

Liminal Detective Technical Plan

Team Forsaken Hearts | Sprint 12
By: Samantha Shomo and Jeffrey Popek

TABLE OF CONTENTS

DIFFICULTY SCALING SYSTEM	3
Scaling Overview	3
Delivery Platforms	4
Windows 10 and Mac OS	4
Development Environment	4
Game Engine	4
Unreal Engine 5	4
Tortoise SVN	4
Art Engine	5
Maya	5
Zbrush	5
Substance Painter	5
Logic Pro x	5
Logical Flow Diagram	6
Game Mechanics and Systems	6
Combat	6
Pipelines	11
Art Pipeline	11
Programmer Pipeline	12
Design Pipeline	14
Dialogue/Writing Pipeline	14
Audio Pipeline	15
Sprint Updates	15

DIFFICULTY SCALING SYSTEM

Scaling Overview

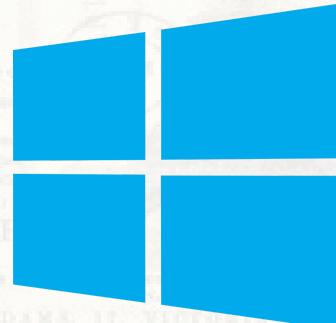
The risk scaling system for this document will go through 1 to 5. This will be represented using the flowers shown below:

	The team has no problems with implementation.
	Easy features with minuscule problems
	Will be harder to implement and may take a couple of hours
	Will take up a lot of time to implement and may be out of scope for the project. Multiple bugs would arise from the implementation
	Near impossible and extremely out of scope

Delivery Platforms

Windows 10 and Mac OS 

Liminal Detective will be developed for Windows and Mac computers as it is the easiest platform we can port the game to. Unreal Engine can create games that work on both operating systems so this is a pretty easy task to complete.



Development Environment

Game Engine

Unreal Engine 5 

Our game will be built in Unreal Engine 5.3. Our designers and artists are both proficient in using Unreal. One programmer also knows Unreal well, but the other programmers have never worked with it before. In order to make up for our shortcomings, we plan on using tutorials as well as having our fellow team members help us learn.

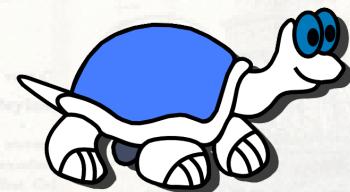


5

Tortoise SVN



For our version control, we will be using Tortoise SVN because Git cannot be used with Unreal. Everyone has used it aside from the artists. Those who know how to use it will help those who do not.



[Tortoise SVN](#)

Art Engine

Maya



For modeling, our artists will be using Maya. They both have extensive experience in the software. The models will then be exported to Zbrush for sculpting.

Zbrush



After models are created in Maya, they will be exported to Zbrush for sculpting. Our artists have been working with this for about a year so the risk is low. Once the sculpting is done, it will be sent to Substance Painter for texturing.

