REMNANTS OF TIME

Chapter 4: Alpha Release

New Features

This section outlines the new features implemented since the interim report.

Boss

The boss fight is the conclusion of the game built up from the introductory cutscene. The main fight is split into phases. During the first phase, the boss continuously advances toward the player and performs one of several possible attacks. Whenever the player shifts in time, the boss follows with a slight delay. In that time, the player is able to set up a trap. Once the boss is trapped, the force field protecting them from the player's attacks vanishes and the boss can be harmed. Once the boss's health is completely depleted, the next phase starts. For all following phases, a new remnant of the boss is generated, following their previous actions. Once the boss is defeated and the maximum number of phases is reached, a door in the boss room opens and the player can leave the game, leading to an end-of-game screen.

The boss fight is also aided by cutscenes consisting of dialogue boxes with the boss's dialogue.

Sound effects

The player, enemies, puzzle elements and more things create sound effects. It is implemented with a system that caps how many sound effects of the same type can be played at the same time. This allows for e.g. dozens of fireballs to be spawned, or multiple enemies to be alerted at the same time without it being too loud on the player, while still getting all necessary information.

Music framework

Although the tracks themselves are not finished yet, the framework for the music system stands. With this the music will dynamically change to different versions depending on the current situation of the player. For example there is an overlay track playing, when in combat, intensifying the music. Furthermore in the past and present different versions of the same music will be playing. There will be a theme for the dungeon and a separate one for the boss.

Puzzle elements

Interactable Objects, the player has to use, to solve time related puzzles.

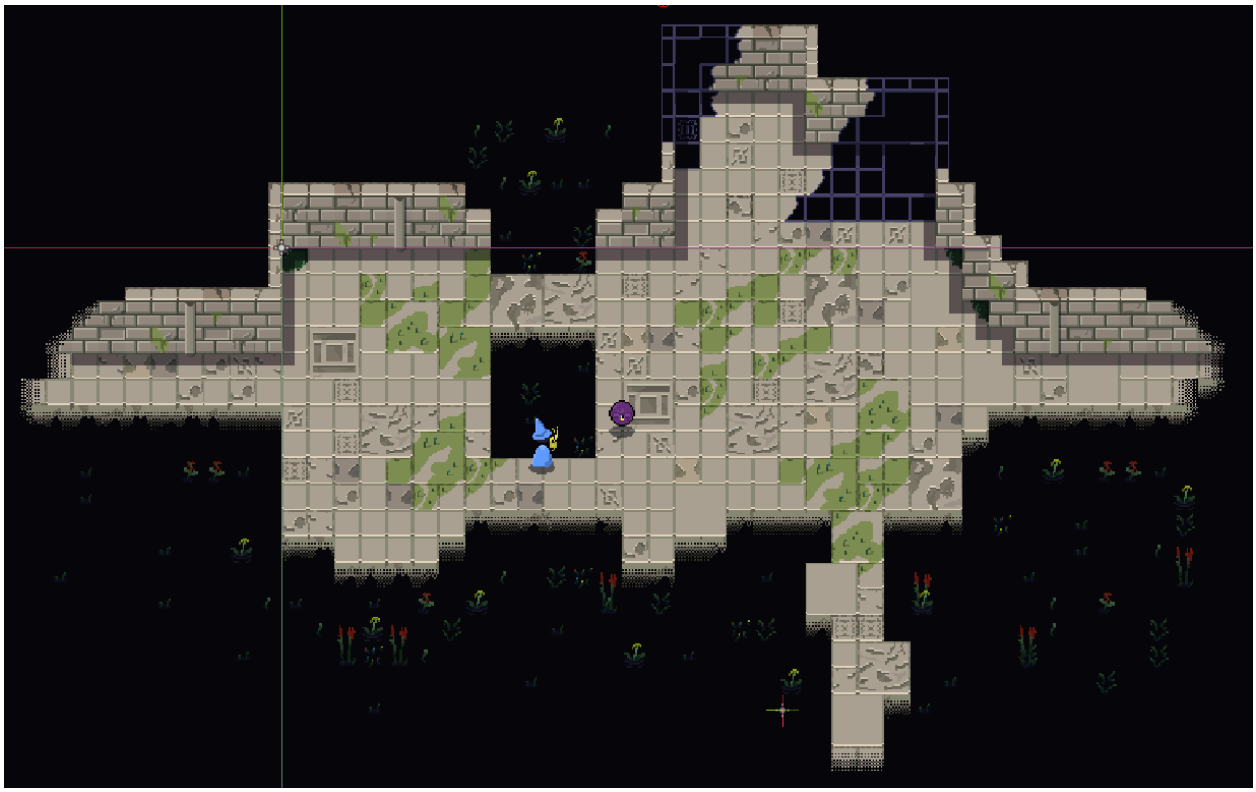
Pushable boxes can be used as dynamic solutions, e.g. to block line of sight of enemies, or solve other puzzles. If the same box exists in both the past and the present version of the level, they influence each other. The Player can move the box in the past and will find that the box in the present will also have moved to the same position.

