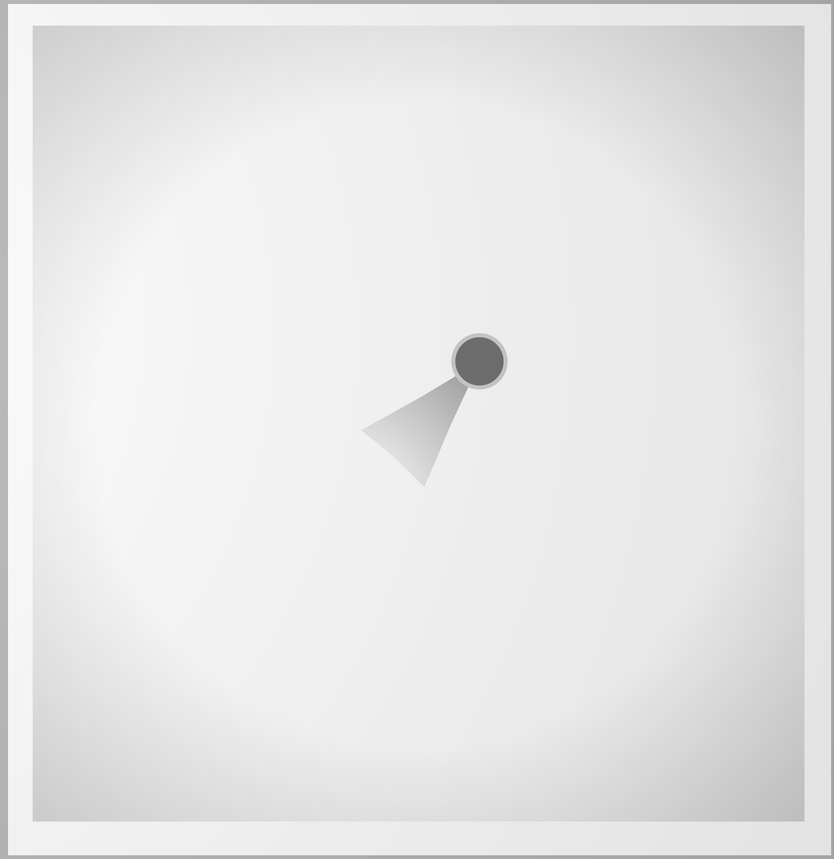
Viewport

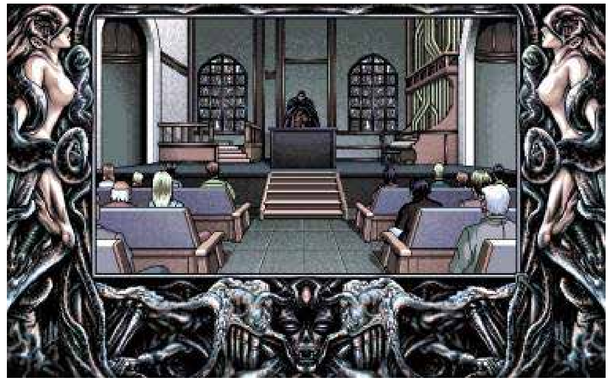
randomized visual elements
comprised of foreground,
midground, background

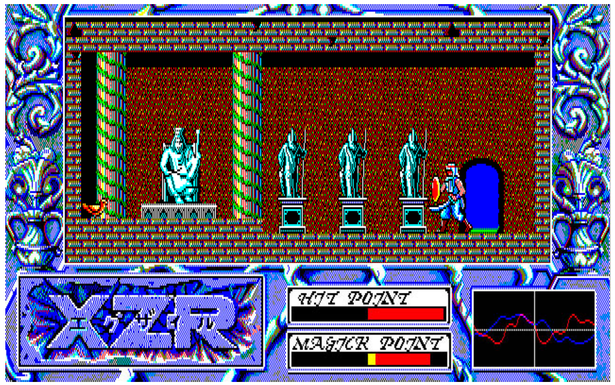
Mini Map

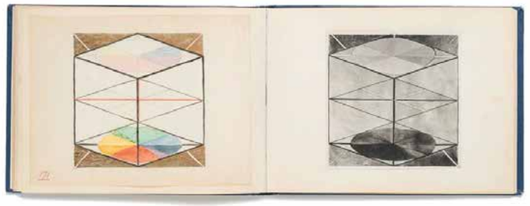
The user controls a character
that emits spatialized audio