Substance Painter

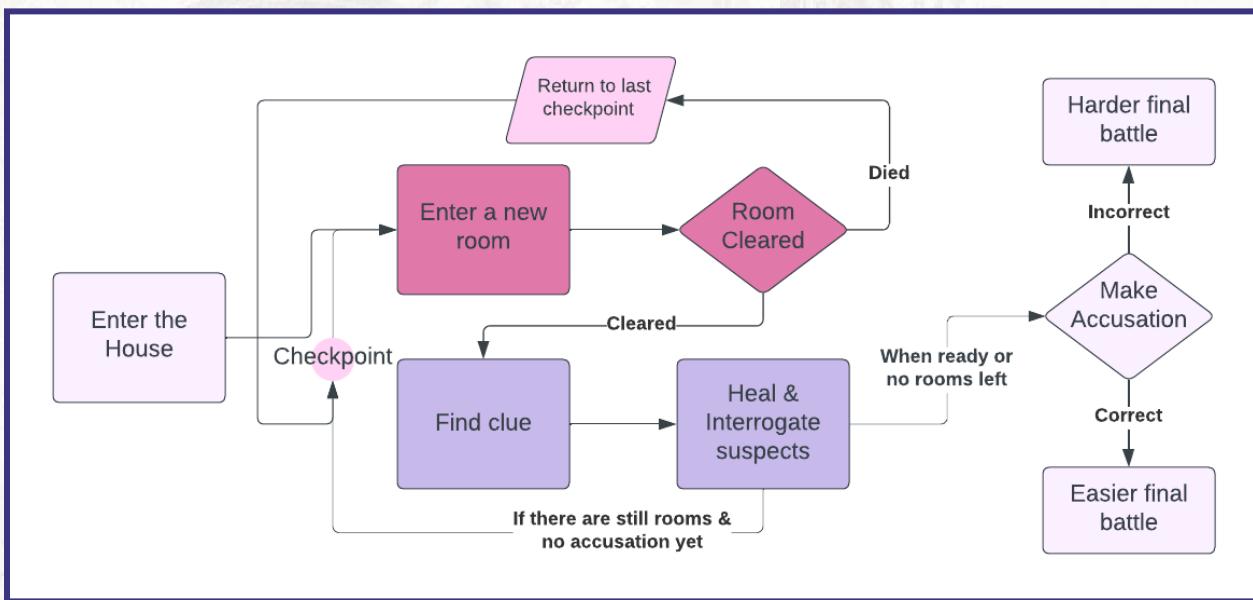


Substance Painter is used for texturing models. Similarly to Zbrush, the artists have at least a year of experience working in this software.

Logic Pro x 🔎

Logic Pro X is a professional digital audio workstation (DAW) developed by Apple for macOS. It offers a large set of tools for music production, recording, editing, mixing, and mastering. Logic Pro X is widely used by musicians, producers, and audio engineers to create high-quality music and audio projects.

Logical Flow Diagram



Game Mechanics and Systems

Combat

- The player can perform 2 different attacks, an auto attack and a grab-and-throw attack.
- Auto Attack

7

- The auto attack will fire a projectile when pressing left click. The player has 3 shots before you must manually reload.
 - When a projectile hits an enemy it will mark them. An enemy can be marked multiple times.
 - On the reload, all enemies marked will take additional damage.
 - A projectile will fly off of marked enemies back to the player and deal damage to any enemies hit along the way.
-
- Grab and Throw
 - The grab-and-throw attack allows the player to pick up an enemy that is in their melee range.
 - The player can only grab an enemy when at full stamina.
 - To grab an enemy, hold down right-click while looking at an enemy in your melee range.
 - To continue holding an enemy just keep holding right click.
 - While holding an enemy your stamina will consistently decrease. When the player reaches 0 stamina the enemy will be released without being thrown.
 - When an enemy is being held it cannot attack the player.
 - To release and throw an enemy release the right-click button. The enemy will be thrown towards the direction the player is looking.
 - Thrown enemies can hit other enemies which will knock the hit enemies back.
 - The player can throw world objects to damage enemies or break other world objects.
 - All stamina rules apply to grabbing and throwing world objects.

Enemies

Enemies will have three different states they will transition through searching, active, and attack.

- Searching State
 - When enemies are in the searching state they will wander around the room every 2 seconds until the player enters their line of sight which will put them into the active state
 - An enemy will not leave its room that it's searching
- Active State
 - When they are in the active state they will move towards the player. If the player is in their attack range the enemy will move to the attack state
- Attack State
 - When an enemy enters the attack state they will perform an attack on the player
 - The enemy will see where the player is at the moment, and then after a short delay it will jump toward the spot where the player was
 - After the attack ends the enemy will enter the active state again unless the player is still in the attack range
- One enemy per room is holding a clue. When that enemy is killed, the clue is dropped
 - This enemy is highlighted when the player presses F to focus

Boss Fight

- When the player accuses who the killer is, they will be transported to a new room to fight the killer
- The boss is a larger and slower-moving enemy
- The player cannot grab the boss
- The boss does more damage than normal enemies

9

- The boss will spawn smaller enemies at the start of the fight and when it reaches half-health
 - The smaller enemies are the same as the base enemies except for their lower health
 - These enemies spawned by the boss can be grabbed

Mystery

Liminal Detective has multiple mechanics to aid the player in solving the mystery

- Telephone
 - The telephone can be accessed in each room after combat
 - The player can talk to suspects, make an accusation, and heal
- Clues
 - The player will need to find the clues to progress the mystery
 - Once a clue has been found it will be saved to their files which can be viewed again
 - The clues are scattered throughout the mansion held by enemies
- Files
 - The files store vital information for solving the mystery such as clues and interactions with suspects
 - The files store a suspect's motive (or clear their motive) and alibi
 - Your files will update when you gain new information from the telephone or a clue
- Focus
 - The player can press and hold F to zoom in and highlight enemies that hold a clue or the clue itself

Narrative

The narrative portion of our game is based on questioning the suspects based on new clues found

