Project Management Document

2329108 - Andrew Kapp 2351879 - Ricardo Costa-Tré 1946550 - Ash Jurisich

Scope

Forbidden Labyrinth (FL) consists of 3 main subsystems: a FirstPerson combat, Inventory and Crafting system, and procedurally generated dungeon.

First Person Combat

Controller

- First Person Camera
- Player Interaction
- Player Control

Combat

- Responsive Enemies
- Multiple Enemy Types
- Enemy Navigation

Inventory and Crafting

Inventory

- Item Holding
- Item Collection

Crafting

- Item Recipes
- Item Creation
- Upgrade System

Procedural Generated Dungeon

Will be used to generate a new dungeon for the player upon player-entry. Room by room generation to allow unique dungeons and experiences.

Background

This group is far more technically inclined without a resident artist. In light of that, a more challenging project has been thought up. Thus this project is seen as a technical challenge for the group where challenging aspects such as enemy AI, and Procedural Generation are included.

With each challenge in this project, there are more and more challenges underneath as the challenges are explored and systematically Solved.

Division of Roles

Andrew Kapp

- Character Controller
- Back up Asset Creation
- Combat System

Ricardo Cost-Tré

- Enemy AI
- Inventory
- Crafting System
- Combat System

Ash Jurisich

- Procedural Dungeon
- Sound Assets

Pre-Alpha Goals and Challenges

Character Controller

The character controller requires more versatility with regards to the actions the player can perform. The layout of the player mimics Kratos from God of War closely with regards to the actions it can perform and the format of Weapon and Offhand systems having roles that complement each other; defense and attack.

The Weapon was established in the first prototype as the source of player damage, with attack animations and events. The offhand was the first process to be added and built, the offhand is to be the main source of playstyle versatility as it is planned to either hold a shield for defense or a spell for more aggressive players. The extra versatility of the offhand adds an extra layer of complexity to system design since the in-game system needs to respond differently to the same input based on the currently equipped offhand.
Due to that the shield was the best starting point since it is a fairly simple motion and premise.

Goals

- Begin with **playtesting -** the Controller is the most direct link to the player's experience since it is how they interact with the game.
    - The main feedback being sought after is to get a sense of where the look sensitivity lies for various players.
        - Having a camera that moves too slowly or too quickly can negatively impact the player experience.
        - This will give a sense of where the settings in future iterations should lie.
    - Recording and Noting player experience with the controller is important.
        - Noting what they try, what they do not engage with and how much of the controller functionality they use/expected to have.
- Build **functionality** for versatile off hand functionality.
    - The offhand should hold the player's shield or magic abilities.
        - The same input for different functionality: shields behave differently to a spell.
- Begin **integration** with inventory system
    - The Player's ability to pick up and store resources and enemy drops is a cornerstone of the vision for the game due to the inclusion of a crafting system that serves as the main source of player progression.
- **UI integration**
    - Character Controllers and systems need to communicate with the player.
        - Player health, inventory, whether an item can be picked up or not is integral to Game-Player communication.

Outcomes

Playtesting

Due to a lack of a comfortable time frame in which to sit with a playtester and interrogate how they interact with the prototype and ask them questions about the system; playtesting was minimal for this prototype.
However the small amount of testing that was achieved showed that base mouse sensitivity was far too high, the existence of an attack combo was not obvious to the player and that the attack animations felt sluggish and uninteresting due to the small amount of player feedback.

Since there was a small amount of testing that actually came back, the only thing that could be improved upon at this time was the mouse sensitivity being too high. The tester said they became easily disoriented since a small movement of their mouse translated to the player turning around multiple times, thus the sensitivity was drastically reduced.

Functionality

The plan for this iteration quickly became making a framework that will theoretically support the use of multiple types of offhand abilities. This means creating a pathway through which the equipped offhand dictates the ability that is currently available for use.

This being said, the ability must be attached to the object and be able to communicate to the animator and the player's input. Thus, the offhand holds a reference to then animation name and handles all component handling on the off hand itself such as collider handling and effects.

Combat System

The Player combat system contains a simple 3 hit combo and the ability to block incoming damage. The current system make the player take damage if enemies get too close to the player. This is a simple solution to demonstrate and test player health and communication systems. In later iterations the method call for the player to take damage will be called externally and thus place the act of taking damage in control of the enemies rather than the player just taking damage for the sake of it.

The enemy combat system contains a basic state machine that will allow for further states to be implemented with ease. Currently, the enemy AI has two different states that include a chase state, and an attack state. Further plans have been made to implement further enemy states,such as that as a dodge state.

Inventory and Crafting

A basic crafting system is still in development and can unfortunately not be previewed by the “Pre-Alpha”. While the system is near completion, it is not in a working state that allows for its use and functionality to be properly displayed as of yet.

The inventory system currently implemented allows for the player to view their inventory, pickup items, as well as having a UI that shows the system updating the inventory.

The system is also currently set up to allow for easy expansion of resources and actually are in a state where they are working well with the crafting system currently in development. Most of the time on this system was focused on creating a modular system that allowed for easy expansion and easy integration with other systems.

Enemy AI

Changes to the original idea of the dynamic AI have been opted for. Originally, the idea was to have an AI that acted and learnt on the spot, manipulating movements accordingly, however this system was overly complex and advanced.

As such, the use of an enemy state machine was used to control the basic states of the enemy, such as chasing the player, and attacking the player. With this change, the use of the neural network has been transferred to that of a dynamic difficulty system.

Dynamic Difficulty AI

The idea for this dynamic difficulty system is for the game to adapt and change enemies after every complete/failed run and change attributes of the enemies to either strengthen or weaken them. Ideas currently involve things such as stronger attacks, quicker attack speeds, more health, etc.

Currently this system is still in very early stages of development and experimentation, hence unable to appear in this prototype. Since this idea is also very theoretical, it may come to fruition that this idea may have to be scrapped entirely, however that is not current goal.

Development Tools

Unity will be used as our primary application for development. Within unity, the new input system will be used to handle Player Input.
Blender will be used in the event of creating an asset, with Krita as a supporting software for textures.
Assets and Animations will be obtained externally from websites such as SketchFab, and Mixamo.

Communication Tools

Primary Project Communication takes place on a private discord server. Less official communications will be carried out over Whatsapp on a group.

Risks

- Lapses in communication
- Unforeseen challenges with other courses or personal issues.
- Load Shedding.

Challenges

Given that the project holds many obvious challenges and aspects that will prove challenging or possibly problematic for the group:

- Procedural Generation
- Enemy AI
- Implementing a Neural Network into the dungeon.
- Time Constraints
    - Being a group of engineers, the time available to work is less than that of some other groups.
- Weapon balancing
- Enemy synchronicity
- Loot drops
- Art Direction and Implementation

Conclusion

This is a highly challenging but highly rewarding project. It was thought up and chosen to include aspects from games the group members enjoy from the games we play and what we would like to see in the games in the future.