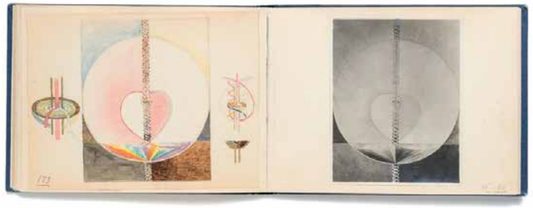


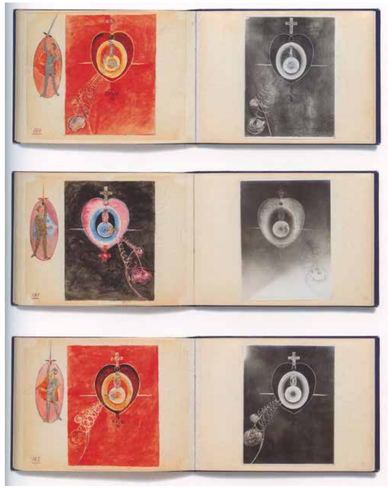
Although in this overview the HUD and example images are simple and “clean”, my aim is for the final version to follow these inspiration images:

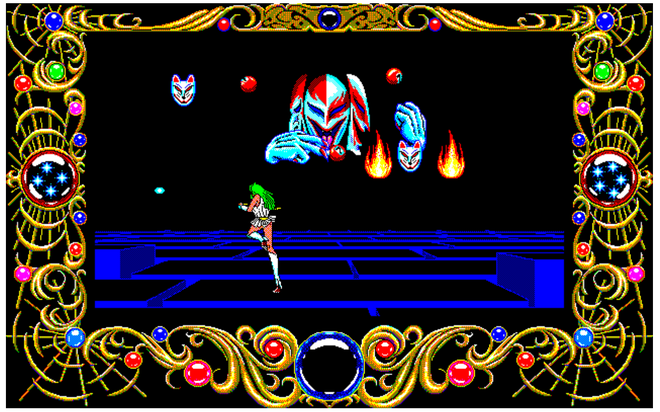


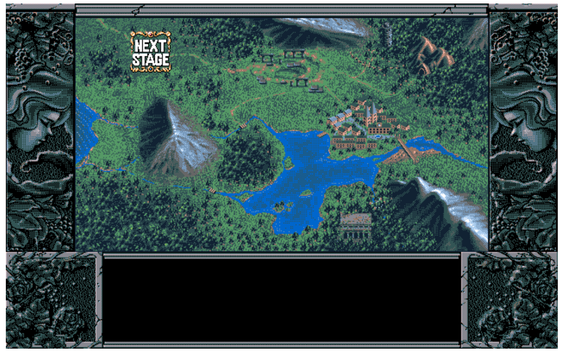


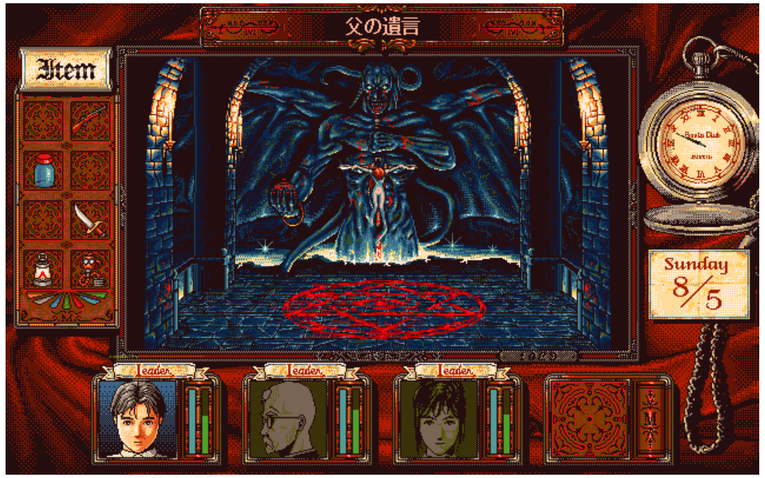












Project 2 Proposal - “Reunited at Last” (Working title)

For my final project, I have decided to create something that contrasts with the hyper minimalism, functionalism, and supposed clarity of the Web 2.0 user experience. I’d like to create a program that channels the sense of obscurity and mystery I found when playing early PC games as a child, particularly those with fantastical settings. In this sense, my program will require some opacity for the user, though ideally this will encourage engagement rather than frustration.

