Contents

XB3992 – Pocket Planter - Brief

Name

Effie Sampson

Subject Area/Areas

Blueprints/Mechanics/Systems Design

Mobile game design

Brief Project Description

To create a mobile game based on the Cropout training project on Epic. I will create a small-scale planting game with a focus on peaceful cosy games. I will take elements from Cropout to further understand blueprint communication (using tags), managing data and saving performance to work on mobile devices.

Proposed Final Outcomes

A basic understanding of creating games for different devices.

Comprehensive understanding of manipulating timelines to trigger events that mimic the real-world passage of time.

Comprehensive understanding of clean blueprint communication and how these things affect performance.

Dependencies/Unknowns that may affect the Project.

I have never created a game for another device other than windows.

Research

Similar Game Research

Coming straight out of the Search for a star competition project I was completely lost of what my next project should be. I tried some idea generation techniques and was coming up with story ideas but no gameplay, and nothing that felt like a good show of blueprinting skills.

I decided to have a look at unreal training resources and see what areas I wanted to grow my knowledge in and came across a downloadable game project sample. The purpose of the project is to good practices for creating cross platform games, since I'd never looked at using unreal for anything other than the standard PC platform, I thought this could be an interesting area to explore.

Being a little farm game, it could also help with some of the areas I found challenging in my first project, such as inventory management.

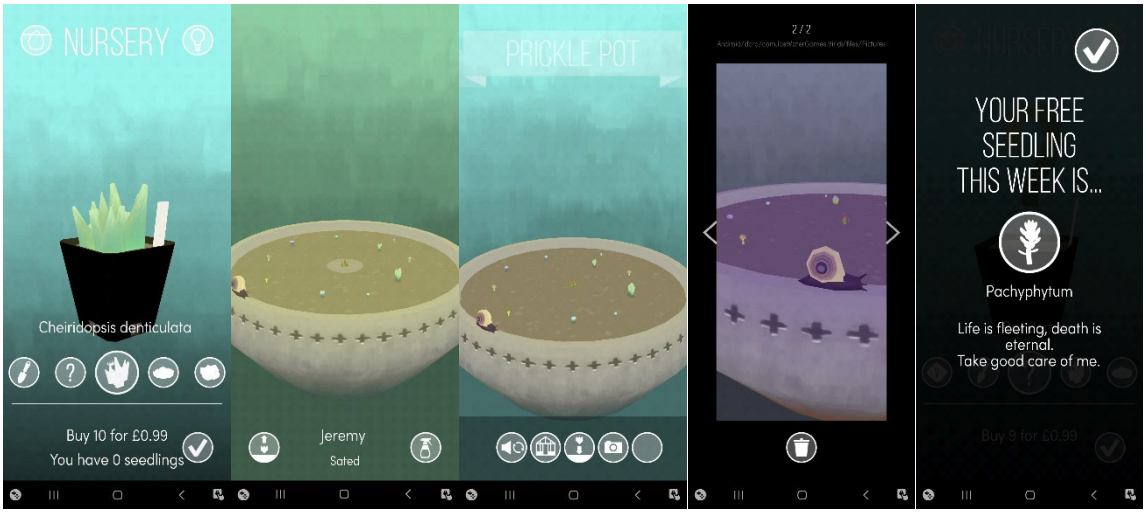
https://www.unrealengine.com/marketplace/en-US/product/cropout-sample-project?sessionInvalidated=true

After downloading the project, I explored it while watching the accompanying video. Then I spoke to my tutors for some ideas and advice. The suggestion was a small-scale plant growing game. I spent some time researching plant growing phone games. The two main ones being "My Little Terrarium" And "Viridi" both very different takes on similar ideas with Viridi being a much smaller and simpler game, though in my opinion, much more charming. Viridi is also much closer to what I had in mind to make for this project as it is more focused on the core idea.

Viridi key points:

- Pick a pot and starter plant set.
- Name your pot.
- 3D pot, no background, kind of similar to viewing a mesh on SketchFab.
- Touch each little seedling to splay with water and wait for them to grow.
- Touch weeds to cut them.
- Little snail friend who snails around the rim of the pot.
- Pick up and repot any plant, also pick up and move the snail.
- Can name all plants, and the snail.
- Can take in game snapshots of your plant pot.
- Does have a very simple and cheap shop area to buy new plant collections.
- No controls menu, Guidance is a very simple information drop down on some pages. Very simple intuitive gameplay so there isn't really much need for it.

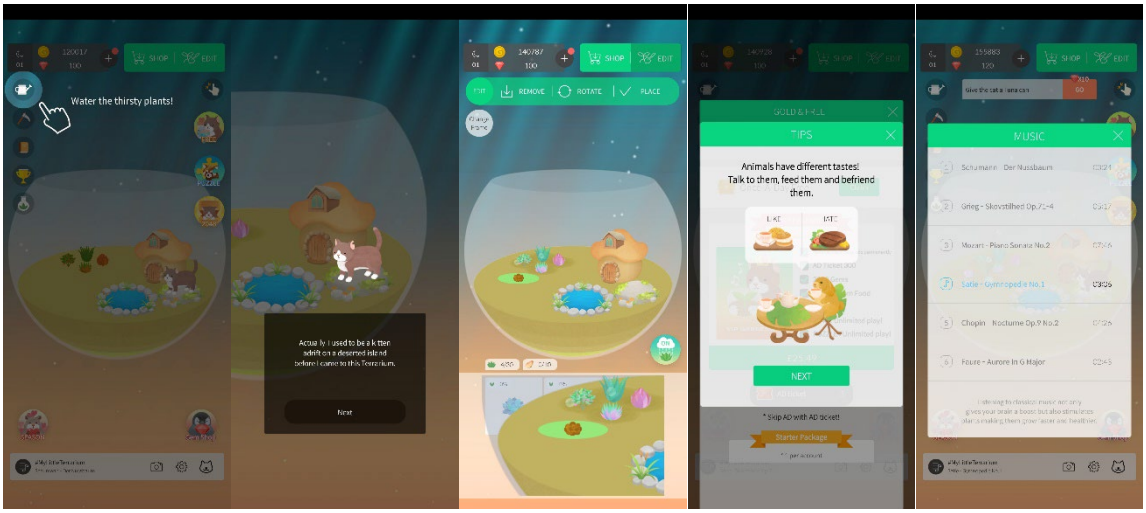
Final thoughts: Cute, simple, real time plant growing game with no complicated extras. Slow progress, but this is a selling point as the game is described as a meditative game.



