlds will be challenging, which will be touched upon in the next section.

We would also like to include visual effects that highlight the unique aspects of our game. One would be a “ghosting” effect for when enemies or important items are located in nearby “w-slices” but not the exact w-slice that the player is in; the idea is that they would appear translucent until their w-coordinate matches exactly with the player’s, at which point they would appear opaque and be interactable. We also need to think of some intuitive visual indicator that gives the player knowledge of the w-slice that they are currently in; we think that at least during development, a color change of the environment may suffice.

Tools

A majority of our group is very comfortable with Unity, so we decided that this would be the game engine of choice for us. Because Unity was designed as a three-dimensional game engine, it currently contains no four-dimensional capabilities. There are therefore a range of extensions that we would have to implement in order to

(1) have a working four-dimensional world and (2) properly render three-dimensional slices of that world. Most importantly, we would need to develop a new coordinate system for locations of objects using four coordinates (x, y, z, and w) instead of the typical three (x, y, and z). This is very simple conceptually, but it will likely involve the development of several different tools. For example, we will likely need to create a “model editor” within Unity that would allow us to efficiently and intuitively create four-dimensional tiles and see what they look like in all w-coordinates.

Rendering 3D slices of these 4D geometries also sounds fairly simple at surface level (we simply only render the points that lie in the player’s w-coordinate), but it may prove very difficult—for example, we need to figure out how to interpolate between geometry vertices located at different w-coordinates and properly display these surfaces as meshes. This may involve writing HLSL code and working with shaders.

We would also likely need to build a 4D tile system in Unity to facilitate procedural generation. This would allow us to quickly and intuitively put together “example” dungeons that the wave function collapse algorithm can use as a base for its procedural generation. Tuning our example dungeons to get the desired kind of output for our dungeon will likely take a lot of trial and error, and a 4D tile system would help to reduce this headache.

Milestone Goals

There are a number of milestones that we would like to achieve in order to meet both our basic and extended goals. We will first cover what we will ideally have completed before our “milestone” presentation. This would be having a playable, procedurally-generated 4D dungeon that can be navigated. This would be achieved by extending Unity to support 4D coordinates and meshes, creating 4D models to use as tiles for the dungeon, programming a procedural dungeon generation script, and implementing a movement scheme through the fourth dimension.

There are many areas where we would like to go beyond this, time permitting. One feature we would like is to have enemies and a combat system. Enemies alone bring a range of development opportunities—we are specifically interested in the contrast between enemies that can “see” all four dimensions versus enemies that can only see three dimensions at a time (such as the player). The enemies that can see in four dimensions would perhaps be able to trick the player by leading them into traps that are invisible in other w-coordinates, for example. In contrast, if the player has knowledge of adjacent w-slices of the world, they can use this knowledge to evade and deceive the enemies that can only see in three dimensions.

Another possible expansion we thought of exploring is the idea of a “minimap.” The player would be able to hold down a button to see a 3D overview of the parts of the dungeon they have explored so far; they would then have to use their fourth-dimensional movement keys to see the minimap change in the fourth dimension.