- Questioning

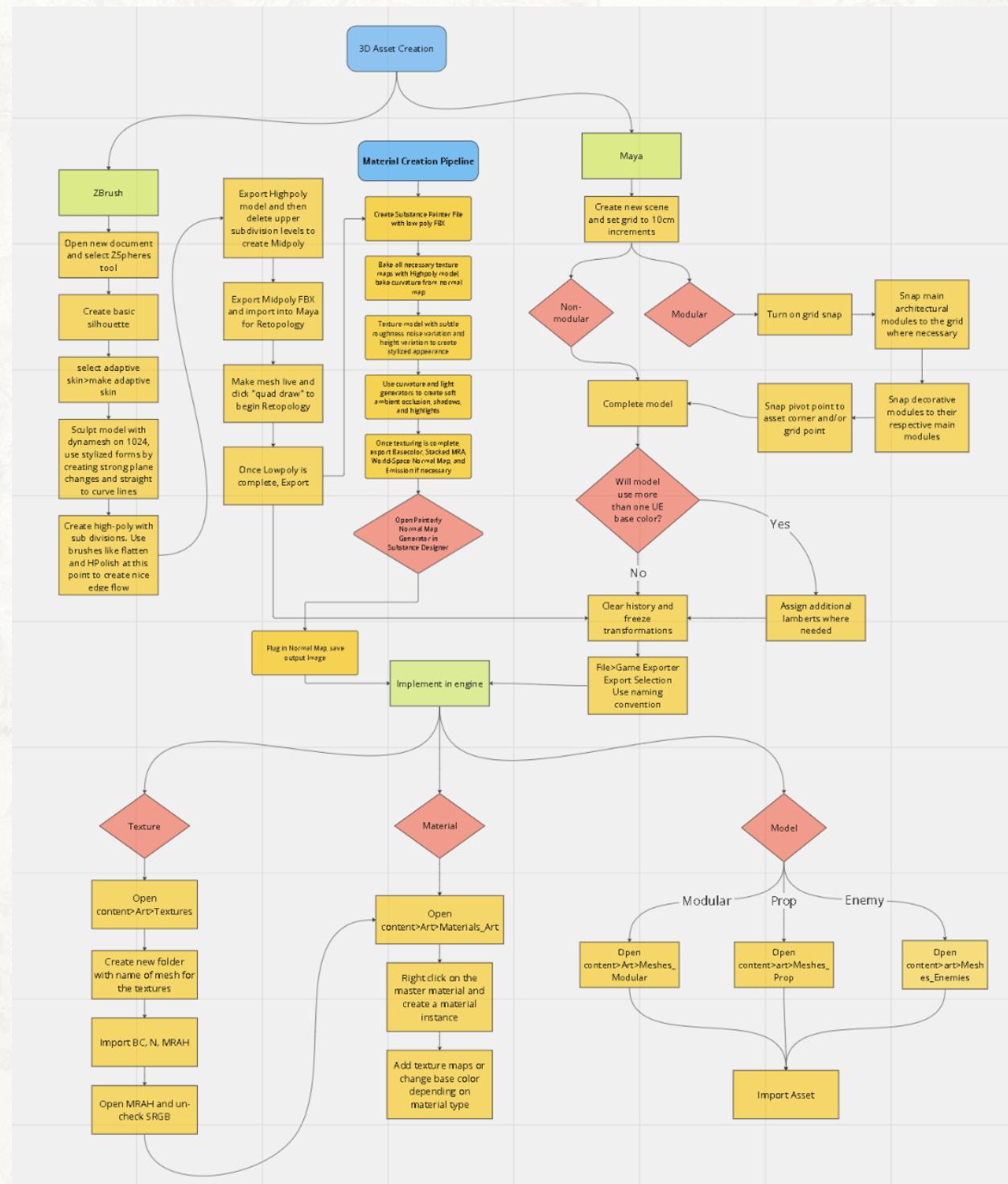
- The player will start the game by asking questions to each suspect through the telephone
 - After a new clue is gathered, new questions will appear when talking to those characters again
- New questions will only appear for suspects that have a connection to the newly collected clue. Each clue is connected to only two suspects

- Final accusation

- The player can try to accuse one of the 4 suspects as the killer once they have collected one clue through the telephone
- If they are wrong or right they will have a disadvantage during the final boss fight
- If the player accuses the wrong suspect without all the clues then the disadvantage in the boss fight will be even greater
- The game will warn the player about accusing a suspect when they don't have all the clues.