My little terrarium key points:

- Select a garden from three choices that seem to control the types of plans that will grow.
- Upon starting the player has limited access as they are guided through the controls.
- There is a cat in the terrarium and the game play seems to be that the player looks after the cat, as well as other animals that visit the terrarium.
- The cat can talk… it talks to you. Not only that but he earns money by fishing too.
- Use the money to buy and plant new crops.
- When planting new crops there is a small pop-up screen that allows a better view as the players thumb blocks the screen.
- A little music player with classical music the player can change.
- Lots of mini games, jigsaws and puzzles that, in my opinion, complicate the game play loop.
- Achievements and incentives to grow the terrarium along with a lot of shop items that require real money.

Final thoughts: The core game play loop seems to be, grow crops, make food, feed the animals that visit the terrarium. The extra puzzles and achievements overcomplicate what could be a relaxing cute game.



Setting Up for Mobile Games

Setting up the project for a mobile device and setting up my phone to test on before packaging.

Step by step guide: Setting Up Unreal Engine Projects for Android Development | Unreal Engine 5.0 Documentation

More info pages:

Setting Up an Unreal Engine Project for Mobile Platforms | Unreal Engine 5.0 Documentation

Setting Up Your Android Device for Developing Applications in Unreal Engine | Unreal Engine 5.0 Documentation

How to Set Up Android SDK and NDK For Your Unreal Engine Development Environment | Unreal Engine 5.0 Documentation

Unfortunately, after spending a whole day trying various methods and downloading various software, I've been unable to get the project working on my mobile for testing. The project is set up within unreal to be produced for a mobile, so I am going to continue to develop it with this in mind while I extend my research on the software needed to package for Android.

Cropout Research

https://www.youtube.com/watch?v=EjPp5A43HQI&t=5s

This is a link to the guided tour of Cropout that I had watched this before I had any ideas for my own project, so I thought it made sense to watch it again from the perspective of what I can take into my own project.

Async load is mentioned a lot, I think this sounds like something that would be unnecessary for a game as small as my own, but it's an interesting area for further research on future projects.

Inheritance. Though not a new concept this is done very cleanly, and I'd like to aim for my own to be this clean.

Resource types are handled with tags, this is something I've never implemented before and would like to explore, but I'm not sure it would be needed in this project. I will however keep it in mind in case it does seem useful. An auto Enum to string converts resource type into an array of tags that give all child classes that tag. Tags are then copied from the clicked resource to the villager selected by the player, without any need to cast. This also applies to the growth state of crops.

The player, hover states decide the mode from: Placement mode, Build mode and villager mode.

There are two mapping contexts within the player that are activated dependant on the hover mode selected. Position check on click indicates what mode this should be. (For mobile this also does a check for more than one touch/tap detected.)

Pop up layers in the HUD, (Yes to confirm etc) Done via a push widget. This is something I want to implement for removing plants and starting new games.

Resources and villagers are stored in the game mode. Event dispatchers send info to GI.

Notable blueprints I want to have a closer look at: BP_Resorce (Tag creation) and BP_interactable (To check for overlaps)

Development

Tracking The Click

I have used the top-down template to get started, but having a look at the default player character, there are many settings that would be redundant, including the mesh itself. Because of this I have set up a new player character that is simply a pawn rather than a character, and using Cropout as the example, added a floating movement to get access to further settings. However, I have not added a camera, as I want my view to be static, pointing at the flowerpot or window box.

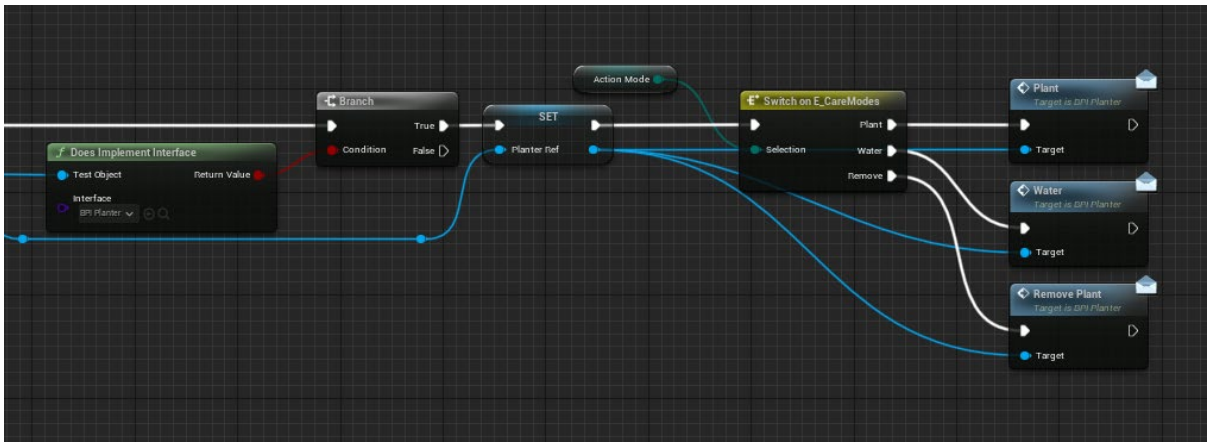
I changed the default “move to” function into a “teleport to” function within the player character movement. This way when the player teleports to the planting location a spere trigger on the player can handle the desired action in the plant pot.

I added the static camera and linked the players view to this camera in the level blueprint. I then made a new level and blocked out the windowsill.

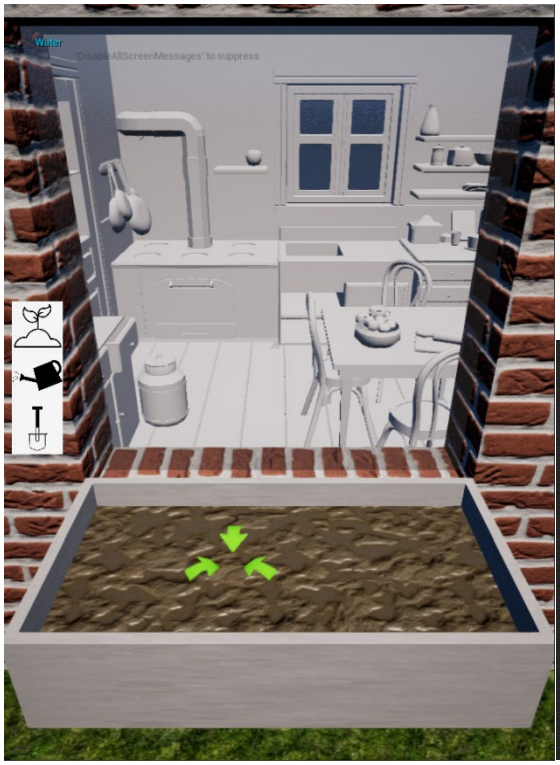
In my first attempt I made planter blueprint, I was hoping that I could use a tag or material that tells the player which component of the planter they are in contact with. I wanted this to be a signal that the player was clicking on mud, and therefore could plant a seed. But I struggled to interact with a specific component of the mesh, the tag would trigger on any part of the mesh, and I didn’t want the player to be able to plant a seed on the wooden box of the planter.

