Game Studio Project
Director Meeting

28th February 2025 - Group H

Josh Holloway

Producer - Technical Art, Prop Art,
General Design

Summary of Current Idea

3D Action/Adventure Game blending narrative depth, exploration and fast paced combat

Themes:

- Horror
- Post-Apocalyptic

Narrative

- Play as a child (named Arnion) with no memory
- Begin in a destroyed, ruined land infested by a force that displays as black goo
- Arnion finds an object that acts as a narrative device (speaking to the player) and the player's weapon
- Enemies are manifestations of this force that are hell-bent on destruction
- Ancient greek religion inspiration

Gameplay

- Isometric Perspective
- Player parrys enemies not to kill but to fill a meter
- When meter is full, the object is activated and follows the player's cursor to go through enemies to damage and kill them
- Levels have one critical path and multiple branching pathways

Art Style

- Dark colours with contrast of a few deeper, neon colours
- Shape of models are bubbly and cute in contrast with the tone and colour scheme

Josh Holloway

Team Insights

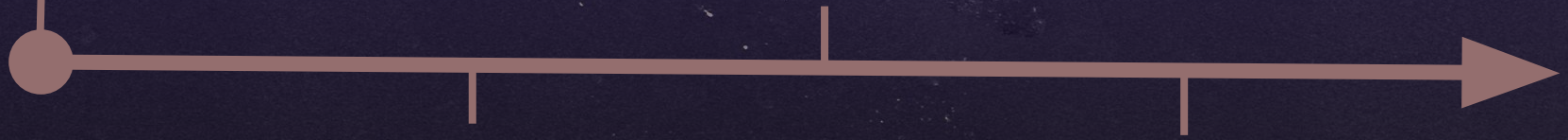
- Most of the team are actively engaging in at home work.
- At team meetings, most of the team has been attending whether that be online or in person.
- During the meetings, the majority has engaged in conversation, discussion and ideation.
- In the discord server, everyone presents their work and provides constructive criticism to others.

Josh Holloway

General Project Timeline

Pre-Production Weeks 2-4 **Currently Here**

- Team roles established
- Full idea about what the game will be like
- Concept Art for characters, environment and props
- Level Blockout created
- Systems Architecture framework concept made
- Version specifications established
- GitHub repository created
- Market Research conducted
- Game Design Document started
- Pitch created and performed



Production Weeks 5-11

- Core gameplay mechanics implemented
- 3D Assets for environmental and intractable props created, textured and implemented
- Prototyping combat mechanics
- Enemies system developed and implemented
- 3D Assets for characters textured, animated and implemented
- Satisfactory game testing conducted
- UI (HUD, menus etc.) designed and implemented

Refining Weeks 11-12

- Graphical Quality updated if necessary
- Lighting updated if necessary
- Any found bugs in gameplay systems fixed
- Any found bugs in graphics fixed
- More in depth playtesting conducted

Finalizing Week 12

- All gameplay systems implemented and functioning well with minimal bugs
- All graphics implemented and functioning well with minimal bugs
- All animations implemented and functioning well with minimal bugs
- Game executable built
- Gameplay Preview created for presentation
- Deliverables submitted

Josh Holloway

My Contributions

This week I have been focussing on creating pre-production images for the lantern designs. For the vertical slice demo, we are planning to incorporate 2 elemental lanterns that influence combat and allow you to solve more puzzles upon unlocking.

Each of the lanterns will also be used as a light source for dark areas in the level as we have a focus on the dark and fear.

Josh Holloway

Lottie Hill

Lead Designer - Character Art

This week's Timeline

21st
Discussed the level blockout further, refined the player/ combat mechanics and finalised tasks for the week

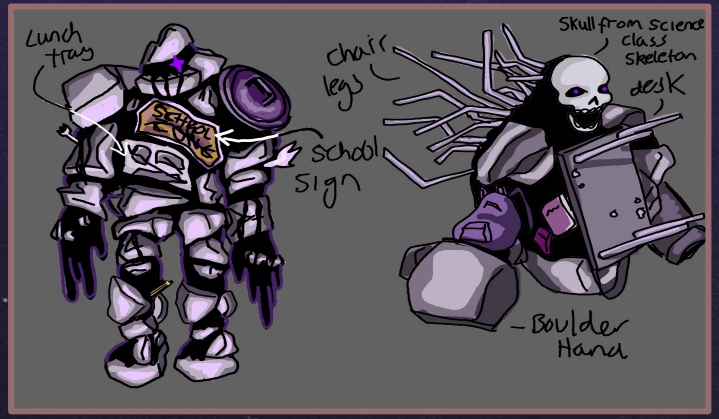
26th
Discussion of worldbuilding (through religions and the central narrative) and refined our ideas for the companion and monster designs. Also reiterated the colour scheme and art style.

28th
Further discuss level design, worldbuilding and central narrative.