A similar concept can be seen with mechanical doors. It would be unfortunate, if the lever that opened the door is not functional anymore, because time wore it down. Thankfully the player can time shift to the past, activate the lever, and see that the door is… still not open? Because there is a Button that also needs to be active at the same time. But it deactivates itself after just a few seconds. Sending “yourself” as a remnant to hit the button can solve this problem, while you yourself stand in front of the door to squeak through.

There are energy pillars where you have to send an energy ball from pillar to pillar until it reaches its destination. But the energy dissipates, if you aren’t fast enough. If “you” aren’t fast enough, maybe multiple “yous” are, requiring the use of the ancient remnant.

New Sprites

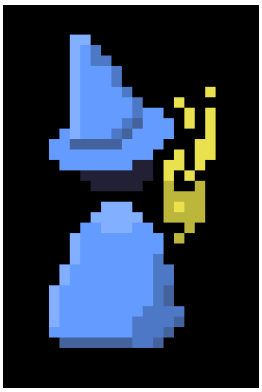
For the environments, we went with a 2D tile system, in particular we settled on 16x16 pixel tiles. Being inexperienced in pixelart, the style is inspired and supported by various artists that shared resources for beginners online. Tiles have many variations to avoid the environments seeming too surreal.



While it could still use some work, we are happy with the result and consider it in a playable state. The environments will feature dungeon structures made of stone interspersed by “ethereal” sections characterized by muted purple outlines over transparent tiles. The presence

of these sections is meant to convey the fact that something is up with the time travel powers present through the game, giving the player additional elements to be wary of.

In addition to the environments we also made sprites depicting a wizard, to be used by the player and the boss. These also feature 4 animations: An idle state, a horizontal walk cycle and separate upward and downward walk cycles. The horizontal walk animation can be mirrored depending on the direction the player is walking toward but, because of the fake top-down perspective, upwards and downward walk cycles see different sections of a character and therefore cannot be drawn separately. These animations are composed of 12 hand drawn frames each and play at 12 FPS.



We also made sprites for some miscellaneous particle effects, such as sword attacks.

Changes and Fixes

This section outlines any changes made to the features implemented up to the interim report.

First Playtests

After the interim report, we uploaded a first version of our game to itch.io and showed it to friends and family for first impressions and to test the implementation of our basic mechanics. Our idea and implementation received positive feedback, but we also discovered several bugs we have not thought of during implementation.

We used this valuable feedback to improve our implementation as described below.

Time Shifts

Due to various movable objects in the past and present environment, free areas where the player can walk can differ across the time periods. Whenever the player shifts in time, they need to keep track of where they can timeshift to, so that they do not get stuck in interactable objects. This issue was noted by one of our first playtesters.

Enemy Behavior

The enemy behavior is very complex and its interaction with the other game systems turned out to be not as clear as we had originally hoped. Some issues have turned out to be very convoluted and difficult to polish. While we tried our best to fix them, we decided to leave some strange behavior in favor of improving other, more easily improved sections of the game.

Nevertheless, several easily fixed issues have been resolved. This includes handling enemy death in combination with players setting and reloading anchors, as well as stopping enemies from getting stuck walking into each other by bouncing off of each other when they're about to collide with other enemies.

Scene Loader

While feasible the initial idea for a dynamic loader turned out to be out of scope for the project. We found many small problems and unforeseen interactions with other systems that would take more time than we have to properly address.

Faced with these problems we decided to scale back the feature rather than completely remove it. Now scenes are pre-placed and positions are decided at compile time instead of dynamically calculated at runtime. This approach requires more manual work on our side but results in a more isolated system that needs less work as development goes on.

Upcoming

While we managed to implement all desired features, we are unfortunately behind on combining all features into fully playable levels. We will spend the rest of the week setting up the levels. Simultaneously, we will prepare the playtesting phase and look for playtesters.

Conclusion

While we are satisfied with our idea, it has turned out to be difficult to implement in many ways. Nevertheless, while we are a little behind on our original plan, we have implemented most desired features and are confident that we will be able to combine everything into a fully playable version with enough time to perform the playtesting.