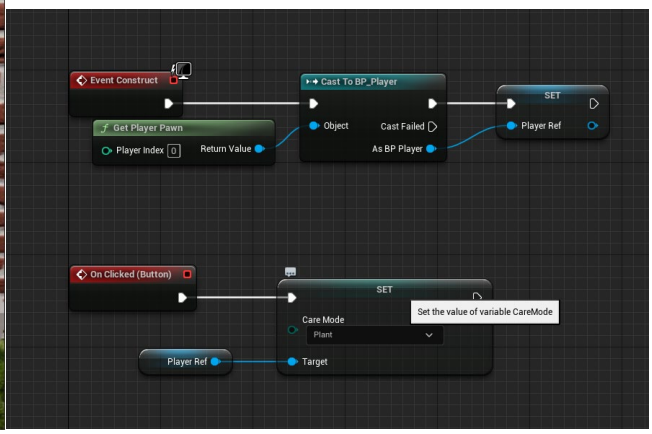
Instead of this I used a trigger box with an interface, this way I didn’t need to check what I was clicking, I only needed to check if the player was overlapping an actor implementing that interface. Within the interface I set some care functions (Plant, Remove Plant and Water) to use or alter later.

I wanted to try switching modes with a HUD overlay, I set up an Enum to switch between different care modes and for now just plugged this into a print string to make sure it was working. I intend to set the mode by clicking the menu button, though I’m not 100% sure if this is as easy as I’m assuming it will be. It feels like a trap!

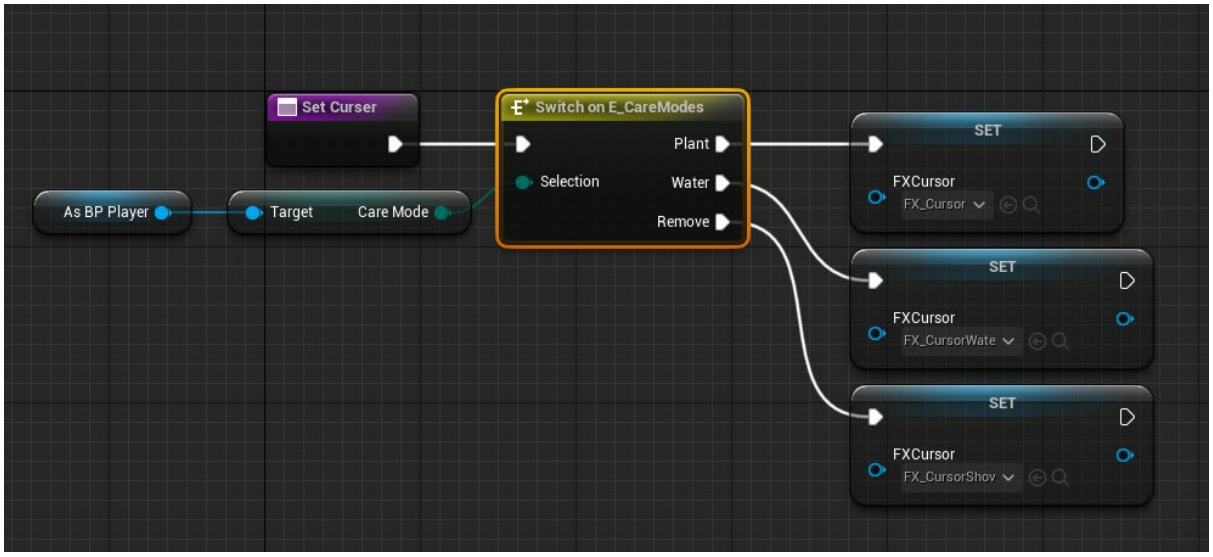


I set up a basic UI to test with and this does work. But it requires a lot of casting, so I am trying to find a cleaner way of doing it with an event dispatcher.

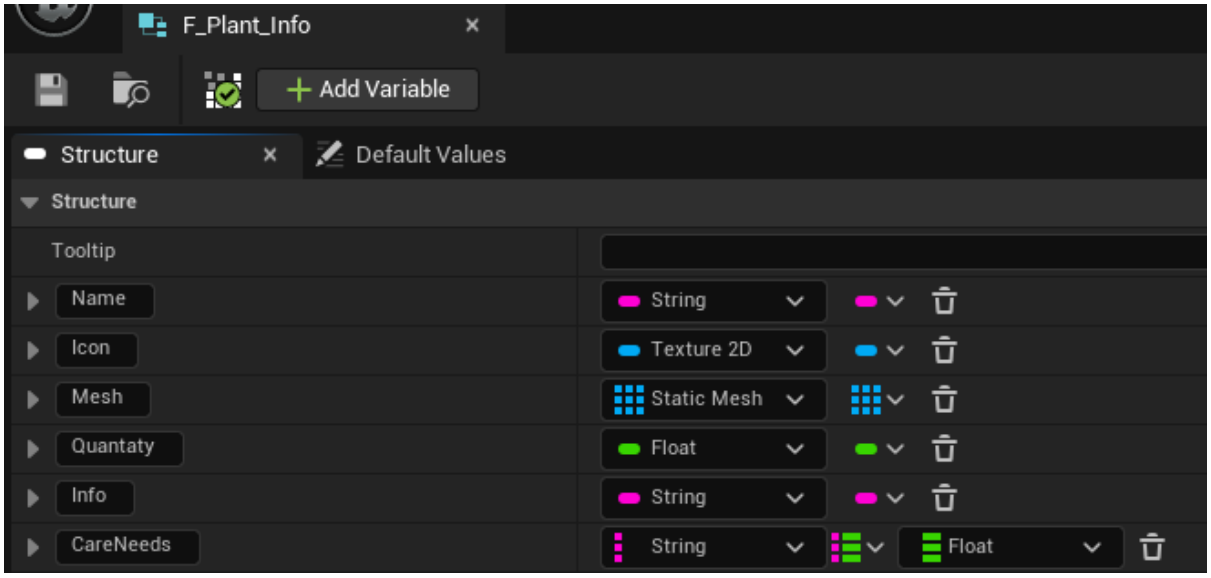




I also wondered what else I could change with this same setting and made some variation on the cursor FX by duplicating the default arrows and adding different images. Admittedly this was a bit of a detour from programming at this early stage, but I did enjoy it.



After this I started working on a base blueprint for a plant. I created a Structure where I wanted to hold the plant type info. I set this up with the static mesh as an array so that I could add three meshes to show the stages of plant growth to each child of the base plant blueprint.