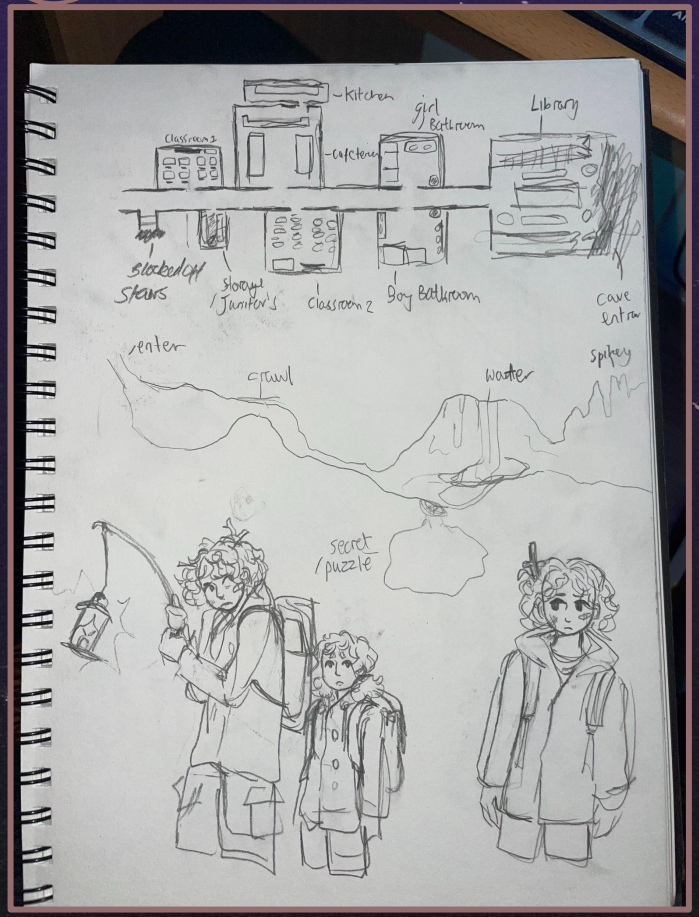
Lottie Hill

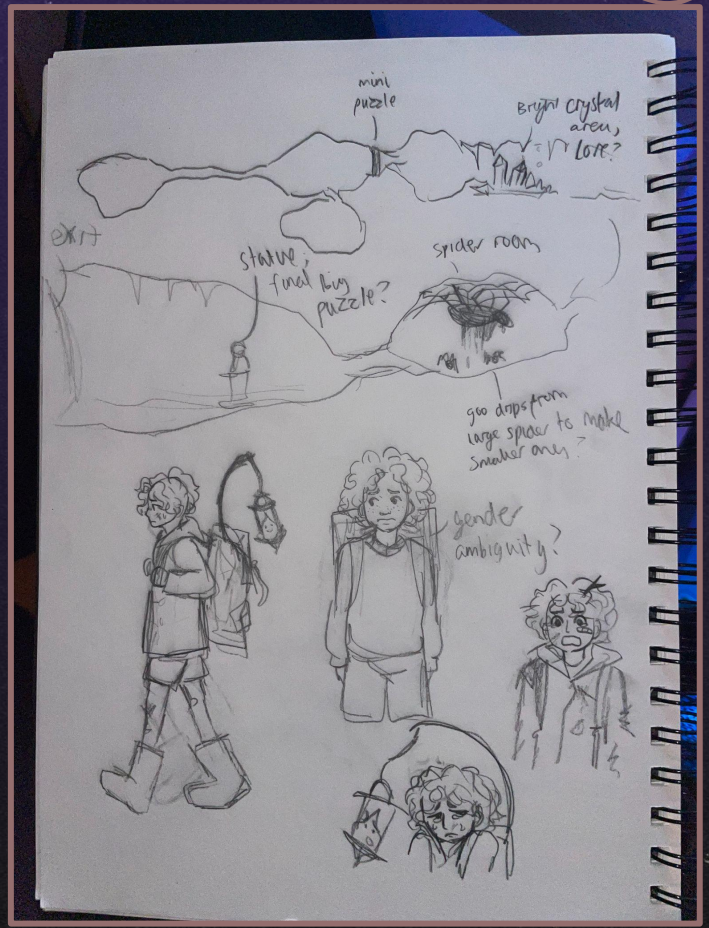
My Contributions

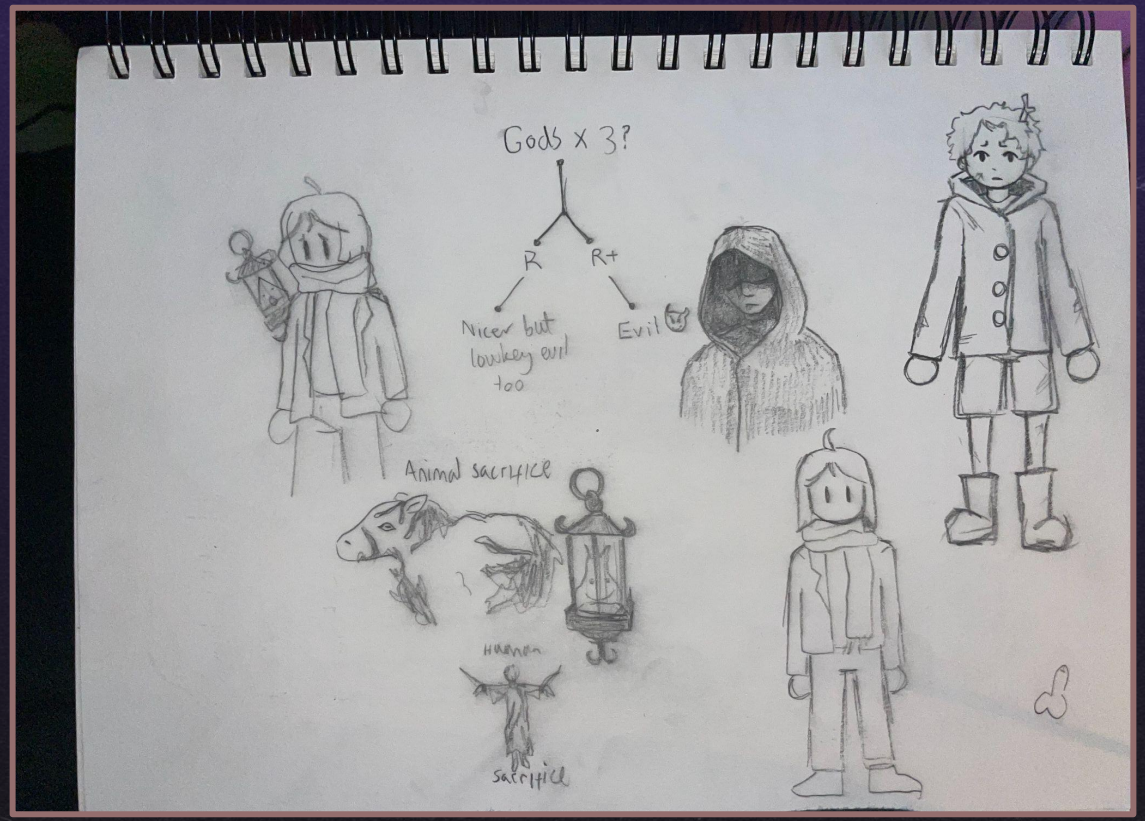
Since the last director's meeting I have been trying to refine the character design and monster designs through concept art. On Wednesday, we discussed the character design further and since then I've been working on a rough model based on the design so we can see what it could look like in 3D.



