

SLASKI, Jack

S013671M@student.staffs.ac.uk

**Feudal Living**

Technical Design Document

Contents

[Project Introduction 2](#_Toc1836881130)

[Project Goals 3](#_Toc1004620091)

[Challenges and Risks 3](#_Toc782404623)

[Hardware Requirements 3](#_Toc2117671363)

[Platforms 3](#_Toc1515692660)

[Target Platform 3](#_Toc1650599673)

[Engine Summary 3](#_Toc602409686)

[Mechanics and Diagrams 3](#_Toc1248574988)

[Main Level Game Mode 4](#_Toc2083892037)

[Player Controller 4](#_Toc1525863297)

[Camera Pawn 5](#_Toc255880846)

[Needs 6](#_Toc1866599105)

[Skills 6](#_Toc861674520)

[Interactable Objects 7](#_Toc300337603)

[Character Data 8](#_Toc305611259)

[World Time 9](#_Toc2130785734)

[Working / Doors 9](#_Toc1881927138)

[Interactable Manager 10](#_Toc696992178)

[AI Takeover 10](#_Toc350196839)

[UI 10](#_Toc1712549454)

[Coding Standards 11](#_Toc746095545)

[Programming Standards 11](#_Toc478137008)

[Style Guide 11](#_Toc1524154209)

[Commenting Rules 11](#_Toc221902668)

[Code Review Procedures 12](#_Toc1763510126)

[Production Overview 12](#_Toc1929319866)

[Moscow 12](#_Toc823969880)

[Timeline 13](#_Toc564295167)

# Project Introduction

The idea for this game is that it will be a ‘Sims-like’ medieval life simulator named “Feudal Living”. In this game, the player will take control of a character who is living in medieval England and attempt to control their life by managing their needs, improving their skills, earning them money, and designing their home.

## Project Goals

The ultimate goal for this project is to have a solid and functional foundation for a medieval life simulator game. This foundation could then act as a base for possible future expansion and project creation. By having a solid foundation that is modular, easy to understand, and efficient, future work and expansion can be made very easy.

## Challenges and Risks

The challenges this project is aiming to solve are as follows:

* Having modular interactable objects that can easily be created to solve multiple needs and skills.
* Having and AI system that roams and interacts with objects on their own.
* Having a way to earn and spend money.
* Spawning, building, placing, selling, and moving interactable and decorative objects around the world.
* Having multiple different types of objects.
* Having time controls that can stop, play, and fast forward time.
* Having a custom world time that gives the user some more world context.
* Having a realistic environment, objects, needs, and skills for the time period.
* Easy to use and concise camera and player controls.

## Hardware Requirements

Recommended Hardware:

Operating System: Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher

Processor: Quad-core Intel or AMD, 2.5 GHz or faster

Memory: 32 GB RAM

Graphics RAM: 8 GB or more

Graphics Card: DirectX 11 or 12 compatible graphics card with the latest drivers

# Platforms

## Target Platform

The target platform for this project is PC as the project's controls can be complex and mostly requires a mouse cursor as well as multiple keyboard controls. Although, in the future a console version could be made through the use of a console cursor and smart console control mapping.

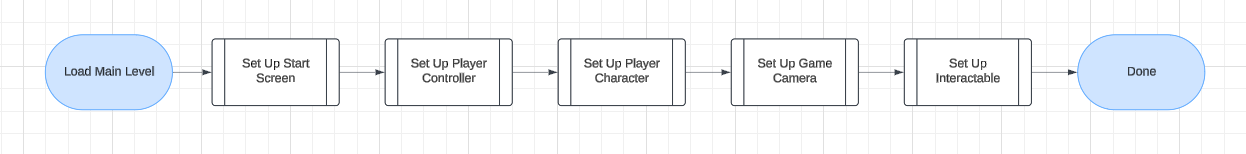
## Engine Summary

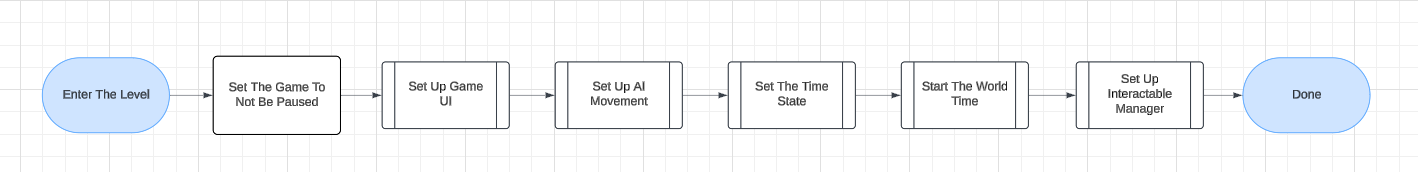
This project was created in Unreal Engine Version: 5.5 with no additional plugins used.

# Mechanics and Diagrams

## Main Level Game Mode

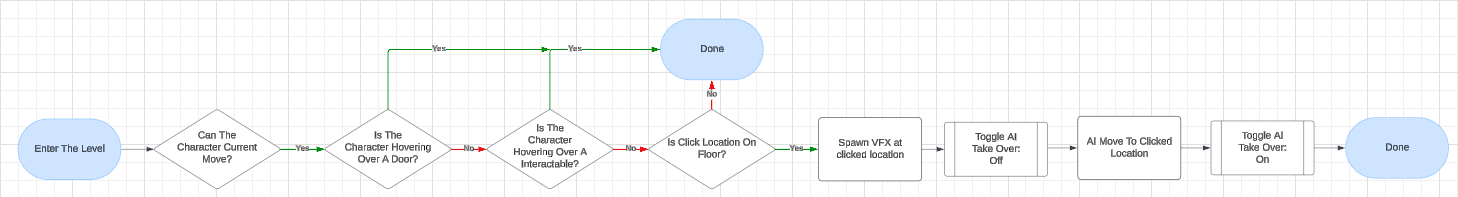
Here is a flowchart showing the logic of the main game mode loading the main level:

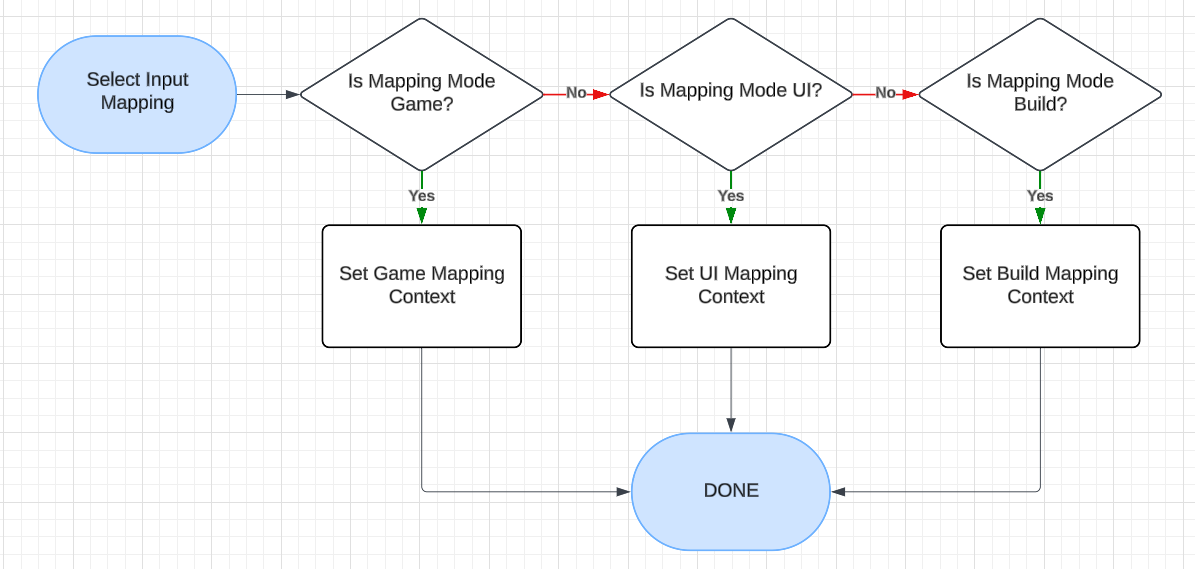
Here is a flowchart showing the logic of entering the main level in the game mode:

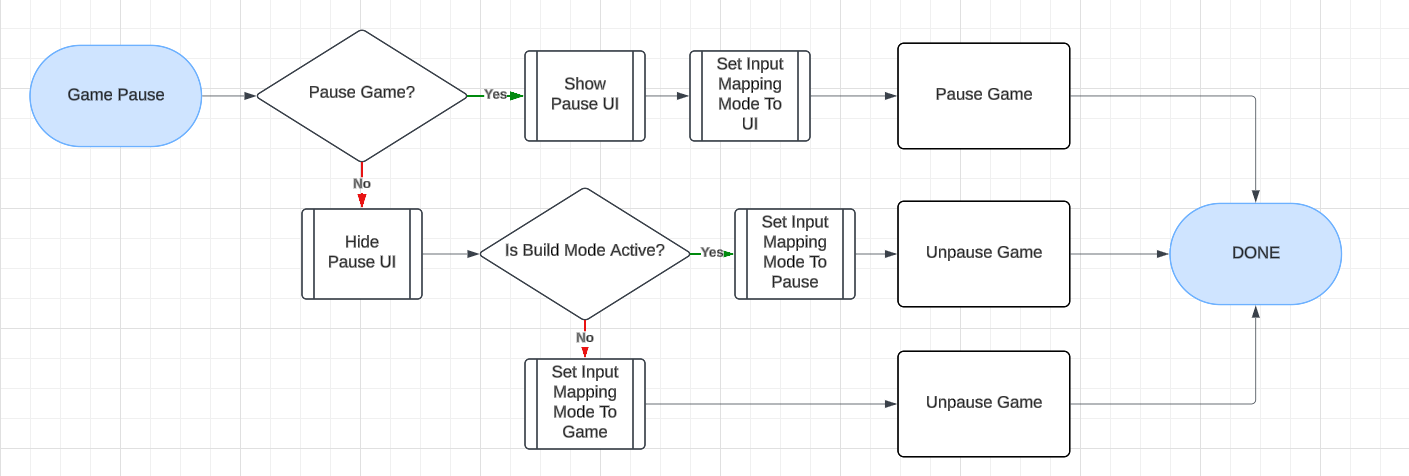


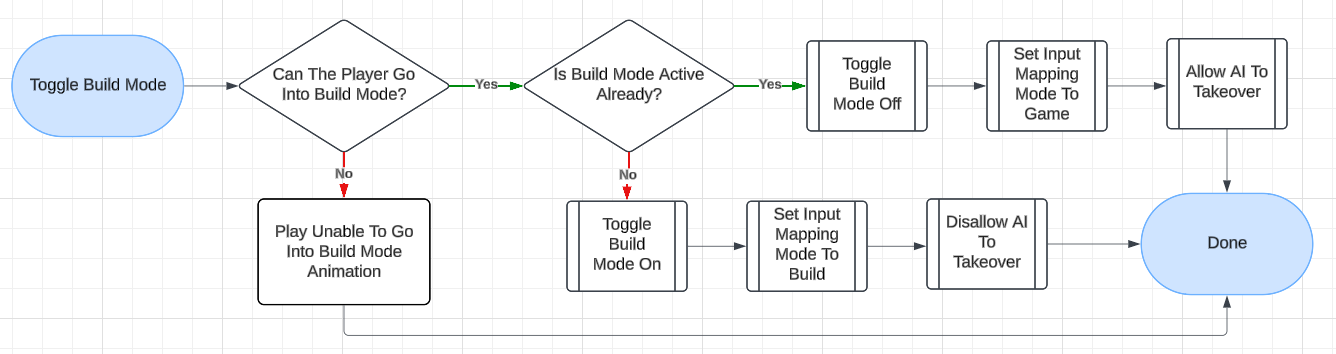
## Player Controller

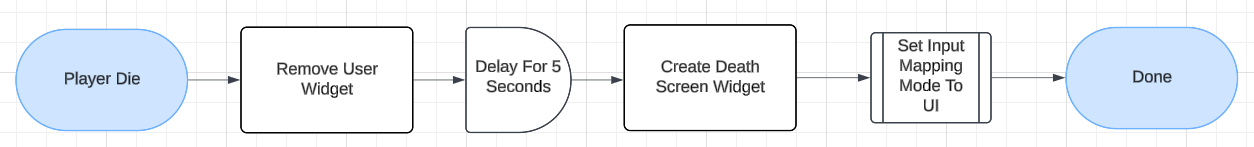
Here is a flowchart showing the logic of a character clicking in the world:

Here is a flowchart showing the logic of selecting an input mapping context:

Here is a flowchart showing the logic for pausing the game:

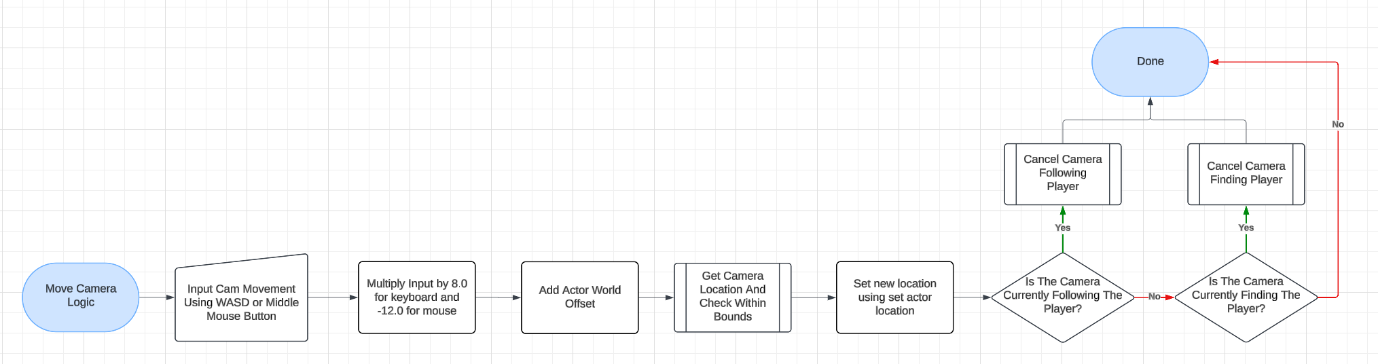
Here is a flowchart to show the logic of toggling build mode:

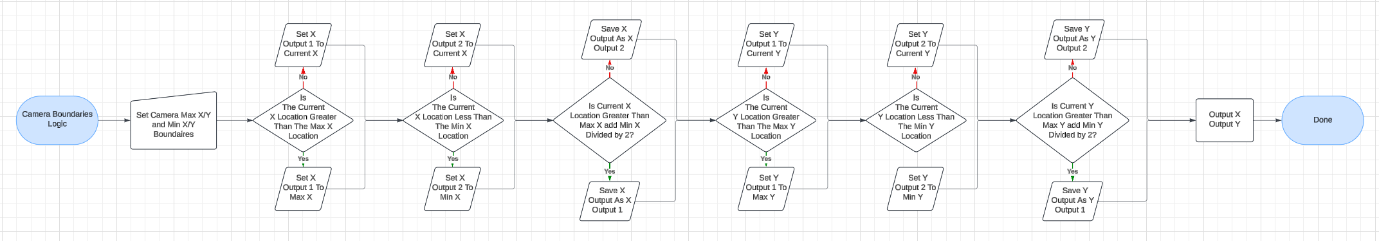
Here is a flowchart to show the logic of a player dying:

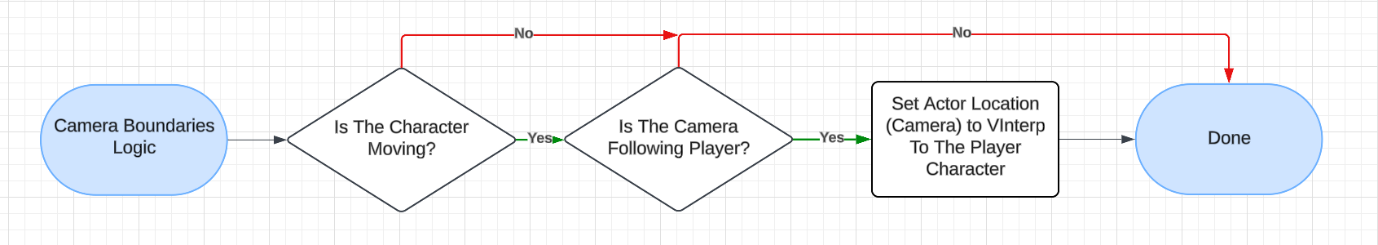


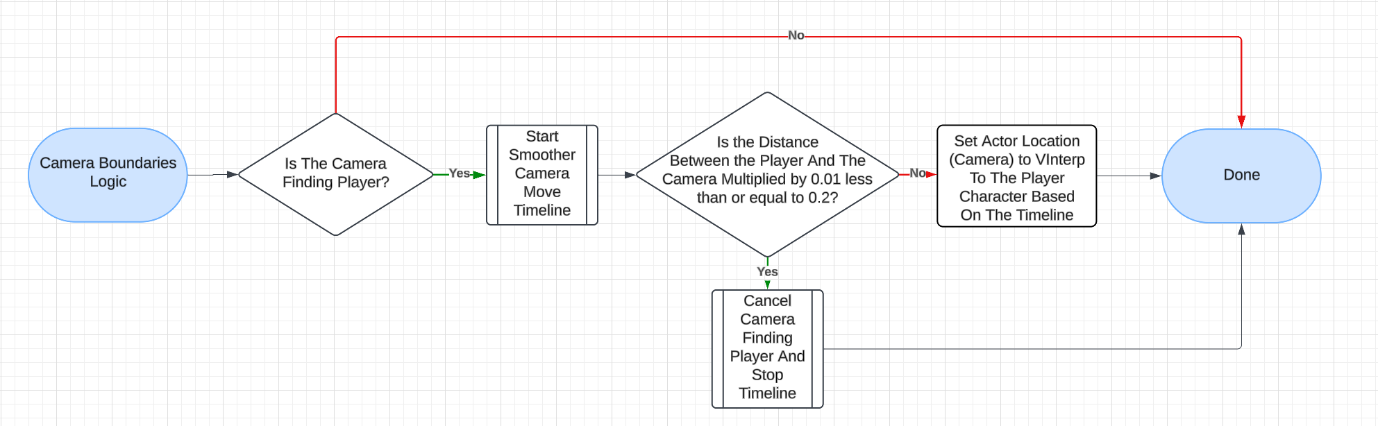
## Camera Pawn

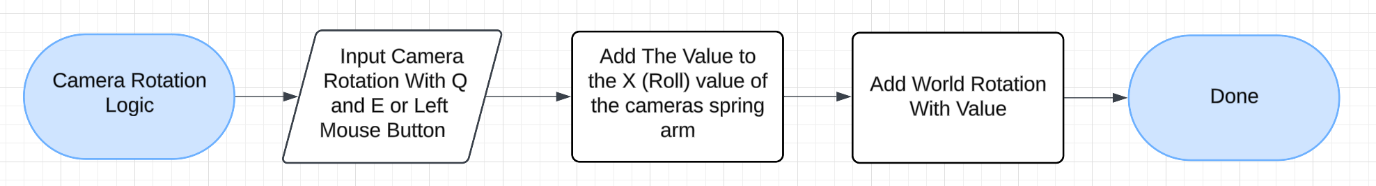
Here is a flowchart showing the camera movement logic:

Here is a flowchart showing the camera boundaries logic:

Here is a flowchart showing the camera follow player logic:

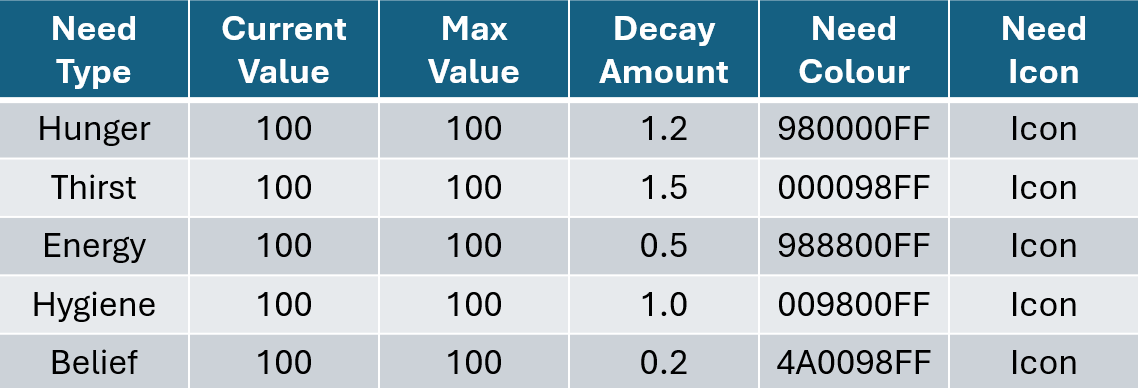
Here is a flowchart showing the camera find player logic:

Here is a flowchart showing the camera rotation logic:

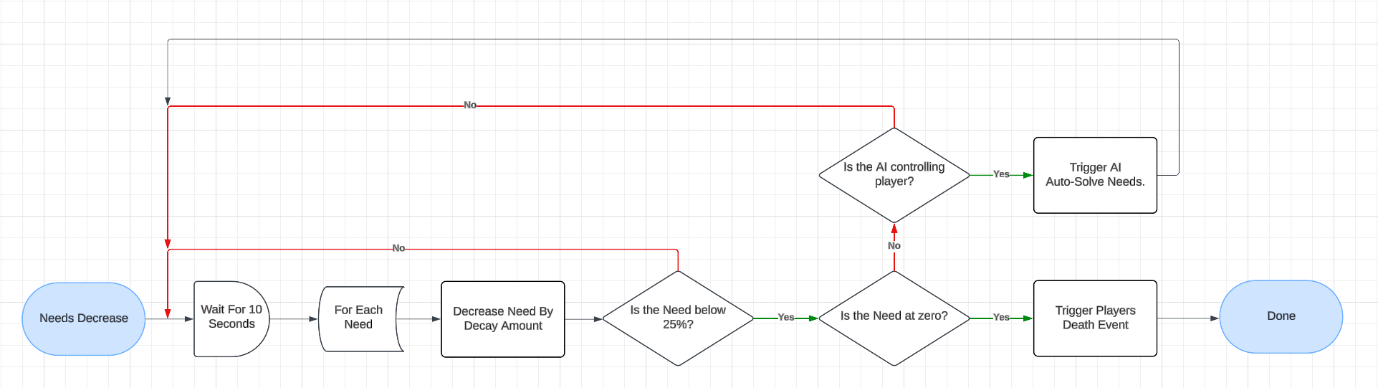


## Needs

Here is a table showing all of the in-game needs and their values.



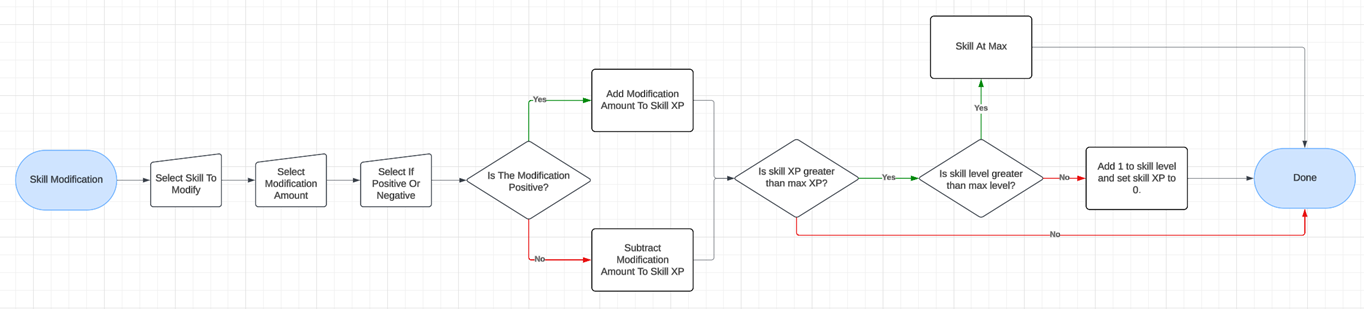
Here is a flowchart showing the process of the characters needs decaying:



## Skills

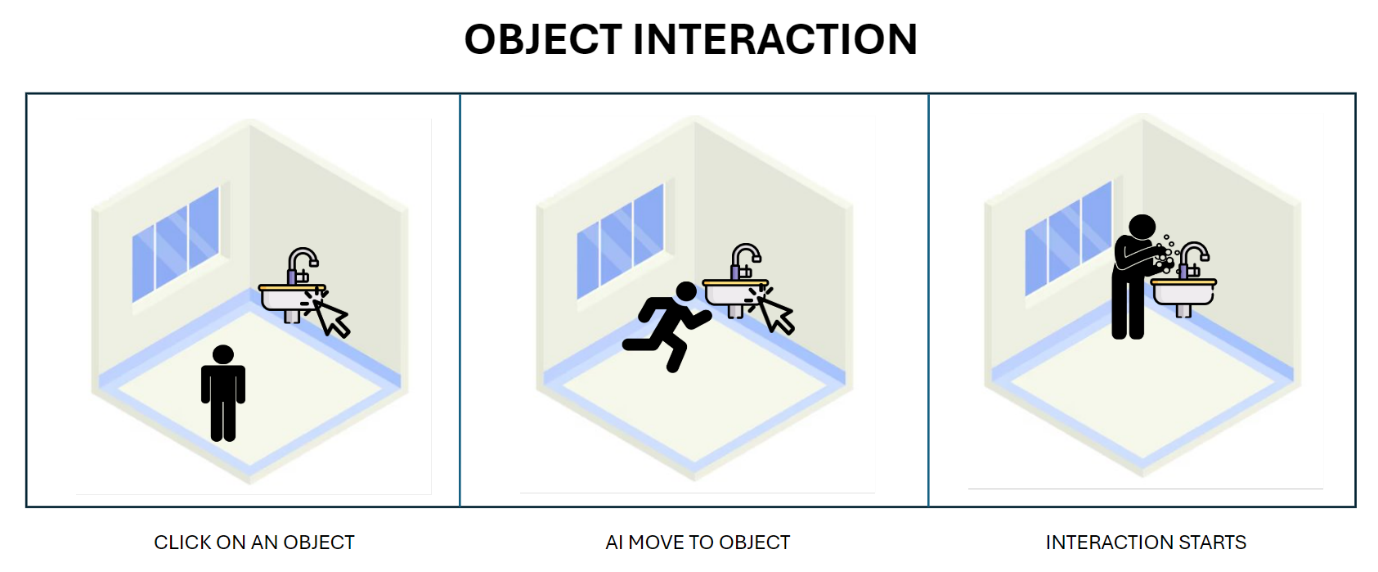
Here is a table showing all of the in-game skills and their data:

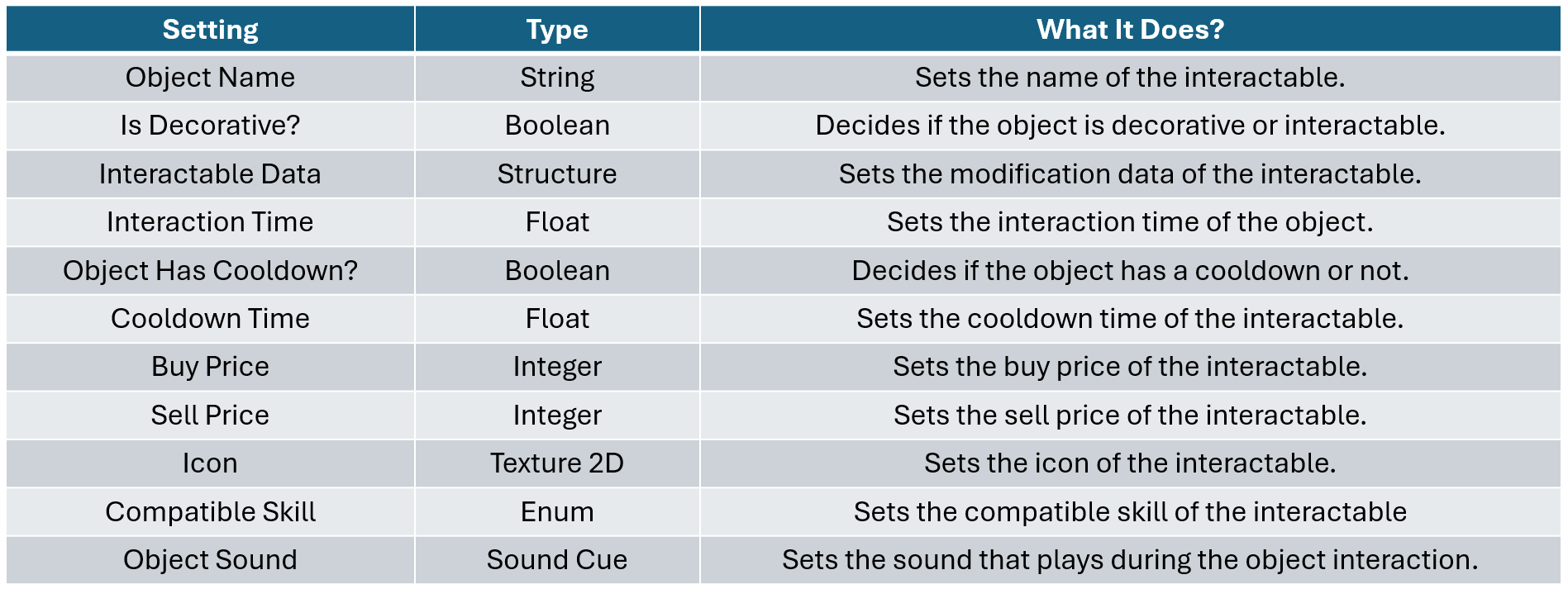
Here is a flowchart showing the process of modifying the characters skills:

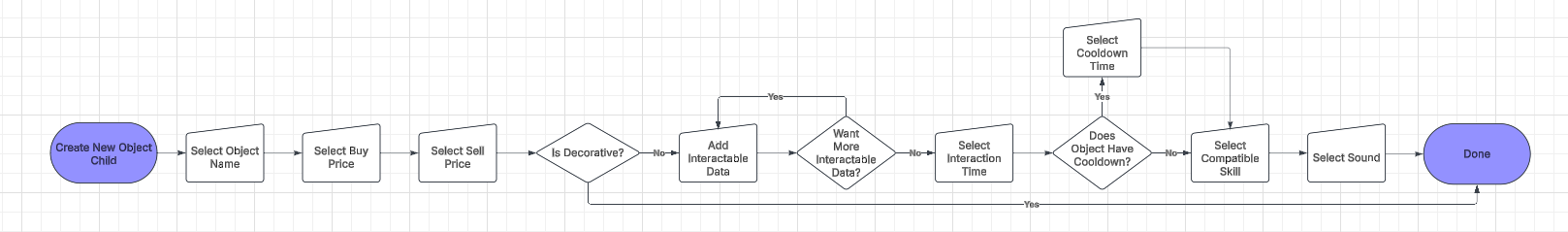


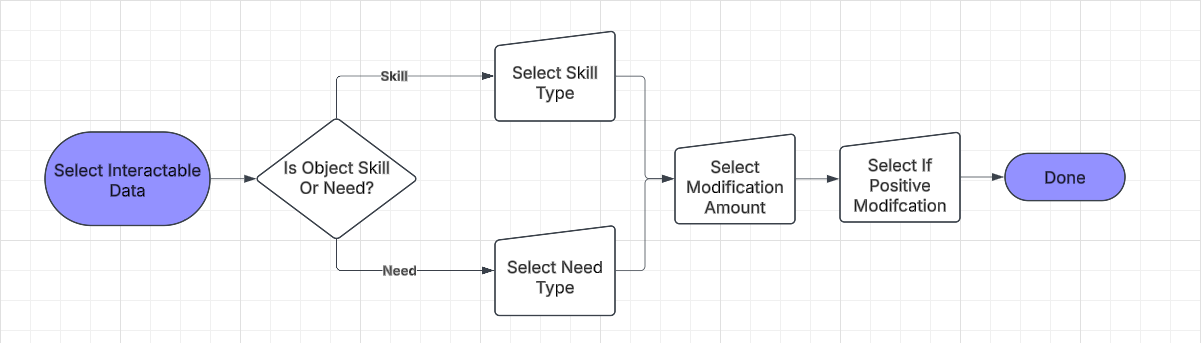
## Interactable Objects

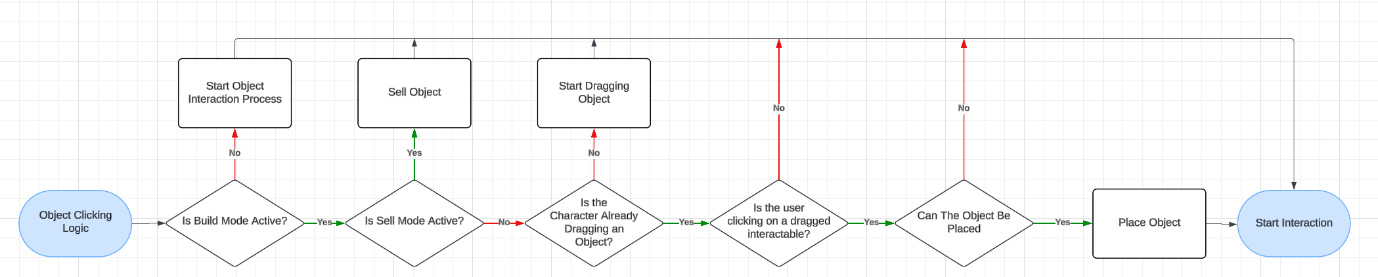
Here is a diagram made to show how the interactable objects will work:

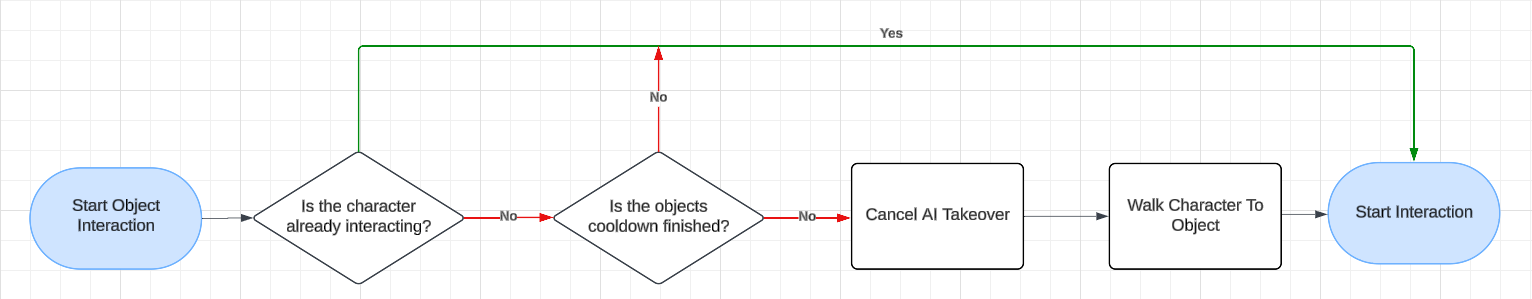
Here is a table depicting the settings that can be changed when creating an object:

Here is a flowchart showing the process of creating an interactable object during the creation of the game:

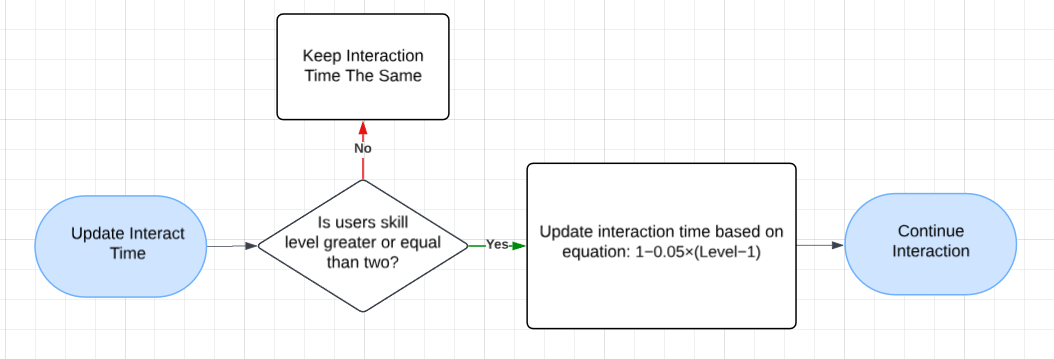
Here is a flowchart to show the process of selected the interactable data for an object during the creation of the game:

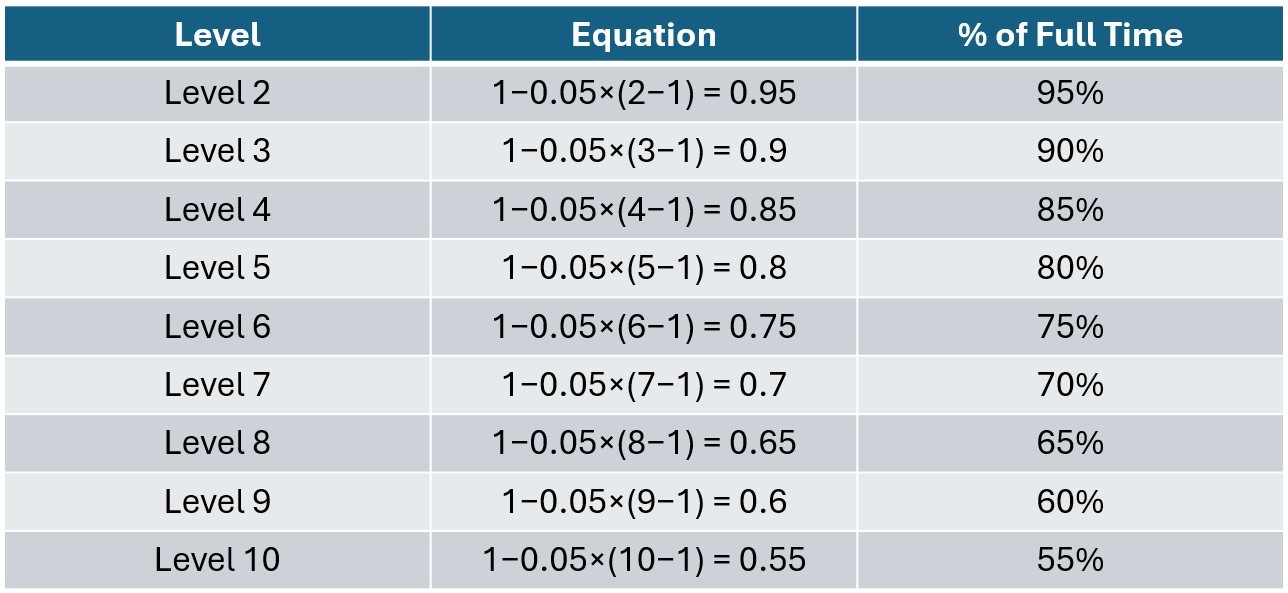
Here is a flowchart to show the logic of the process of when an interactable object is clicked:

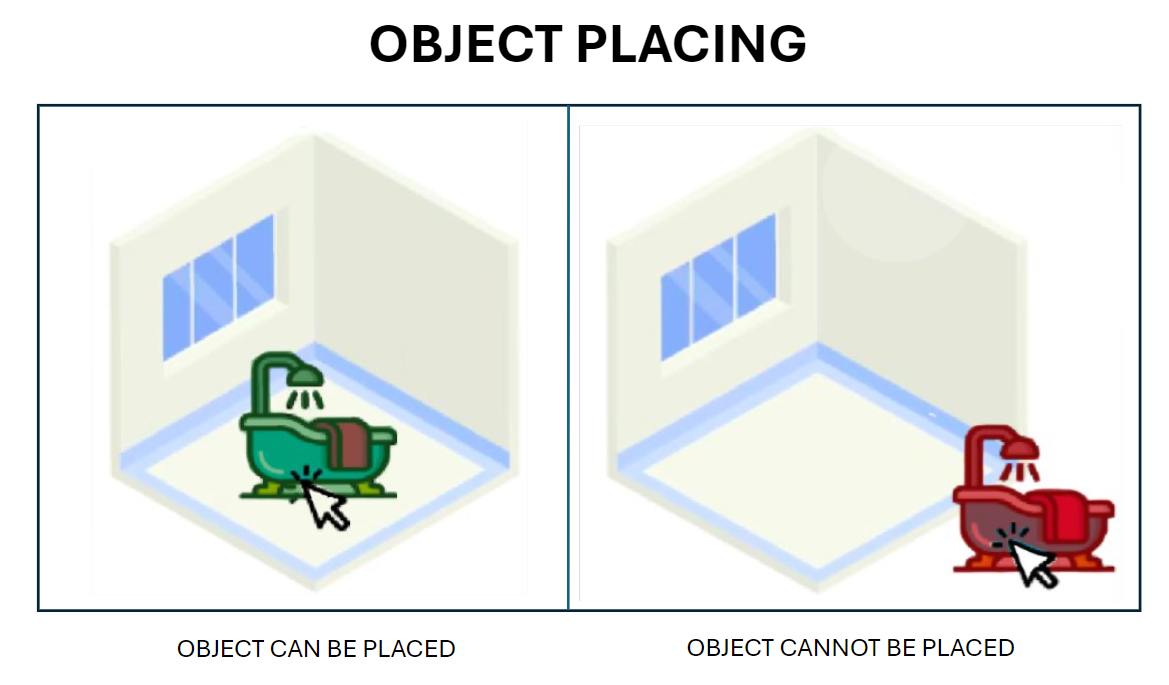
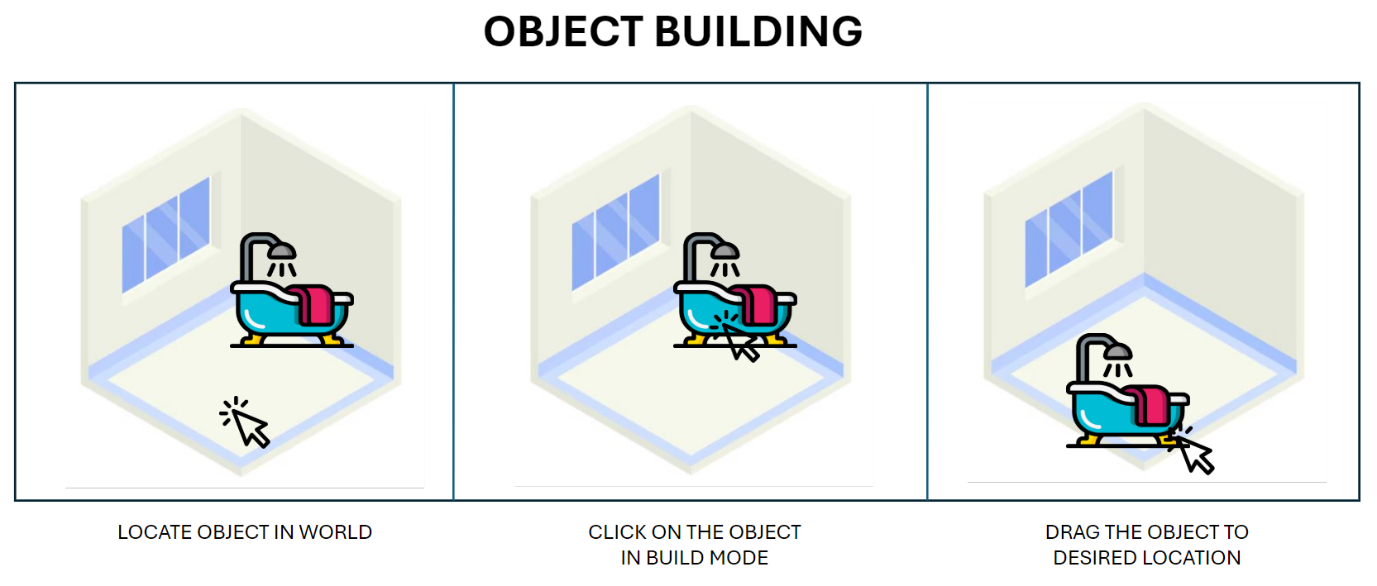
Here is a flowchart to show the logic of the starting the object interaction process:

Here is a flowchart to show the logic of the interaction process with an interactable object:

Here is a flowchart to show the process of checking a user's compatible skill level and updating the interaction time:

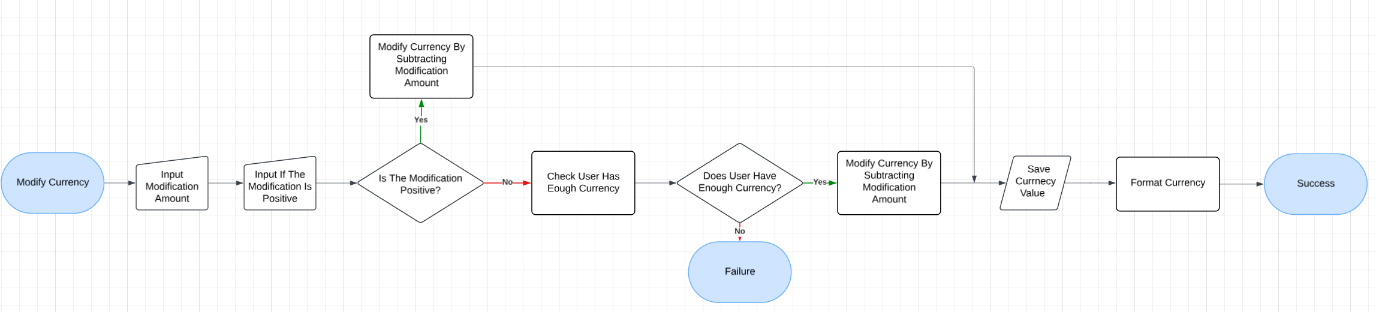
Here is a table showing all of the levels and their converted time:

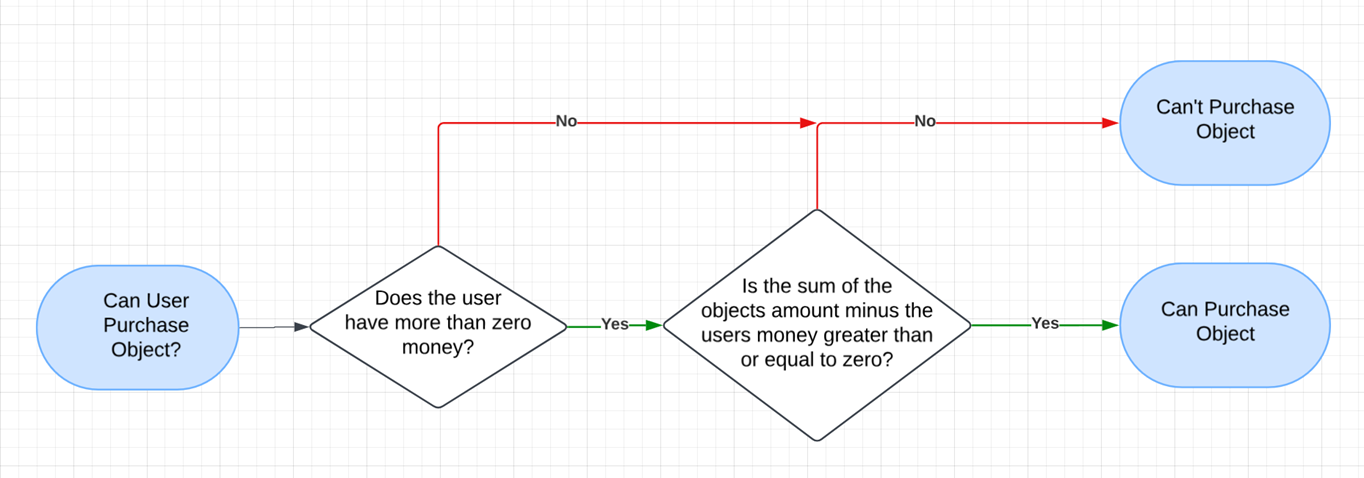
Here is a design showing how object placing works:



## Character Data

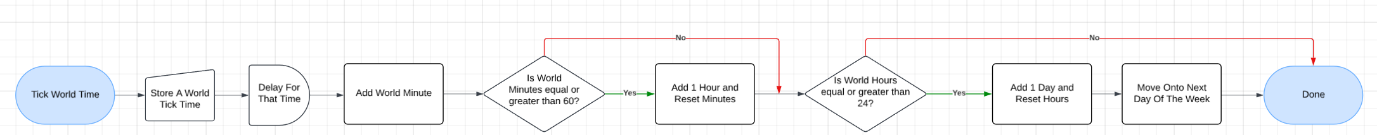
Here is a flowchart to show the process of modifying a user's currency:

Here is a flowchart to show the process of checking a user's currency:

Here is how the currency is format: ’**£xxx,xxx’**

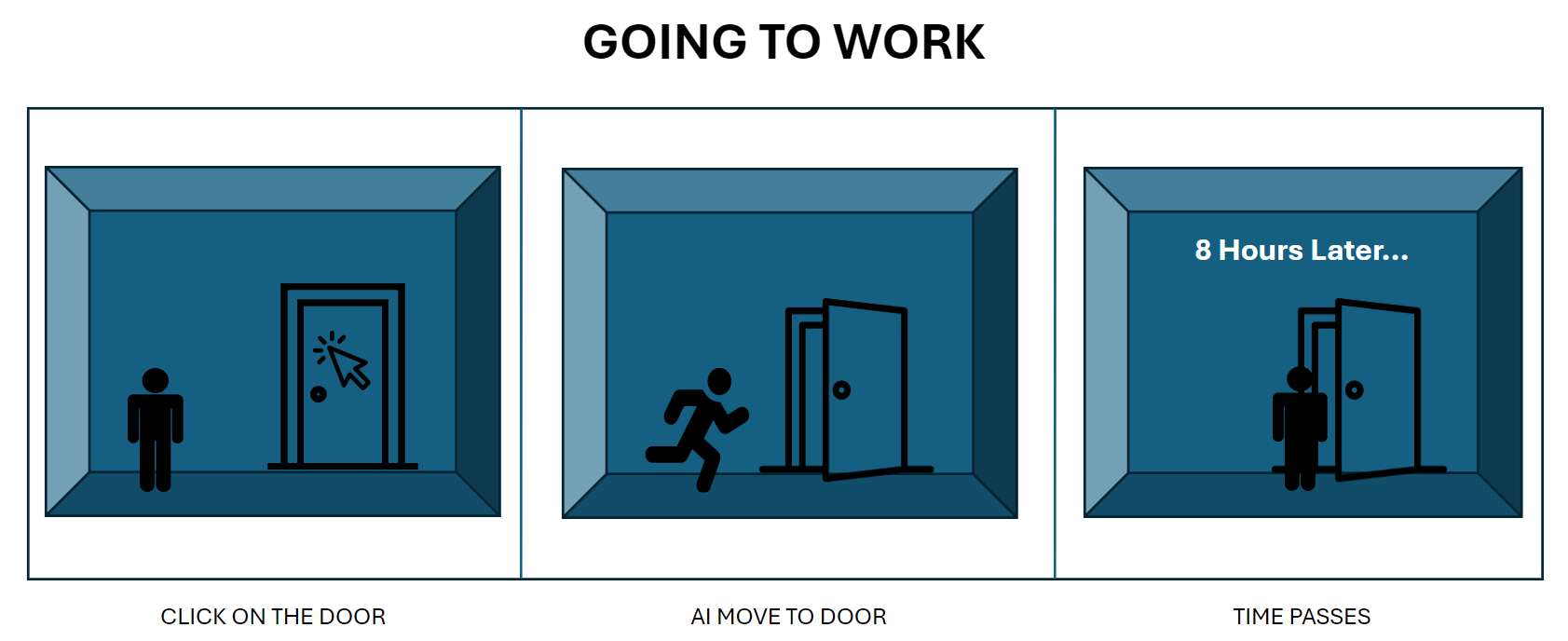
## World Time

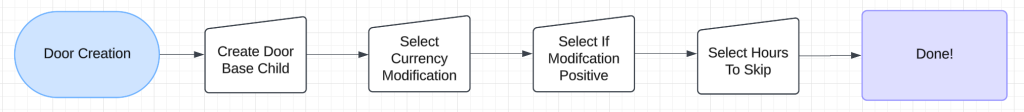
Here is a flowchart showing the process of the world time ticking:

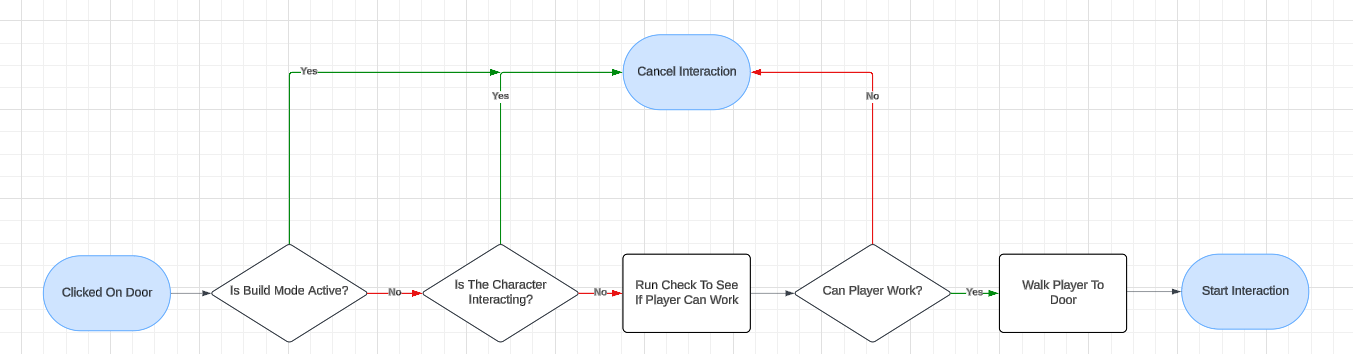


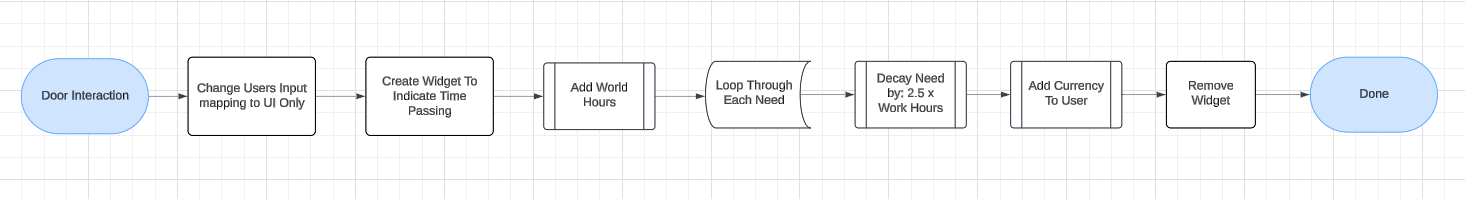
## Working / Doors

Here is a diagram I made to show the character going to work:

Here is a flowchart showing the process of creating a door child during the creation of the game:

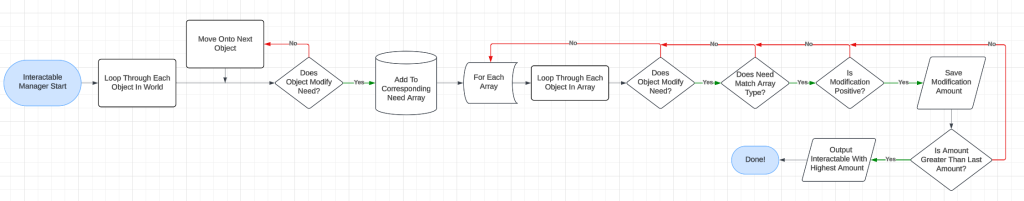
Here is a flowchart showing the process of clicking on a door:

Here is a flowchart showing the interaction logic for a door:



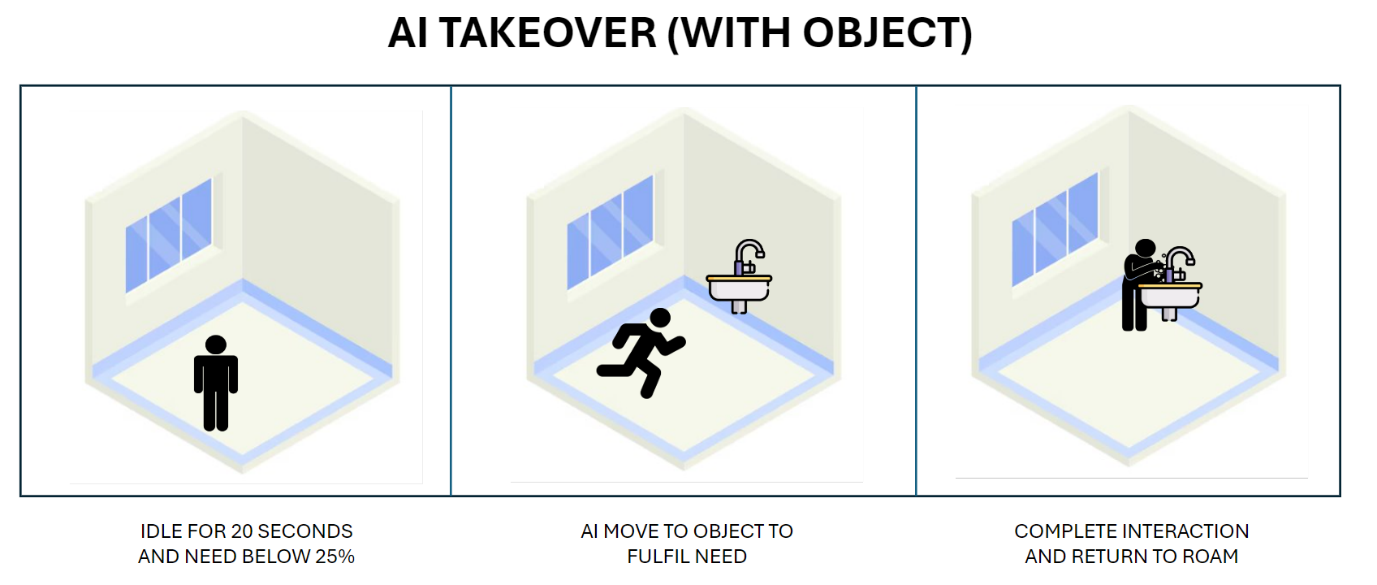
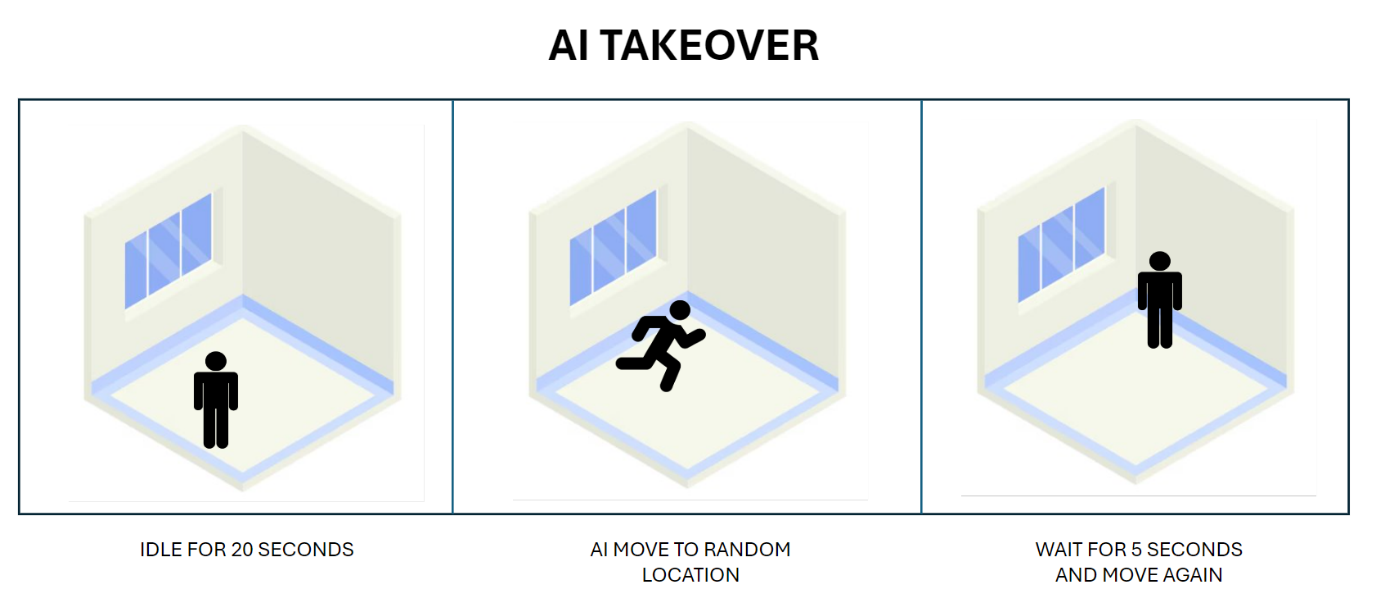
## Interactable Manager

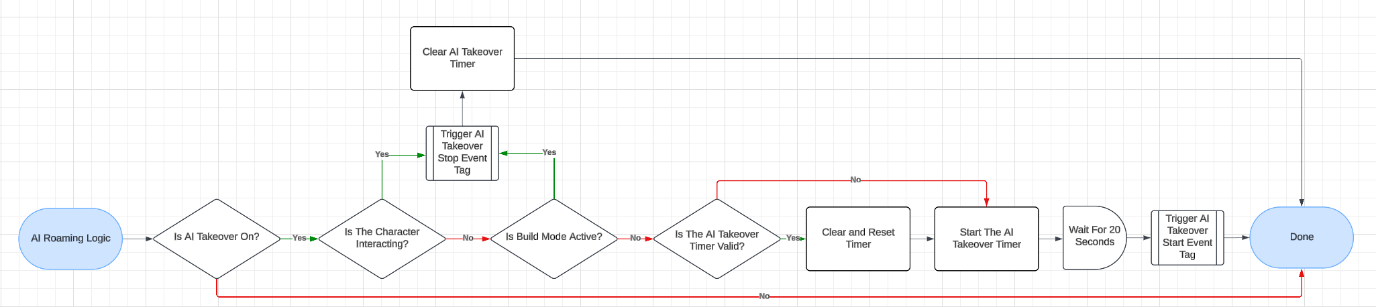
Here is a simplified flowchart showing the logic and flow of how the interactable manager sorts and outputs interactable objects that give the highest modification value per need:

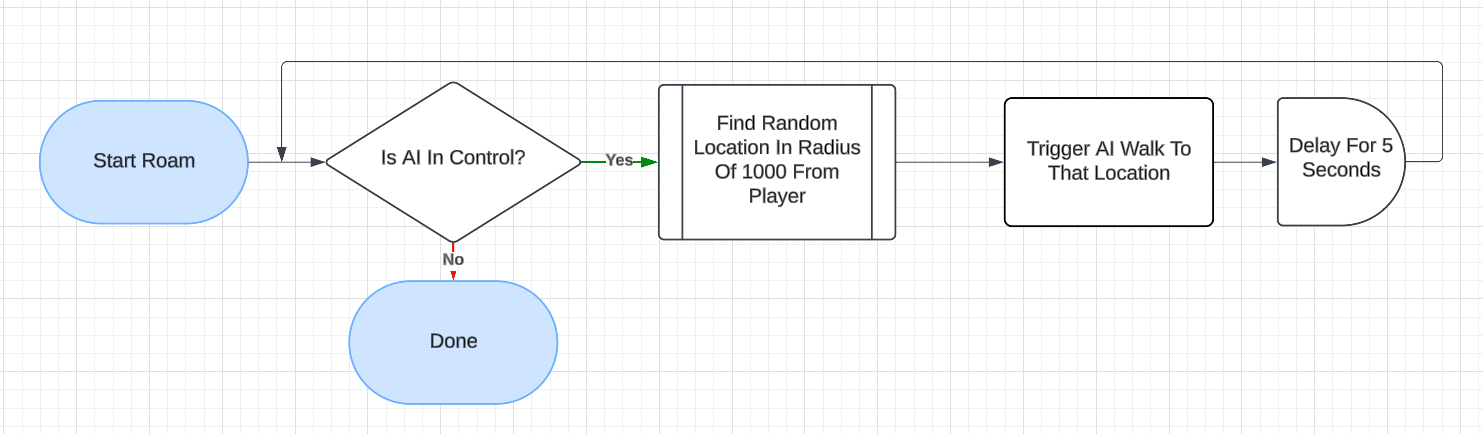


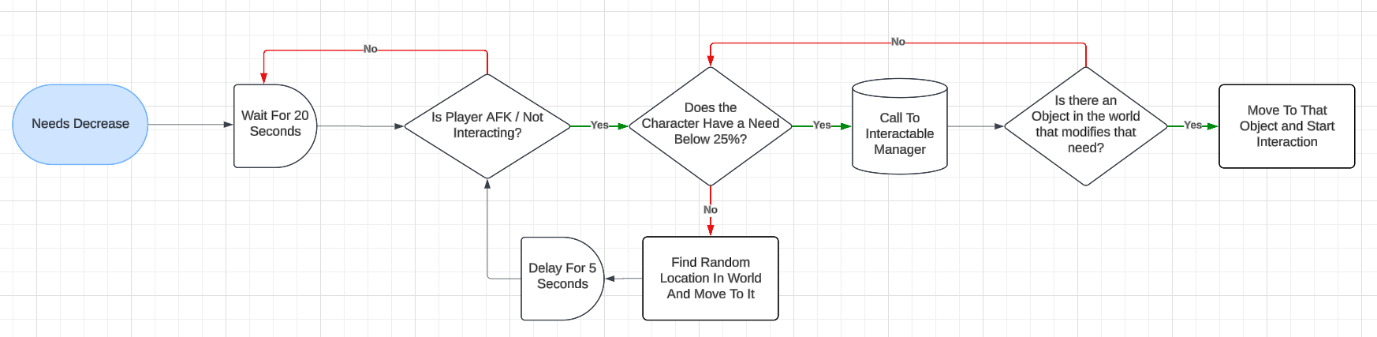
## AI Takeover

Here is a diagram I created to show how the AI Takeover will work:

Here is a flowchart showing the logic of the AI roaming event:

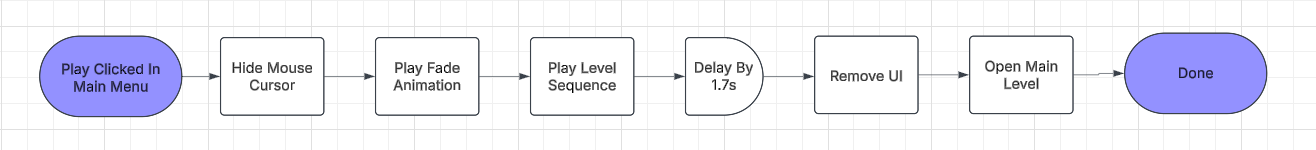
Here is a flowchart showing the logic of the state tree roam functionality:

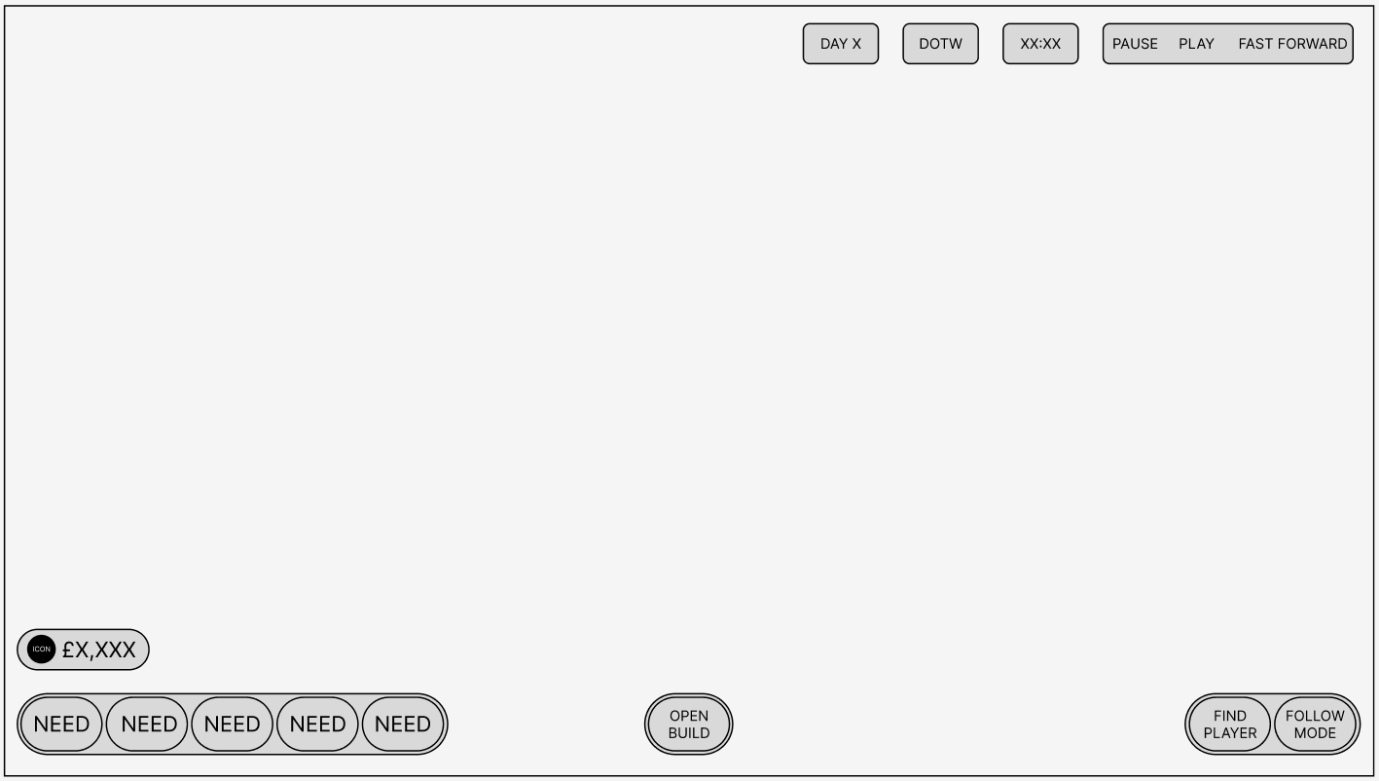
Here is a flowchart showing the logic of the AI Auto Refill Needs functionality:

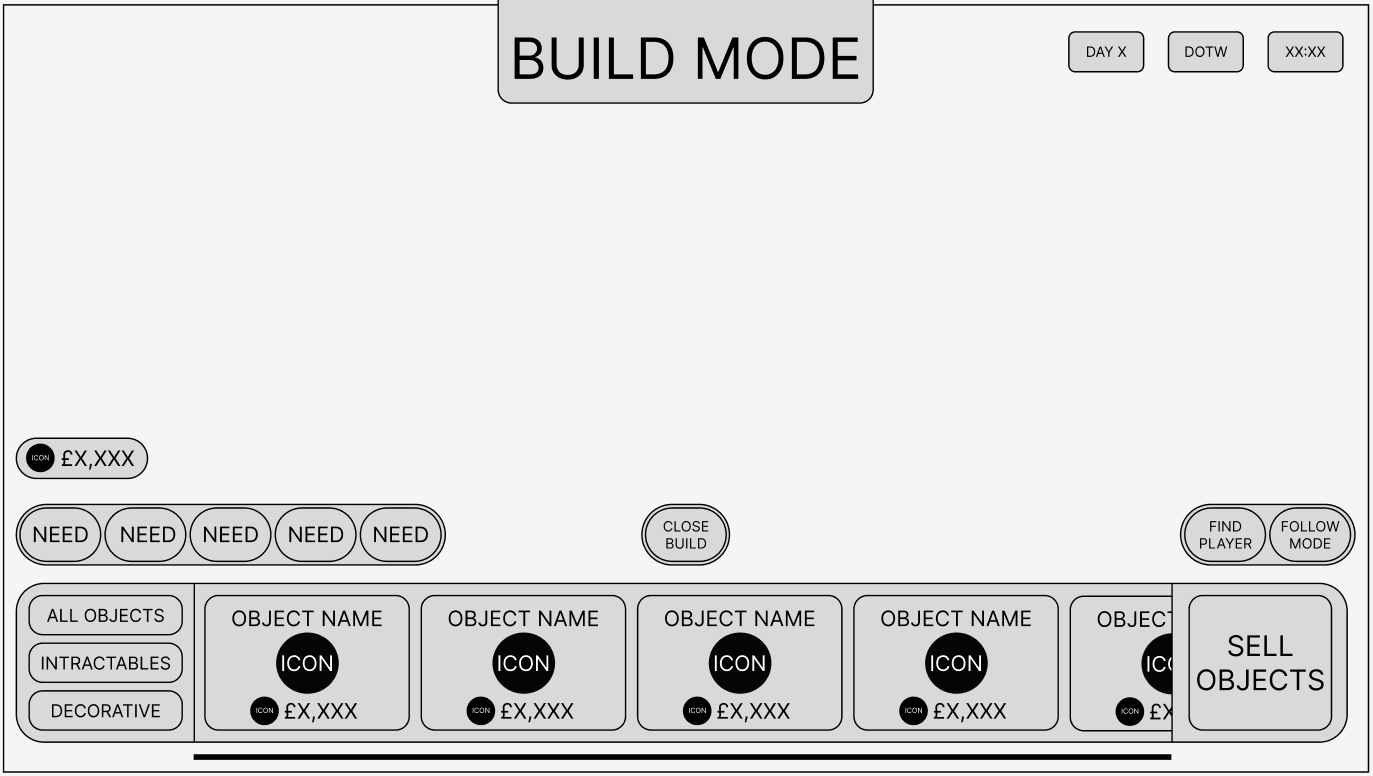


## UI

Here is a flowchart showing the logic of how the game is started and loaded from the main menu:

Here is a wireframe design showing how the main UI HUD will be laid out (no build mode):

Here is a wireframe design showing how the main UI HUD will be laid out (build mode):



# Coding Standards

## Programming Standards

For naming conventions, I will attempt to follow the unreal engine recommended asset naming conventions linked here:

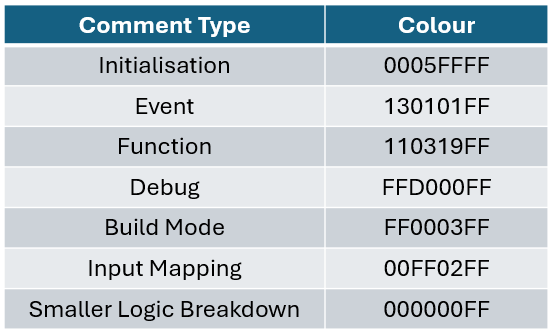
<https://dev.epicgames.com/documentation/en-us/unreal-engine/recommended-asset-naming-conventions-in-unreal-engine-projects>

For efficiency rules, I will attempt to make as much of the blueprint code as modular as I possibly can. This means creating large parent actors to create other world objects and actors, the use of enumeration and structures to ensure that data is correctly and efficiently laid out.

## Commenting Rules

For commenting rules, I will be commenting all larger and important blocks of code with a simple comment title that briefly explains the functionality of the block. For comment colours I also plan on using certain colours for certain types of code. For example, using yellow for debug code and blue for initialisation code. This is being done to ensure that the code is readable. If code is readable it allows for collaborative work with other developers, future project development after a long time a way (comments act as reminders), or even the ability to move the project on to someone else or another company.

Here is a table showing my comment types and their colours:



## Code Review Procedures

The code in my project will be reviewed very regularly. This is because I am going to be learning a lot about unreal during this development process, I plan on noting down areas of inefficient code during the creation process so that they may be returned to later. Furthermore, as my knowledge of the best coding practices increases as well as my skill in unreal, I will be consistently updating and improving code throughout the project.

# Production Overview

## Moscow

Must Have:

* A wide range of camera controls including movement, rotation (with keyboard and mouse), zoom controls, and player find/follow features.
* Game time controls such as stop, play and fast forward.
* Character needs that decay over time.
* Interactable objects that increase player's needs.
* Character currency that can be gained and spent.

Should Have:

* Basic building system that allows for interactable objects to be bought, placed, and sold.
* Character skills that can be gained and has an impact on a player's interaction with different world objects.
* AI system that takes control of the character and automatically solves their needs.

Could Have:

* A more refined building system with more items and snapping.
* Object UI to allow multiple interaction types per object.
* Detailed interaction animations for each object.

Won’t Have:

* Character Creator
* Other characters / NPCs
* House building system
* Other locations

## Timeline

[PTDGanttChart.xlsx](https://staffsuniversity-my.sharepoint.com/:x:/g/personal/s013671m_student_staffs_ac_uk/EWlLOui8W4BMoyMQ2FtMQHABEYCMfwwWBlCqmwm5_7deDw?e=cUCTEc)