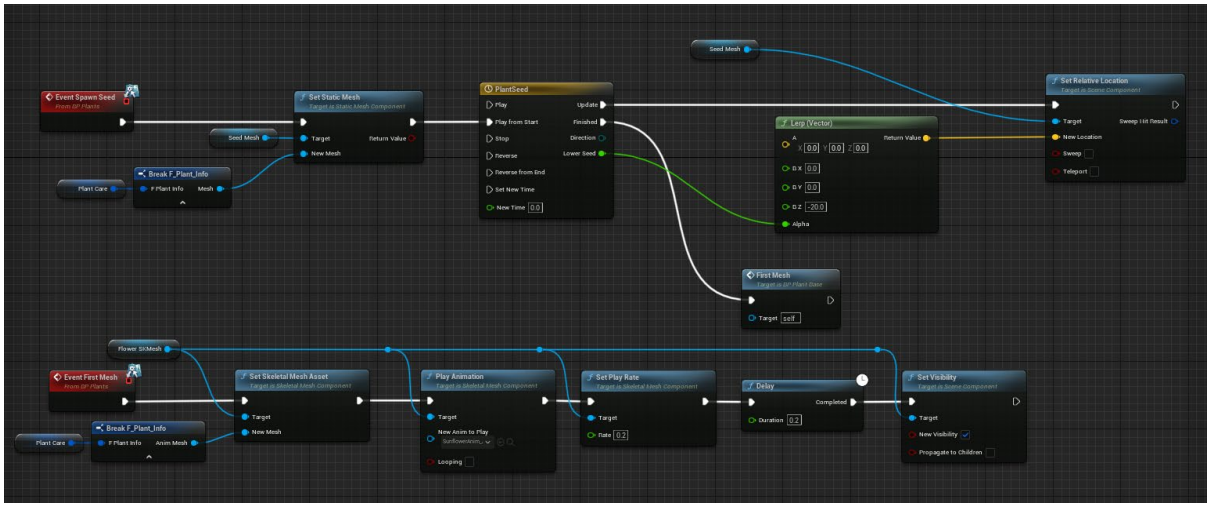


I asked a friend, Oliver, if he could make me three simple low poly meshes to test this out on and he got inspired to create an animation of the plant growing too. I really like the animation, so I changed the mesh array to a skeletal mesh input and instead connected the animation to a variable playback speed. My intention is to change the play back speed depending on how healthy the plant is.

I also added a variable seed mesh and set the seed to spawn before moving down on a timeline before the animation starts to give the effect of the seed being planted and a flower sprouting. I also thought this would work well in the future to add different seeds and grow different plants. For now, I will test this on some different coloured versions of the sunflower.

I set the plantable area blueprint (mud) to handle the spawning of the sunflower. Since this was already colliding with the player pawn, it was really easy to transfer the clicked location from the mouse, using this as the spawn location and avoid casting to the player controller.

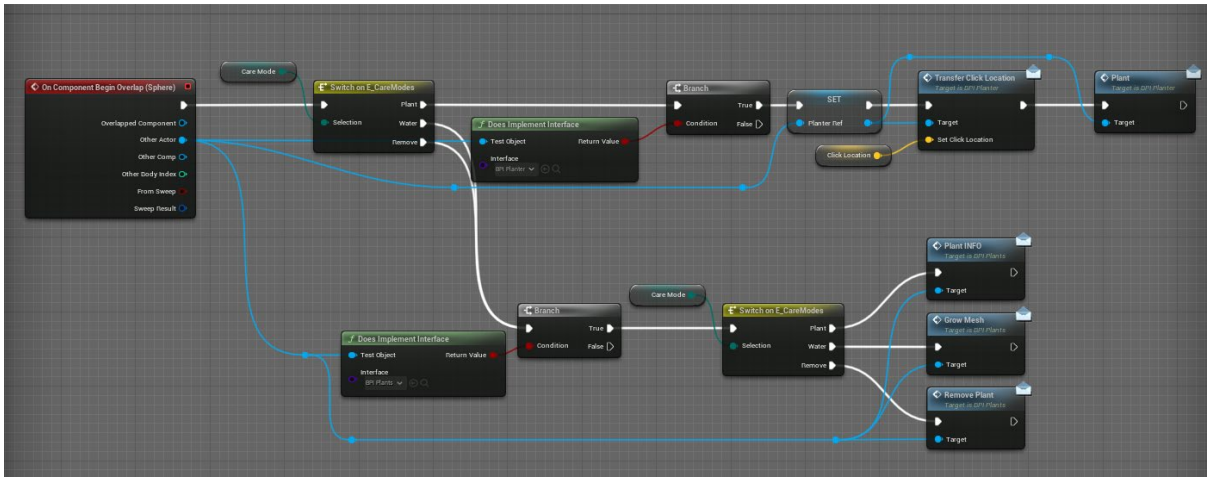
After spawning I noticed the animation quickly flashed up the full flower before playing so I set the flower to hidden before the animation starts and then after a short delay, made it visible again. I'm sure there are probably much cleaner ways of doing this, but my knowledge of working with animations is still very limited, so I left it like this for the time being to focus on the rest of the plant care functionality.