Lottie Hill


Kitchen
girl Bathroom
Library
Classroom 3
cafeteria
blocked off stairs
storage / Janitor's
classroom 2
boy Bathroom
cave entrance
enter
crawl
water
spikey
secret / puzzle


mini puzzle
Bright crystal area, lore?
exit
statue, final key puzzle?
spider room
goo drops from large spider to make smaller ones?
gender ambiguity?


Gods x 3?
R
R+
Nicer but lowkey evil too
Evil
Animal sacrifice
Human
sacrifice

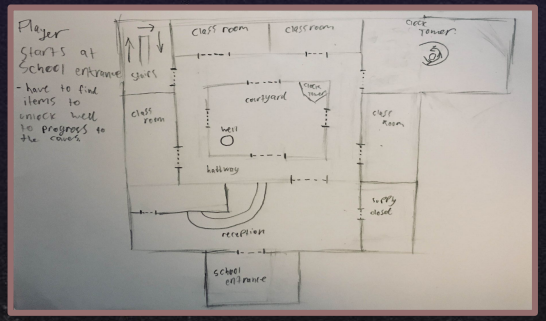
Rui Da Silva

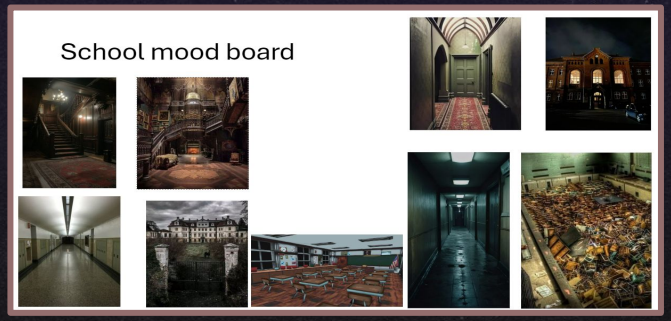
Designer - Narrative Design, Level Design, Environment Art

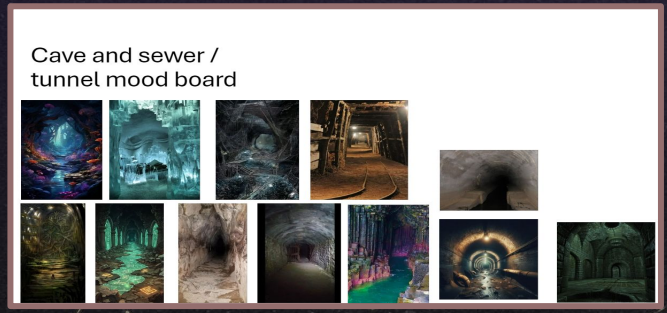
Level design

Since the last meeting I started working on the level layout of the school section of the game. I also created some mood boards to try and get a sense of how I want the level to look.

In addition to this I have been brainstorming ideas of how the levels can connect to each other, the types of puzzles the level could have and how the player will progress through the level.







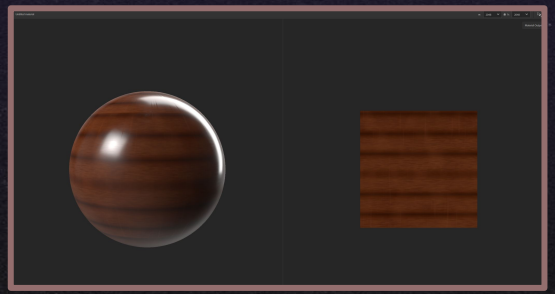
Rui Da Silva

Drew Magnetico

Designer - Texture Art

Texturing

- Creating textures with Adobe Substance Designer and Sampler



Zelda Echoes of Wisdom textures

Hades II colour palette

Try out this week

Approach:

1. **Illustrator** - concepts
2. **Photoshop** - normal and height maps
3. Designer - pattern/reusable textures
   Sampler - realistic details
   Painter - final details

Drew Magnetico

Roman Manzhelo

Designer - Character Art

CHARACTER DESIGN

Throughout this week, I have mainly focused on character design, trying to refine the ideas further and come up with some initial prototypes.