Pipelines

Art Pipeline



12

Asset creation in Maya, Houdini, Substance, ZBrush, etc

1. File naming convention [file_name]
2. For textures: combine(?) textures, should end up with 3 maps, MRAH, normal, base color
3. For models: it may be necessary to assign more than one material to a mesh
4. Export to [folder]

Import into build

1. Open the project
2. Open correct folder
3. Click Import
4. Always open MRAH textures and un-check SRGB

Programmer Pipeline

Getting a Mechanic

First, our designers will come up with an idea and discuss the scope with the team. If the team decides it is within scope then

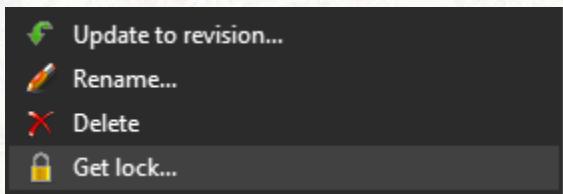
the programmers will start dividing the work into smaller tasks. Once the tasks are all created then the programmers can start working in the engine.

Implementation

1. The designers will come up with an idea they want implemented. They must first run it through with the team to see if it is in scope for this project.
2. If in scope then the programmers can start dividing the work into smaller tasks. If the idea is out of scope you can scrap it or try and rework it to be in scope, repeat from step one when reworking your ideas.
3. Update your SVN repo to make sure your project is up to date. If you have any repo issues stop and fix them.

13

- Once your project is up to date, you must lock the files you want to work on and post which files are locked in the appropriate mattermost channel. Only after you lock your files can you start working.



- Once you are finished working you can commit all the files you changed. The locked files will unlock on their own.

General Naming Conventions and File Structure

Adding different subfolders and folders based on the needs of the project is fine with clear communication to the team as a whole. Trying to keep things organized is very important. Here are the naming conventions to continue to help with that organization;

Format: (NamingAcronym_Filename)

Blueprints: BP_Filename

Widget Blueprint: WP_Filename

Inkpot Stories: IS_Filename

Input Actions: IA_Filename

Meshes: SM_Filename

Materials: M_Filename

Material Instance: MI_Filename

Music: MU_Filename

Niagara Effects: NE_Filename

Static Mesh: SM_Filename

SFX: SFX_Filename

Textures: T_Filename

14

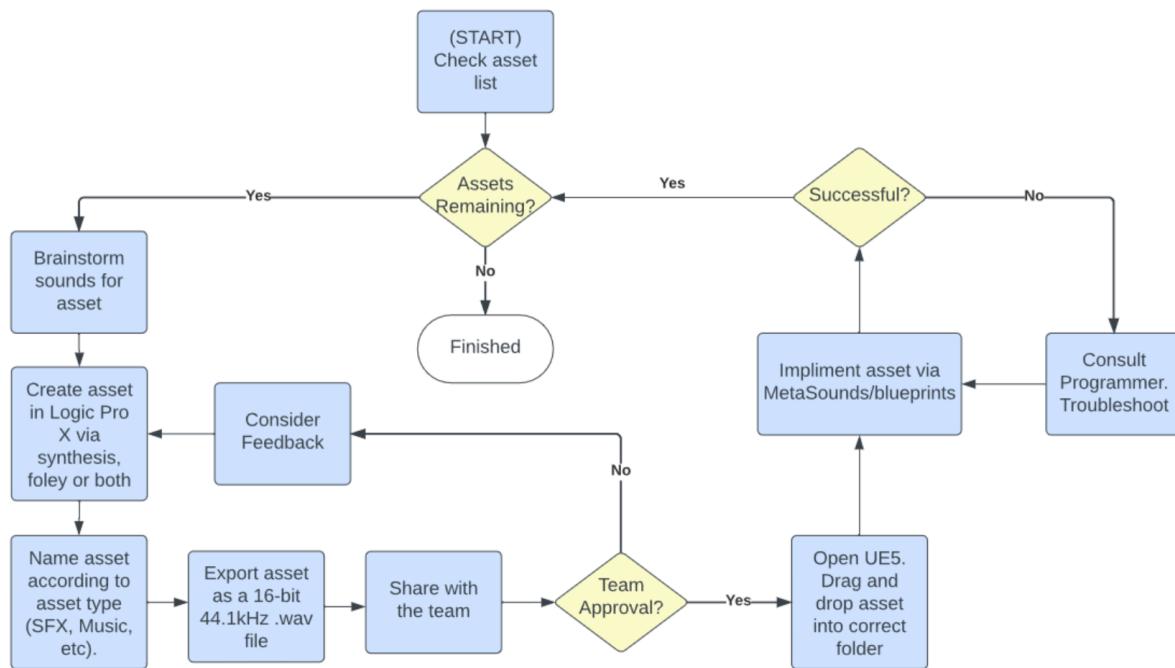
Design Pipeline

1. The idea exists - "I think this should be in the game"
2. Discuss ideas, usually in person at Friday design meeting
 - a. Discussions can also take place virtually if can't be in person
3. Concepting meetings are done in person
 - a. If an idea comes up outside of it, text other designers and the idea is marked down for discussion during the next meeting.
4. Send a draft of the idea to the team for feedback.
 - a. Need full designers and other team members' approval.
 - b. Hold off on further development of ideas until you get buy-in and discussions on what to do next.
5. Create or update official documentation.
6. The team reviews and approves documentation.
7. Start prototyping, either in engine or through necessary development tools.
8. Test prototype and get feedback.
9. Find questions and topics to discuss based on feedback.
10. Create ideas from new discussions.