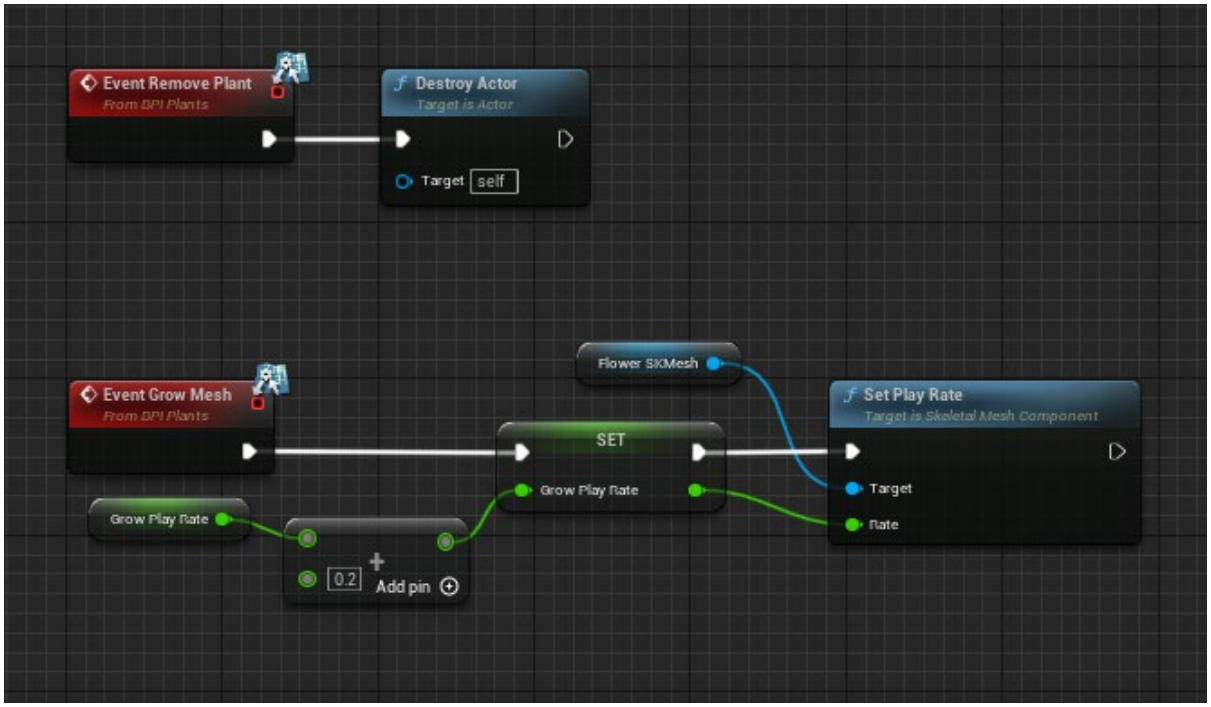


At this point I thought it might be a good idea to add a game instance to hold the current plant info that I might need later. I added a make array function to add this as the plant is spawned. I'll come back to this to add more functionality later.

The next thing I wanted to do was to check the plant collision to make sure the plants don't overlap. I am thinking of doing this in two ways. One: Currently the player has a collision that is checking for a plantable area, this functionality could also check to see if it has hit another plant, and if so, stop the planting function from running. Two: I could use the already stored click location to check for a hit actor. Before deciding, which was the best option, I wanted to tie this in with the remove function, because that could help me decide.

Within the player pawn I moved the switch on care mode to before the check for an interface and set some quick remove and speed up functions in the base plant blueprint to check if the switches were working. This did work to remove the plant, but it didn't stop the collision with the player on the planter so a plant would still spawn when in planting mode.

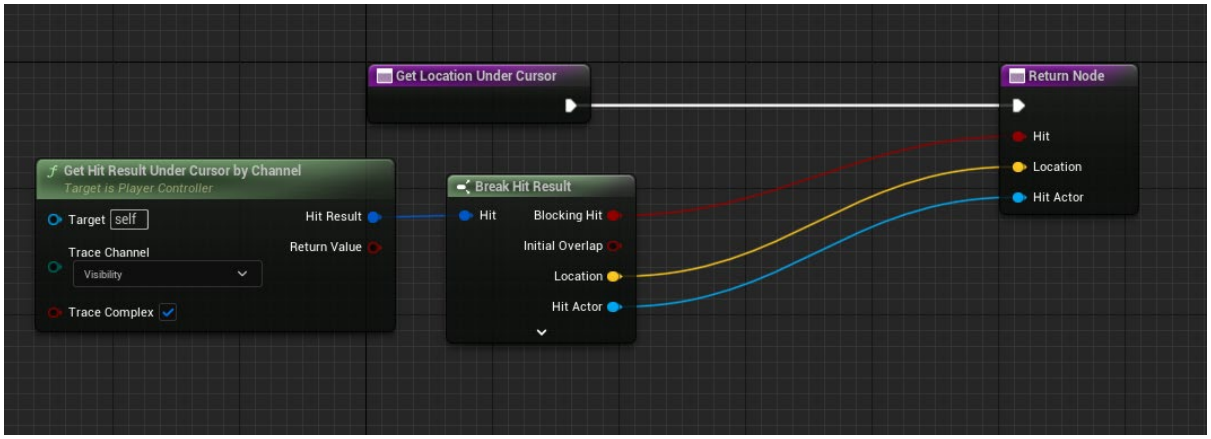


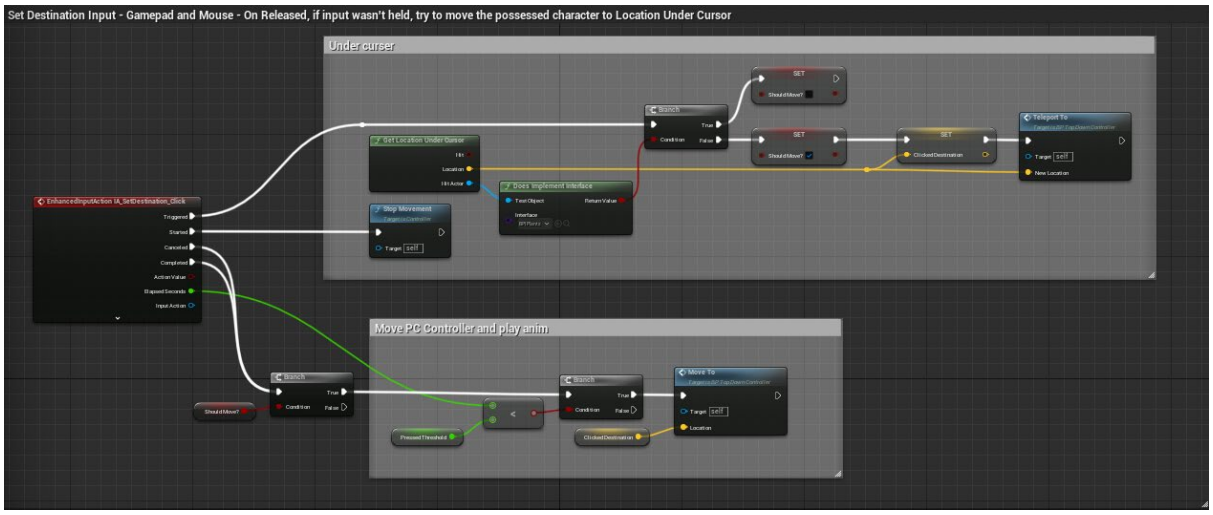


I tried to set up a sweep for plants with a Boolean just before checking for the plantable area within the player character, but the return was a true *and* a false, since the second the plant spawns it hits

the players collision, meaning every click triggers a true and a false. Considering that the thing handling this collision is the player character's collision sphere as the pawn teleports to the click location, it might make sense to use a function separate to this collision that first checks the area.

After thinking about the logic I decided that rather than making a complicated function, I would use the existing hit result in the player controller and stop the pawn from moving at all if the mouse click hits a flower. With the flower being an animation without proper collision, I also added an invisible cube to the base of the flower to trigger the hit result. This worked well because it meant the player could still click on soil behind the flower, rather than being blocked by the stem.