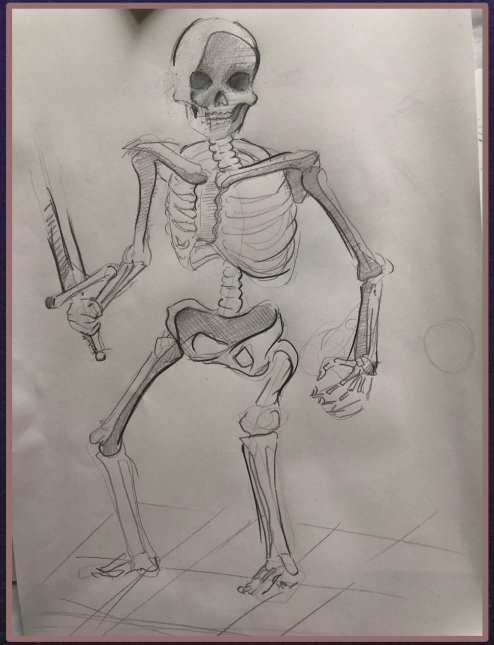
Roman Manzhelo

THE LIZARD

A creature that lurks in the cave level curls up into a ball and charges rapidly towards the player upon sight. The dodge mechanic would determine the player's chance of getting hurt. Its back is covered in sharp spikes, and its main body consists of hardened tar, forming the creature's scales and back. However, this mechanic could be simplified to favor texture simplicity and reinforce the idea that the creature's belly is its vulnerable part.

THE SKELETON

The Skeleton is another creature, a slower mob that patrols the school-cave areas. It has learned how to calcify its own matter to its advantage, mainly using its hands to form heavy weaponry. Its main body, a skeleton, is a vessel supplied by the science class of the school.



Ren Barrett

Designer - Interface Design,
Narrative Design, Level Design

Religion worldbuilding

This week I've been working on fleshing out the two religions present in the world of the game.

I wanted to have both religions develop from a shared source so with on of the two departing quite drastically from the base religion and the other staying a lot closer to the original although continuing to evolve on a separate path. I knew I wanted the base religion to be polytheistic and so I decided that the departing religion should be henotheistic mostly moving towards monotheism in order both to give some conflict. I also want to have the Henotheistic religion to be quite divided and split as it's going through a lot of growth and change finding its feet.

https://docs.google.com/document/d/12vXK5FnmWZM28DBpcZE7nx9dBWSj9ZIpFyD4q4xSyro/edit?usp=sharing

1100 words

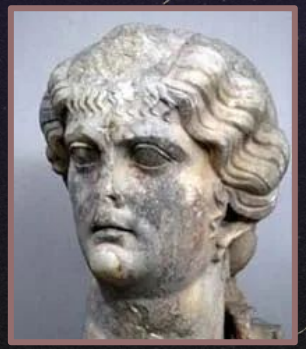
Polytheism - belief in multiple gods
Monotheism - belief in only one god
Henotheism/Monolatry - Worship of one god without denying the existence of other gods
Orthopraxis - Right practice and actions in prioritised
Orthodoxy - Right belief is emphasised

Ren Barrett

Religion worldbuilding

The Henotheistic religion will be more antagonistic than the polytheist one especially as it gets closer to monotheism with things like vandalism close to that of the crosses carved onto the faces of pagan deities in late antiquity.

I also am basing it of of atenism so it'll be more centralised and worship will be done through a religious authority as opposed to directly .



Influences:

Epicureanism

Orphism if we want more mystical things and for a good example of a mystery cult

Hermeticism is similarly mystical and also holds great importance in a particular figure (Hermes Trismegistus or Hermes thrice great) sometimes perceived as a human; this could easily be manipulated for a religion highly based in social and political control.

Atenism for a look at how a particular religion can be used as a social and political tool also for a very rapid change from polytheism to monotheism

Ren Barrett

Religion worldbuilding

As a group at the meeting on wednesday I proposed my ideas so far for the religions and also proposed a few questions so I knew I wasn't diverging too much from the rest of the group.

A few of the decisions we came to:

- The religions will be both more Orthodoxic with Orthopraxic elements (The polytheistic religion will likely be the most Orthopraxic religion out of the two).
- Death is considered final to keep the stakes high - no afterlife belief

Deities

The god of many hands

Omnibenevolence

Domains:

- Mercy/Judgement
- community building and restoration efforts
- Apotropaic (warding of evil)

The god of many teeth

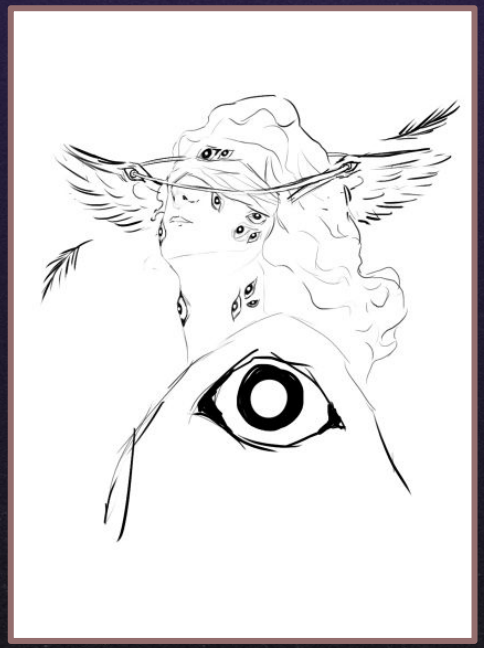
Omnipotence

Domains:

- Sickness / disease
- Base survival (particularly more brutal aspects),
- Destruction / Renewal
- Mutation / Change

Ren Barrett

Deities



The god of many eyes

Omniscience

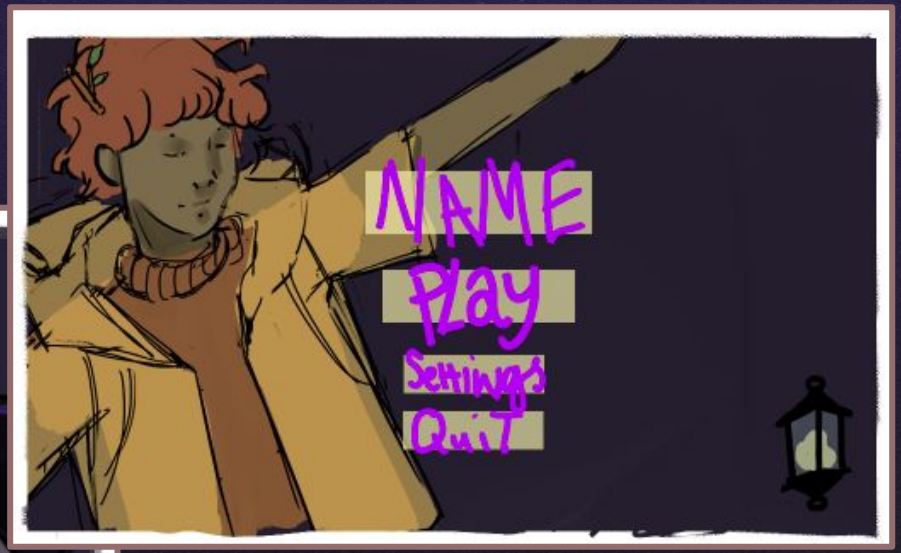
Domains:

- Preservation of knowledge
- Resource management (agriculture + hunting /scavenging)
- Prophecy / Divination
- shares the domain of judgment

The god of many eyes is also the current candidate for the god that moves into the henotheistic religion.
The idea behind that is the god of many eyes would be able to play into the idea of paranoia, religious control via wealth hoarding and tithes as well as control over knowlege

Ren Barrett

UI Concepts



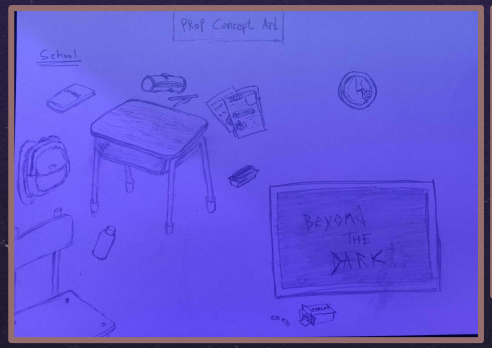
Jovi Travasso

Designer - Audio Design, Narrative Design, Prop Art

The Hustle

In this week, I was given a task to think of **concepts for props** for the game. This includes **props** for **different categories** such as environment, interactive props, and so on.

This is an image of my concept art for props that could likely be in the School region of the level we are planning on designing. I am currently working on prop ideas to add on to this scenario in the game and I will be presenting it soon.