Dialogue/Writing Pipeline

1. Concept dialogue idea
2. Discuss narrative ideas at design meetings, or team meetings
3. Pull the repository and update Ink story
4. Edit Ink story
 - a. Only edit Knot that is relevant to the specific dialogue branch
 - b. Add a new line every time there is a line break in the text experience
 - c. Review Ink documentation for formatting information
 - d. Create and bind new External Functions as needed, for connections between dialogue and outside game
5. Save Ink file

Audio Pipeline



Sprint Updates

Sprint 1

During this sprint, we came up with multiple game ideas during our team meeting. We narrowed it down to two games which were Heartsword and Liminal Detective. We decided to prototype both

games in Unreal Engine 5. This sprint we mainly focused on getting the two prototypes working so we did not finalize any documentation yet.

Sprint 2

16

During this sprint, we started organizing our team into two groups for both of our games. We started getting both games into Unreal with a few features.

Sprint 3

During this sprint, our main goal was to pick a game concept to move forward with after the green light. Before our weekly meeting, our two subteams continued to work on the prototypes without putting too much work into documentation. During our weekly meeting halfway through the sprint, we discuss which concept should go forward. The team decided on going with Liminal Detective because of everyone's buy-in to the idea. After this meeting, the main goal of this sprint was switched to getting a minimal viable product (MVP) working for our greenlight presentation at the end of this sprint. For the rest of the sprint, the team merged and began focusing all of their attention on the MVP and assets for this concept.

Sprint 4

During this sprint, we wanted to continue working on our MVP and polish it for the formal greenlight presentation at the end of the sprint. The designers and programmers focused on getting the mechanics that were already in the build to work better.

Sprint 5

This sprint was spent polishing the mechanics of our game and implementing art. We got Inkpot working in our game but not completely implemented into our main game loop. Our artists were able to complete models and put them into the repo. Our programmers polish enemy AI. We also started rigging our demon enemy model.

Sprint 6

This sprint was spent getting art and sound assets into the engine, continuing work on art and sound assets, rigging the enemy model, polishing the combat mechanics, implementing the chaos engine, and organizing the repo and confluence.

Next sprint we want to focus on code refactoring, continuing work on art and sound assets, continuing rigging, and implementing more assets into the engine.

Sprint 7

This sprint was the sprint during spring break, so our team spent most of it taking a break. After the break our team focused on getting the PC modeled, re-organizing code, iterating on shaders, continuing writing dialogue, continuing rigging, and continuing iterating on the music.

Next sprint we want to continue focusing on code refactoring, getting art assets implemented, and getting all features we want in the game implemented before feature lock.

Sprint 8

This sprint is the sprint before feature lock so our team prioritized getting our final mechanics into the build even if they were not perfect. We focused on getting certain enemies to drop clues, grabbing and throwing objects, and clue enemies to be highlighted. We also continued work on character and environment assets as well as sound assets. A lot of quality-of-life changes were added since we thought we were going to be testing at admitted students' weekend.

Next sprint we want to polish and bug-fix our mechanics and continue implementing art assets.

Sprint 9

During this sprint, we got all our documentation up to date for the documentation review and completed the rig of the enemy.

We implemented features to improve user experience such as a pause menu and tutorial to learn how to play the game, a final boss battle when the player makes an accusation, art assets, and post-processing effects. We created our step three presentation, decided on T for our ESRB rating, and created our pitch.

Next sprint we want to completely shift our focus to bug fixing and polish.

Sprint 10

During this sprint, we did lots of bug testing and fixing. Our characters were finally rigged and we improved our performance by leaps and bounds. We also finally added the main menu.

Sprint 11

This was the sprint we went all in on fixing all bugs. We took our game to testing twice just to find which bugs were the most urgent to fix. Lots of art assets were implemented and our player character got animated!

Sprint 12

This sprint we just made the finishing touches to our game. We made sure all our documentation was complete. We tried to rename the game but nobody could come up with something better than Liminal Detective.