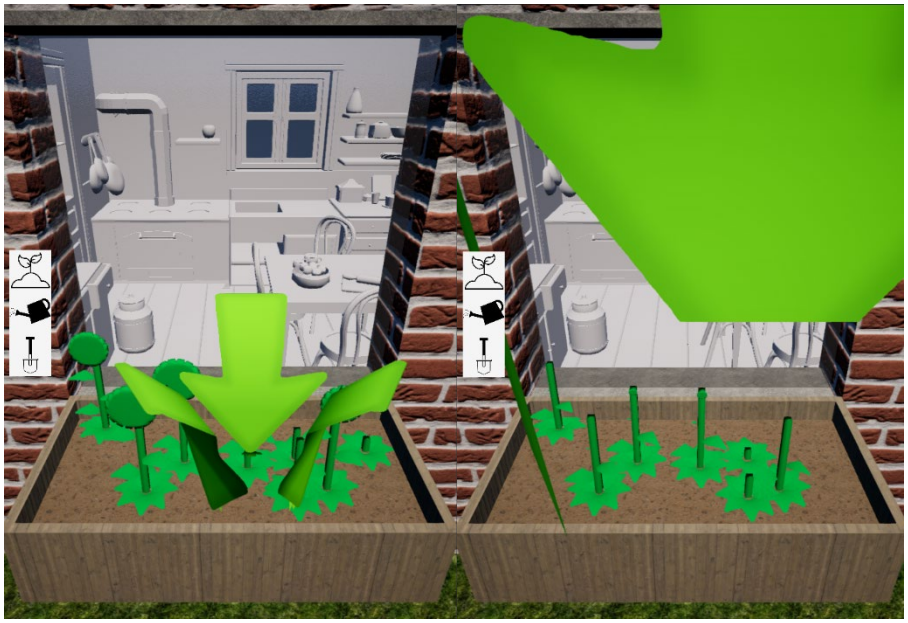




Although the click now blocks the spawning of new flowers, this did create a new issue where the teleporting now doesn't trigger a new collision, meaning the player would have to click off the planter and then back on to generate another hit response. I'm unsure what is causing this, and why it wasn't an issue before as I hadn't changed the way in which the pawn teleports, I had only blocked the ability to.

I also realised on more thinking that when it comes to watering the plants, I don't want to be blocking the movement of the player! So, this method won't work anyway, silly mistake! Instead, I changed the hit result to set a Boolean within the player pawn True if hit a plant, false if not, then I can handle the way each hit works from the player.

The triggering issue seems to be to do with the player pawn, I had changed the collision on the pawn mech (an invisible blue cube.) because the collision was causing the pawn to spawn on top of itself, therefore getting closer and closer to the static camera. Every project I work on, I come across issues with collision!



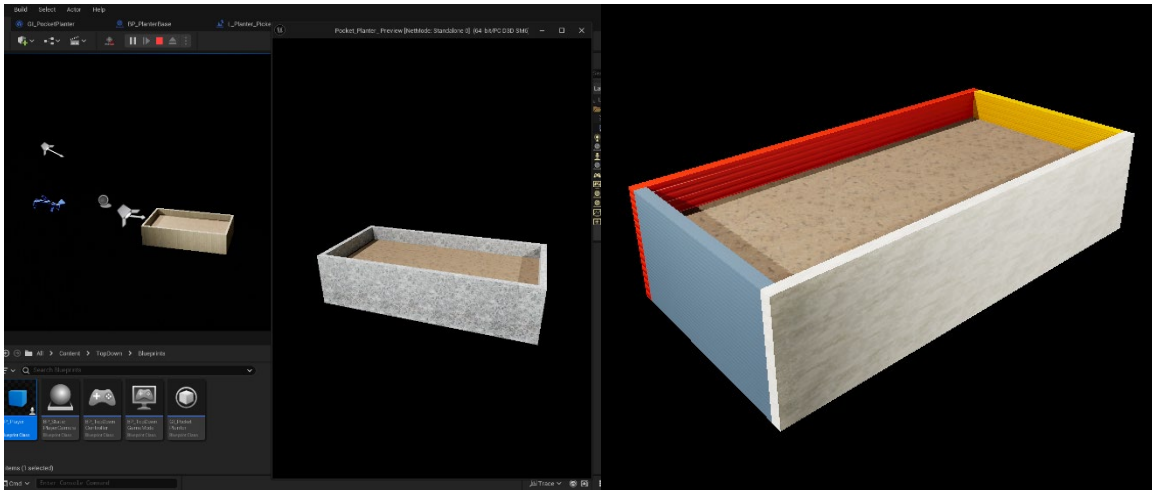
Despite this little glitch the functionality of the destroy and water button seem to be working as intended, and the plant function now does only plant in an empty space. I'll come back to the collision issues.

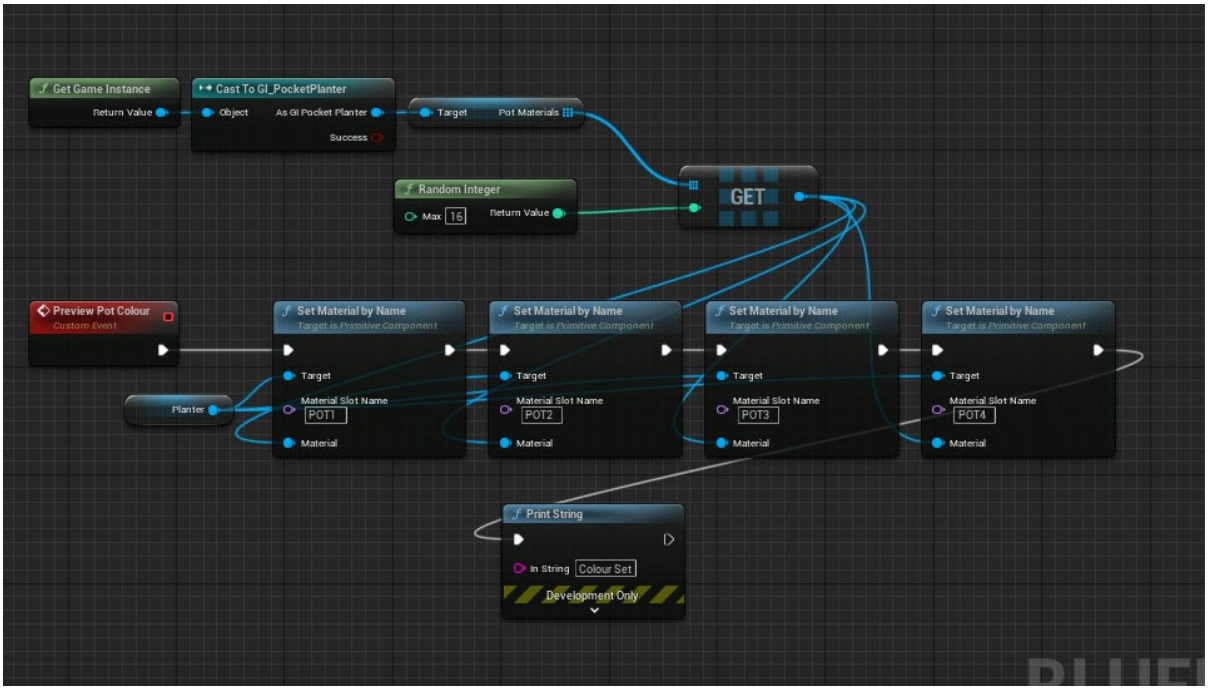
Woah! Big file!