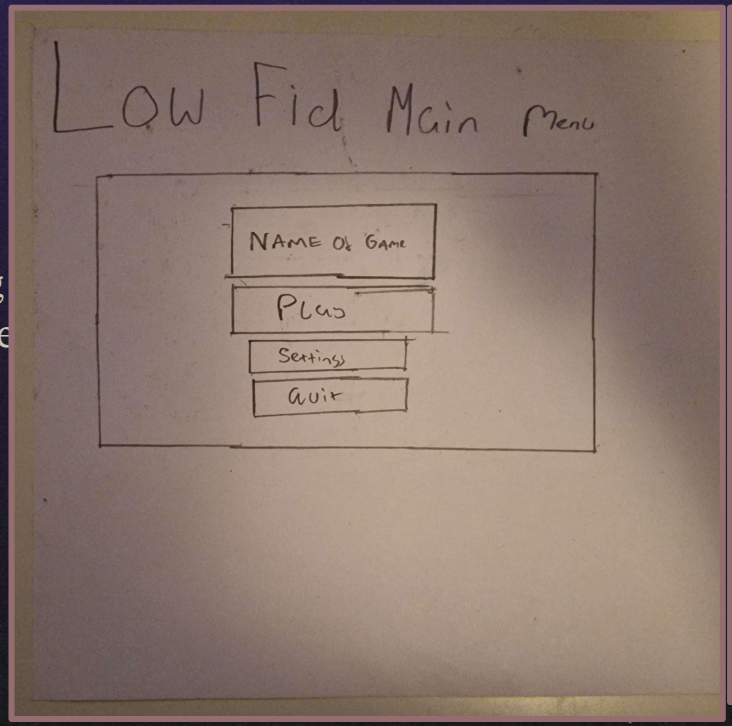
Jovi Travasso

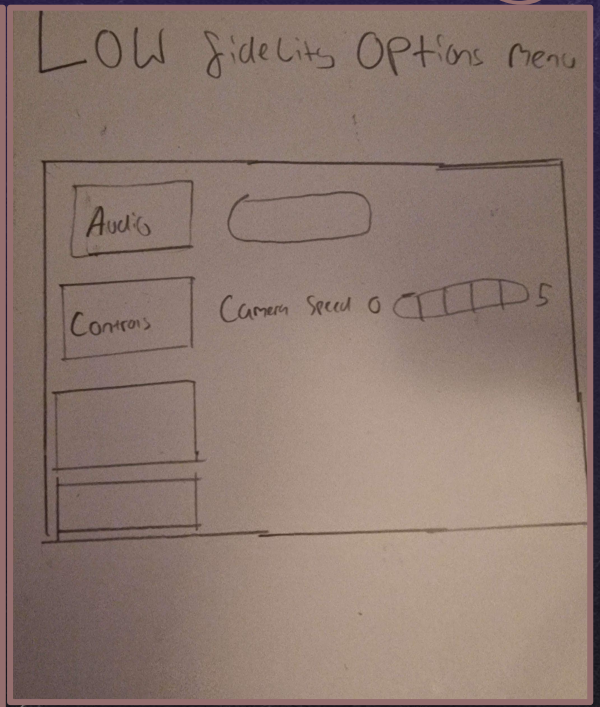
Serfiraz Sunmez

Designer - Interface Design, Prop Art

[UI Layout]

In this week, I designed the layout of the ui for the main menu and then the settings menu as well. Next week I am going to work on making the props and working on the User interface.





Serfiraz Sunmez

Samuel Collins

Architectural Programming,
Character Programming

Github and Unity

- Configured the **Github repository** (now ready for development initiation with full Asset folder structure in place)
- Currently researching and developing **branching strategy** (likely **Gitflow**)
- All team members are now **on-boarded** and have most cloned the repo for the first time
- Produced **four 15 min "workshops"** along with accompanying material for learning Git
- Delivered the **first** of those workshops
- Researched and produced a set of **coding standards/conventions for C# with Unity** development which will be referenced as an appendix in the **GDD**
- Developing proof of concept of central architecture elements like **Event Manager** (for higher level events) and **Mediator** (for storing/reading lower level (e.g. pos) attributes at high freq.)
- Produced several proofs of concept for **proposed core mechanics** for feedback/discussion

Samuel Collins

SESSION WORKFLOW (cont)

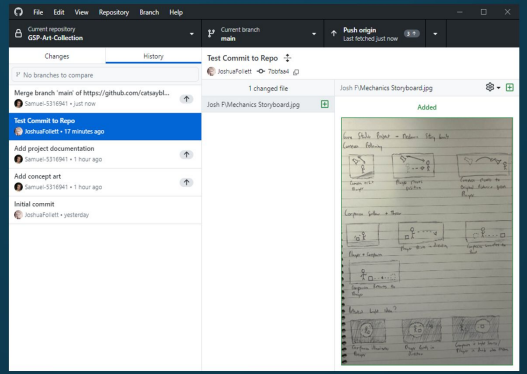
Now that you have **no local (uncommitted) changes**, and we are finished editing, we can **push**. Notice the little 2 next to **push origin**. That's because there are two **unpushed commits**. If you look at the history tab, your **unpushed commits** will have a little arrow next to them to signal that they aren't yet **pushed**.

Before **push**, we should **fetch origin**. There is a chance someone **pushed** changes since I last **pulled**.

Now I've **fetched**, notice how **push origin** has now changed to **pull origin** and a little 1 has also appeared. That's because someone did indeed make a **commit** and I am **behind**.

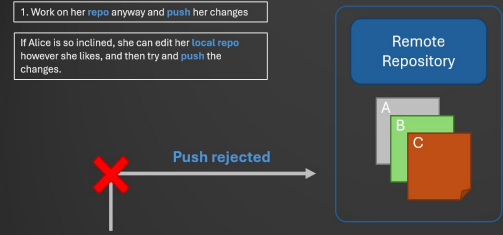
There are ways to see the **fetched** changes and compare them to your **commits**. Regrettably, GitHub Desktop doesn't provide that functionality. So in this scenario, you must blindly **pull** (which will merge them into your local branch) to get back on track.

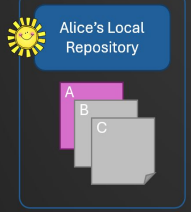
Josh's **commit** has been pulled into my local branch. This created a **merge commit** which merged Josh's **pushed commit** and my **unpushed commits** together. Luckily, we didn't touch the same files. If we did, we'd have to manage a **merge conflict**.



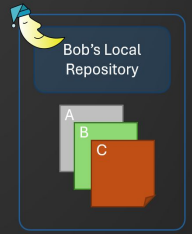
1. Work on her **repo** anyway and **push** her changes

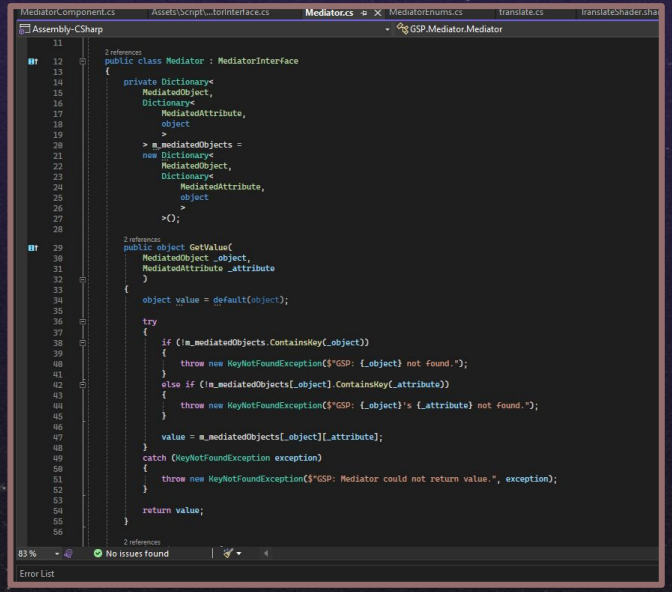
If Alice is so inclined, she can edit her **local repo** however she likes, and then try and **push** the changes.