The program’s narrative, though never explicitly spelled out, tells the story of two spirits who have become separated from one another. The user’s role is to help guide these two entities toward one another with the aim of being reunited as one. One of the spirits is stationary, tied to a specific location, looking out at the horizon. The second spirit has movement but is only able to see the world through a top-down perspective. This spirit also emits sound, perhaps some sort of longing cry? Tragically, neither spirit is able to directly see the other, so the use of spatial audio will be fundamental to their reunion.

To realize this story, I will employ an art style infused with mysticism and surrealism. Practically, this will involve a HUD that is visually cluttered, inspired by the unnecessary and wonderfully garish HUDs of old fantasy PC games. There will be no written instructions or English text in the game to further enhance this aesthetic. The game’s audio will also reflect this sense of otherworldliness through the use of ambience, sparse music, and the repetitive cries of the searching spirit.

For this program, I am building off of what I made for the first project in this class. As in that project, spatialized audio will play a major role. The “searching spirit” is essentially an audio object that emits sounds. The user will be able to move this object through keyboard controls. It will be represented as a blip on a top-down mini-map that represents the landscape. The “stationary spirit” will be randomly positioned on this mini-map, but without any visual representation. The user will control the “searching spirit” and have to rely on the spatialized audio to determine the proximity of the “stationary spirit”. Once the two spirits overlap, the program will enter a conclusion state.

While the “stationary spirit” is invisible on the mini-map, the second main component of this program is the viewport. Here, we are given a stationary first person view from the perspective of the “stationary spirit”. Initially I had hoped to implement some sort of 3D or pseudo 3D code for the viewport, so that the user would be able to switch back and forth between controlling the two different spirits. My initial idea was that when controlling the “stationary spirit”, the user could at least look around in some fashion, even if it was just views from the four cardinal directions.

However, after looking into Phaser, p5's 3D capabilities, and even three.js, I felt a bit overwhelmed by the complexity of it all. Even implementing pseudo 3D as shown in a Code Train video on Rendering Raycasting seemed rather difficult. I'd love to be proven wrong though if you have any suggestions!

So in light of this, I decided that the viewport would be composed of images (with transparencies) layered on top of one another. This involves having a foreground image, a middleground image, a background image, and perhaps a layer or two for extra visual flair/effects. These images are stored in seperate arrays and randomly selected so that every time the user starts the program, there is a good chance that the viewport will look different. It is my hope that this will capture some sense of three-dimensionality and space that I was initially aiming for when I looked into 3D options. The images for the viewport will be created with my usual visual art process. Currently, in the prototype, there is some pixel art I quickly threw together, but this is not at all representative of the style or quality of what I am aiming for for the completed program.

I would still love to have some kind of user control over the viewport through the implementation of a "spirit switching" mechanic. The program could be some bizarre single-player version of the kid's game "Marco Polo". In particular, my initial idea was that it would be interesting to control two separate entities, and that reuniting them could have an interesting effect on the user. However, I have yet to figure out how to best achieve that idea.

In order to increase the complexity of the program, help the user "win", and add visual interest to images in the viewport, I've considered adding some kind of visual effects to indicate when the "searching spirit" enters different ranges of proximity to the "stationary spirit". I'm not entirely sure what the effects would be, but perhaps something along the lines of the image layers changing tint and shaking when the "searching spirit" enters a corresponding hitbox/circle. This would also give visual feedback to the user and give a mechanical function to the layering of the images as foreground, midground, and background. Similarly, I want to add corresponding versions of the "searching spirit's" audio, so that the voice gets more excited as it gets closer to its other half.

One thing I've been really wanting to learn about is shaders, though I am likely still far too novice to take this subject on. That being said, I'd love to use the effects found here: https://bandaloo.fun/merge-pass/example.html?m=motionblur. Apparently there is a p5 library version of those effects, but I was unable to get it to work. After looking into Phaser I am also very interested in that engine's use of shaders, though I'm not sure if it is difficult to bring everything from p5 into Phaser or if it would ultimately be worth the effort. Phaser's ability to use multiple cameras is also interesting, though with my current viewport solution I believe they would be unnecessary.