At this point I made a new back up file and realised that the file size was huge due to the texture pack I'd imported. Since I only wanted to use a few of these textures I went through and reduced the amount saved.

Then I started on the startup menu where the player can choose a starting pot. I wanted to have different styles of pots, but currently I only have one mech, so I'm thinking that instead the player could choose the material of the pot to showcase this function, then perhaps I could add pots later.

To test out some functionalities set an array within the game instance (I put it here because I want the choice to be saved later) with pre-determined material options and hooked this up to a custom event that just selected a random material when I press the H key. I made a separate empty level with just plant pot blueprint and some spotlights. I also added a rotation movement component to showcase the selected colour to the player. Although the random integer did select a random colour for each side of the planter since this isn't one material in the static mesh, but the selection was working, so I moved on to creating the UI to control this selection.

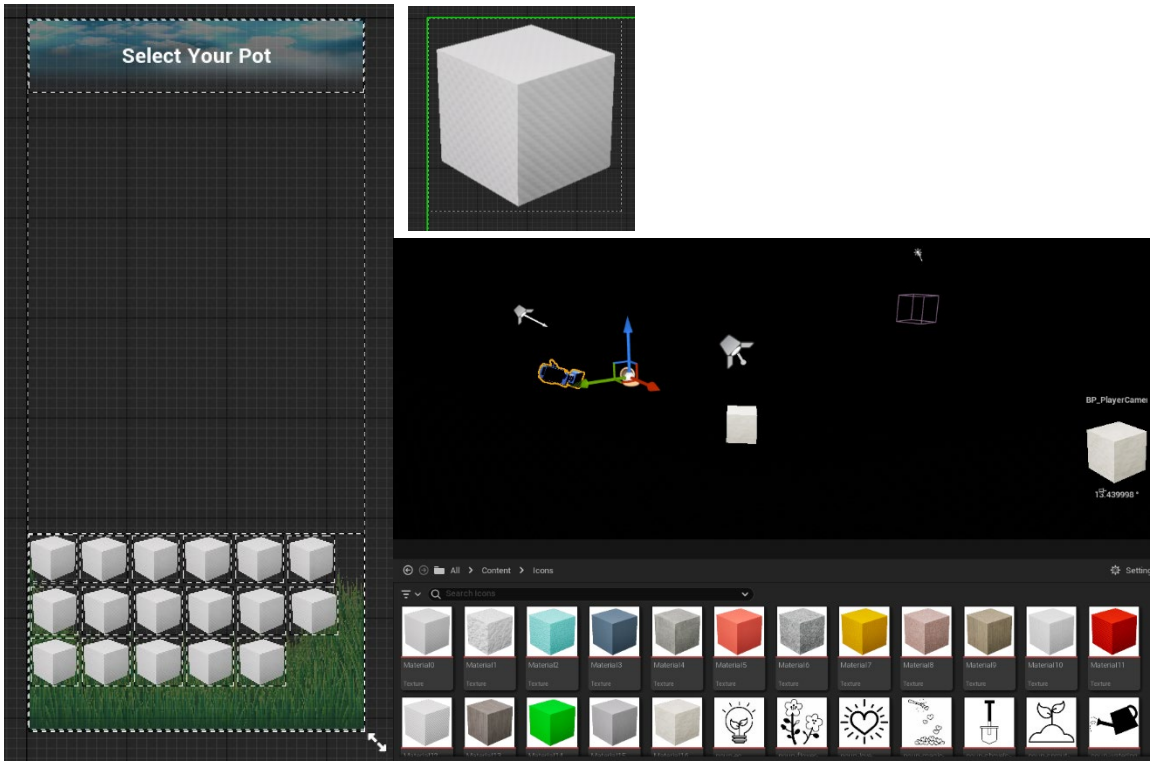


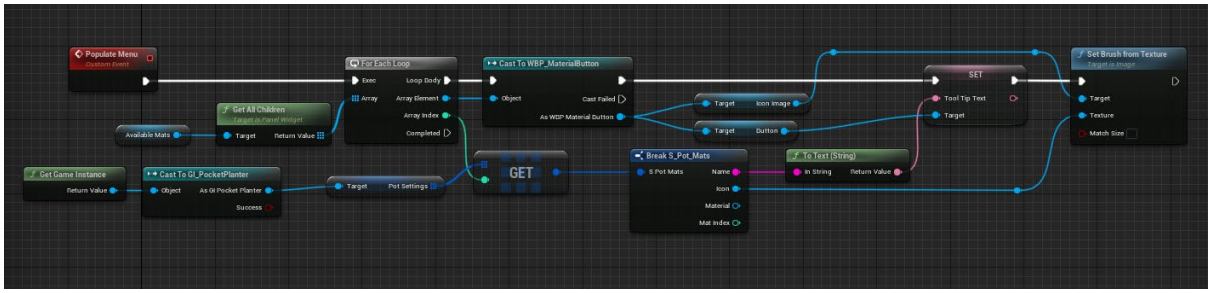


The selection UI

I first created a menu layout and then the button, to get some icons for the materials I used a cube and empty level to get some screenshots.