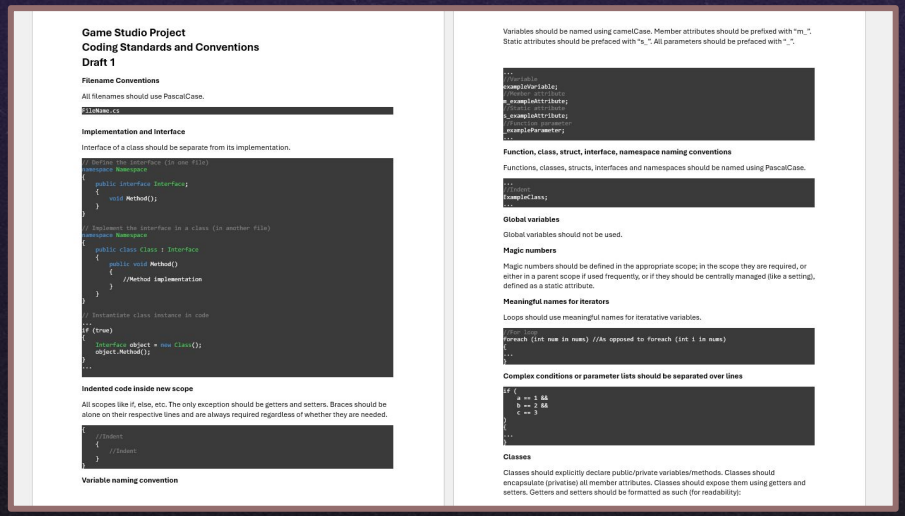




If Alice plans to work on File A, then when she **pushes**, Git will **reject** her **push**. This is different than a **conflict**; **conflicts** are sometimes painful but very allowed. This situation is not, so Git **rejects** it. Git recognises that the **remote repo** has **newer commits** from Bob that Alice lacks.







Game Studio Project
Coding Standards and Conventions
Draft 1

Filename Conventions

All filenames should use PascalCase.

```
FileName.cs
```

Implementation and interface

Interface of a class should be separate from its implementation.

```
// Define the interface (in one file)
namespace Namespace
{
    public interface Interface;
    {
        void Method();
    }
}

// Implement the interface in a class (in another file)
namespace Namespace
{
    public class Class : Interface
    {
        public void Method()
        {
            //Method implementation
        }
    }
}

// Instantiate class instance in code
...
if (true)
{
    Interface object = new Class();
    object.Method();
}
...
```

Indented code inside new scope

All scopes like if, else, etc. The only exception should be getters and setters. Braces should be alone on their respective lines and are always required regardless of whether they are needed.

```
{
    //Indent
    {
        //Indent
    }
}
```

Variable naming convention

Variables should be named using camelCase. Member attributes should be prefixed with "m_". Static attributes should be prefaced with "s_". All parameters should be prefaced with "_".

```
...
//Variable
exampleVariable;
//Member attribute
m_exampleAttribute;
//Static attribute
s_exampleAttribute;
//Function parameter
_exampleParameter;
...
```

Function, class, struct, interface, namespace naming conventions

Functions, classes, structs, interfaces and namespaces should be named using PascalCase.

```
...
//Class
ExampleClass;
...
```

Global variables

Global variables should not be used.

Magic numbers

Magic numbers should be defined in the appropriate scope; in the scope they are required, or either in a parent scope if used frequently, or if they should be centrally managed (like a setting), defined as a static attribute.

Meaningful names for iterators

Loops should use meaningful names for iteratative variables.

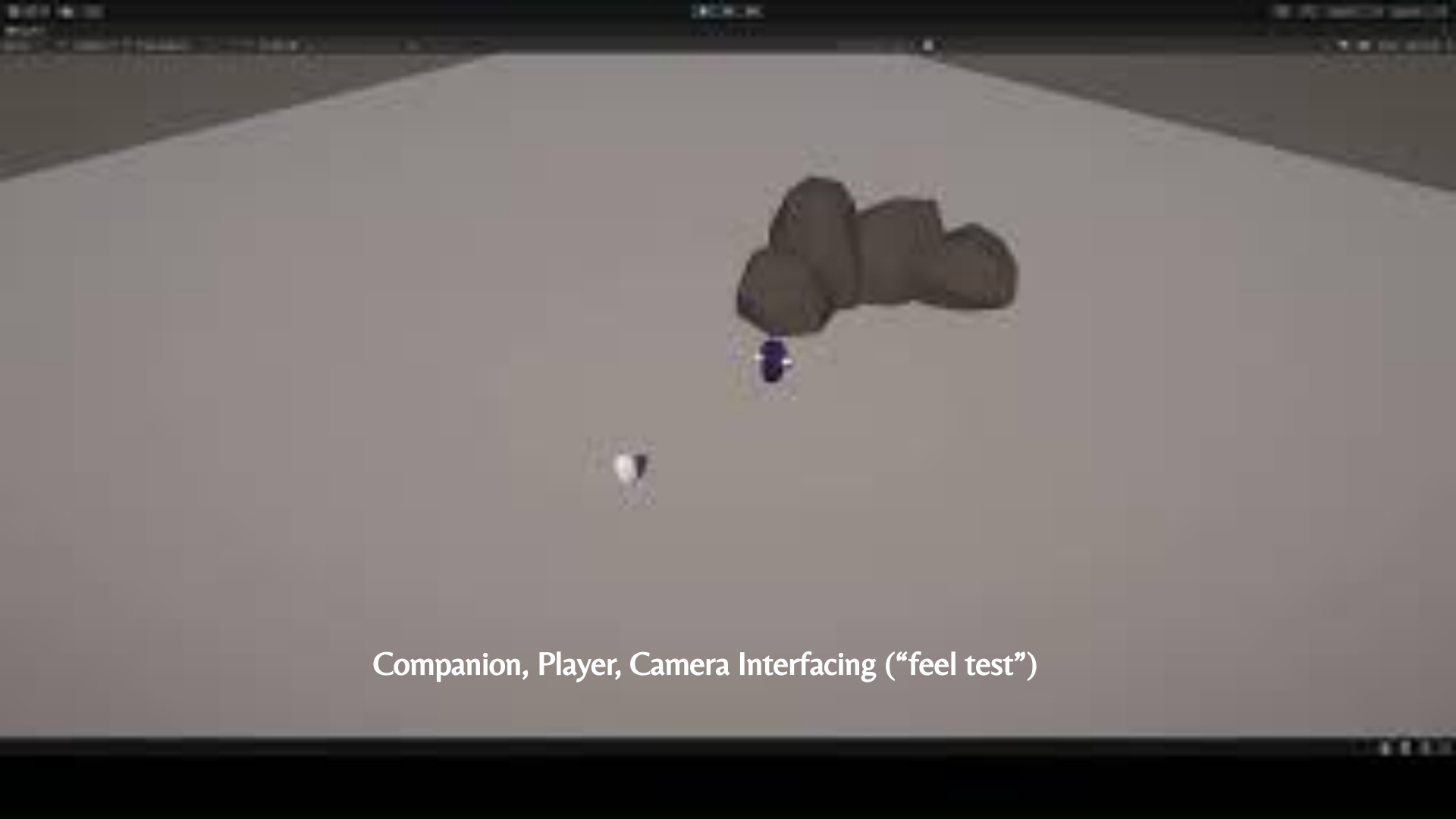
```
//For loop
foreach (int num in nums) //As opposed to foreach (int i in nums)
{
...
}
```

Complex conditions or parameter lists should be separated over lines

```
if (
    a == 1 &&
    b == 2 &&
    c == 3
)
{
...
}
```

Classes

Classes should explicitly declare public/private variables/methods. Classes should encapsulate (privatise) all member attributes. Classes should expose them using getters and setters. Getters and setters should be formatted as such (for readability):



Companion, Player, Camera Interfacing (“feel test”)

Companion Click + Drag Feasibility Tests

(please excuse jittery camera; I was actively working on it when I took the video!)

Goo Stretch Shader (proof of concept)

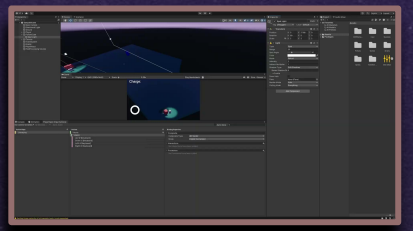
Light/Translate Shader (proof of concept)

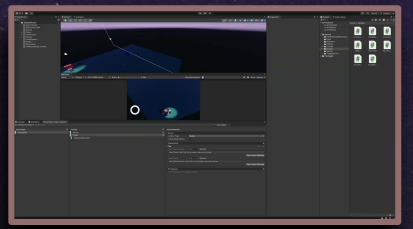
Joshua Follett

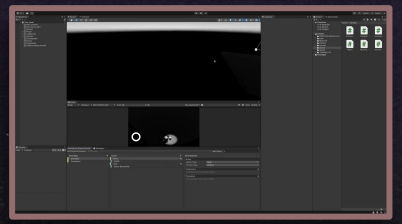
Gameplay Programming,
Character Programming

Gameplay and UI

- Fixed Movement to reduce stuttering
  - Rewrote with Unity Input Actions
- Tested Different Companion Mechanics
  - Example - Throw/Latching Companion + Light Source
- Added a Dodge Mechanic
- Implemented a Charger Meter UI
  - Simple Sprite Image with Scalable values
- Experimenting with Lighting, Post Processing and Contrast







Joshua Follett

Aims for Next Week

DESIGN AIMS

- Complete **level blockout** that reflects level discussion points
- Create **3D model blockouts** for characters, props and environmental assets
- Write up **full game idea** into a document
- Create **game pitch** ready for Friday.

TECHNICAL AIMS

- Continue to **prove programming elements** central to the architecture
- Continue to prove/iterate on **tech demos** of mechanics (and elicit/implement feedback)
- Deliver the other 3 **Git "workshops"** (Mon 03.03; Wed 05.03; Fri 07.03).
- Research and document our **Unity conventions** to complement our C# with Unity conventions