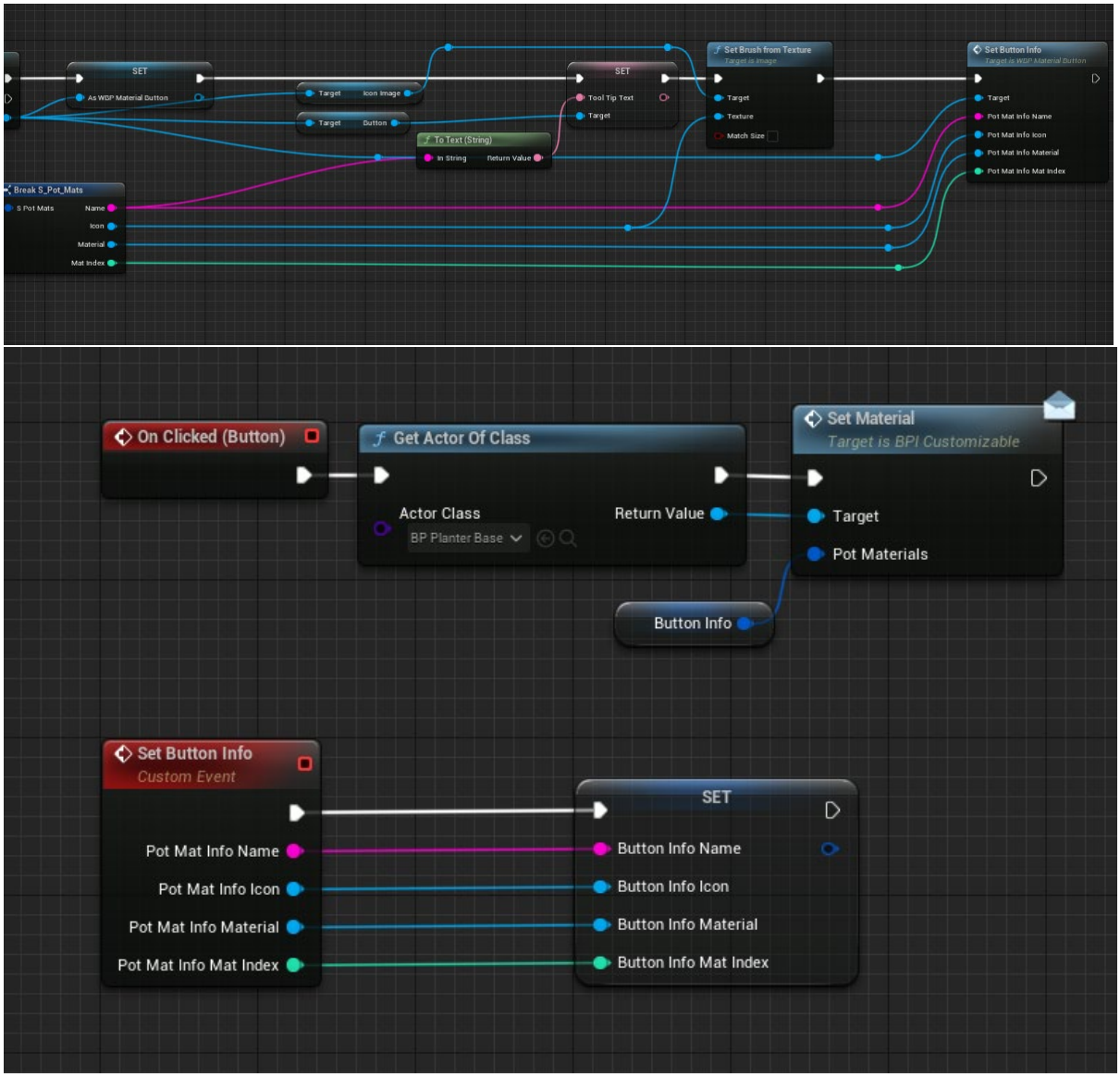
This really took a lot of staring at the screen and thinking. I'm not sure how to write the thought process because it was so jumbled. I created a structure to hold the information of the materials, icon, instance, and a name, and replaces the array within the game instance with this. Then I used this structure to populate the menu.



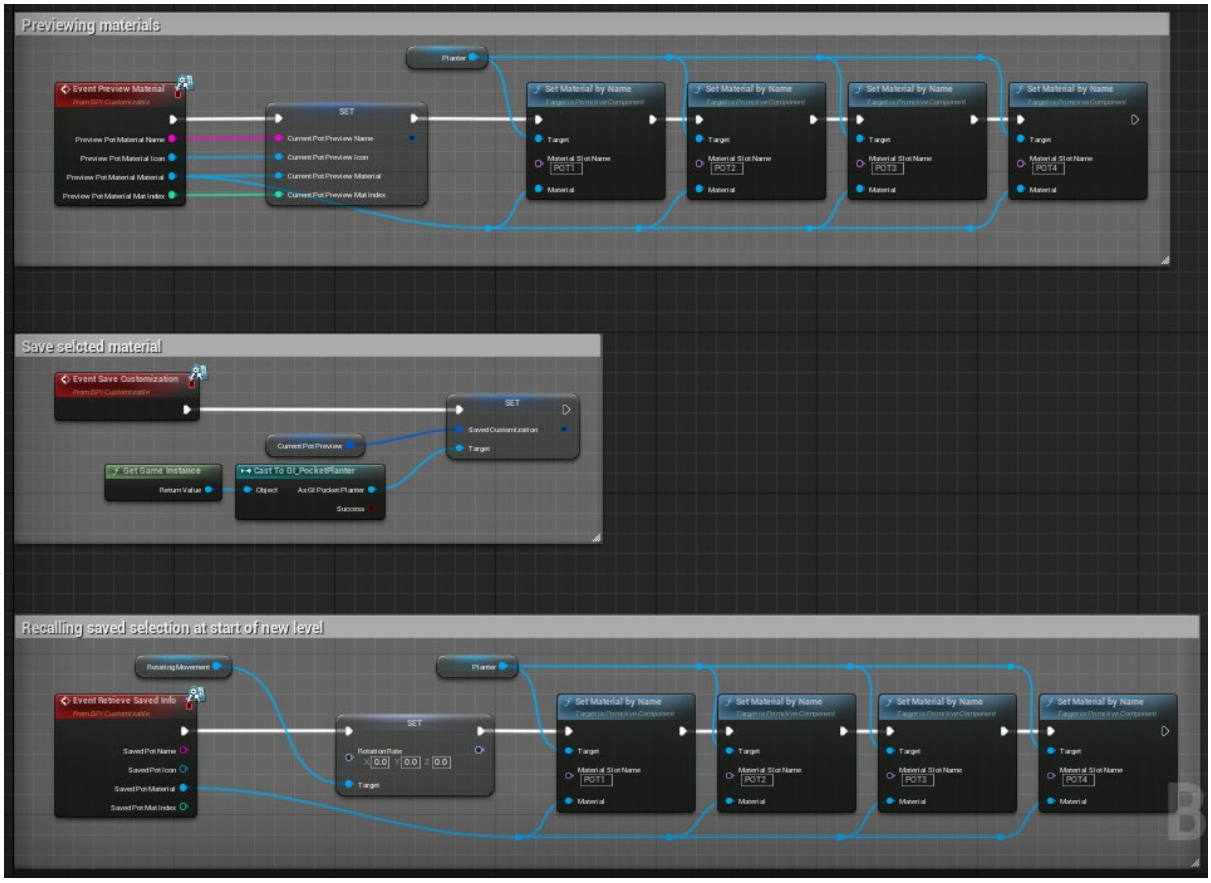


I got stuck when I was trying to set a number to each selected button as each button was essentially the same one populated with different info, and there was no physical blueprint for an item (Like the menu for other games I've done), I tried to add an index number within the structure but wasn't sure how or where to send than info to and from.

Eventually after lots of sitting thinking, talking out loud to myself about the logical path and many, many print strings later (also an engine restart because my print string just stopped printing.) I added the structure into the button and set a custom event to copy the info into each button. Then cast to the planter with a get actor of class and copied the same information again, into the planter.

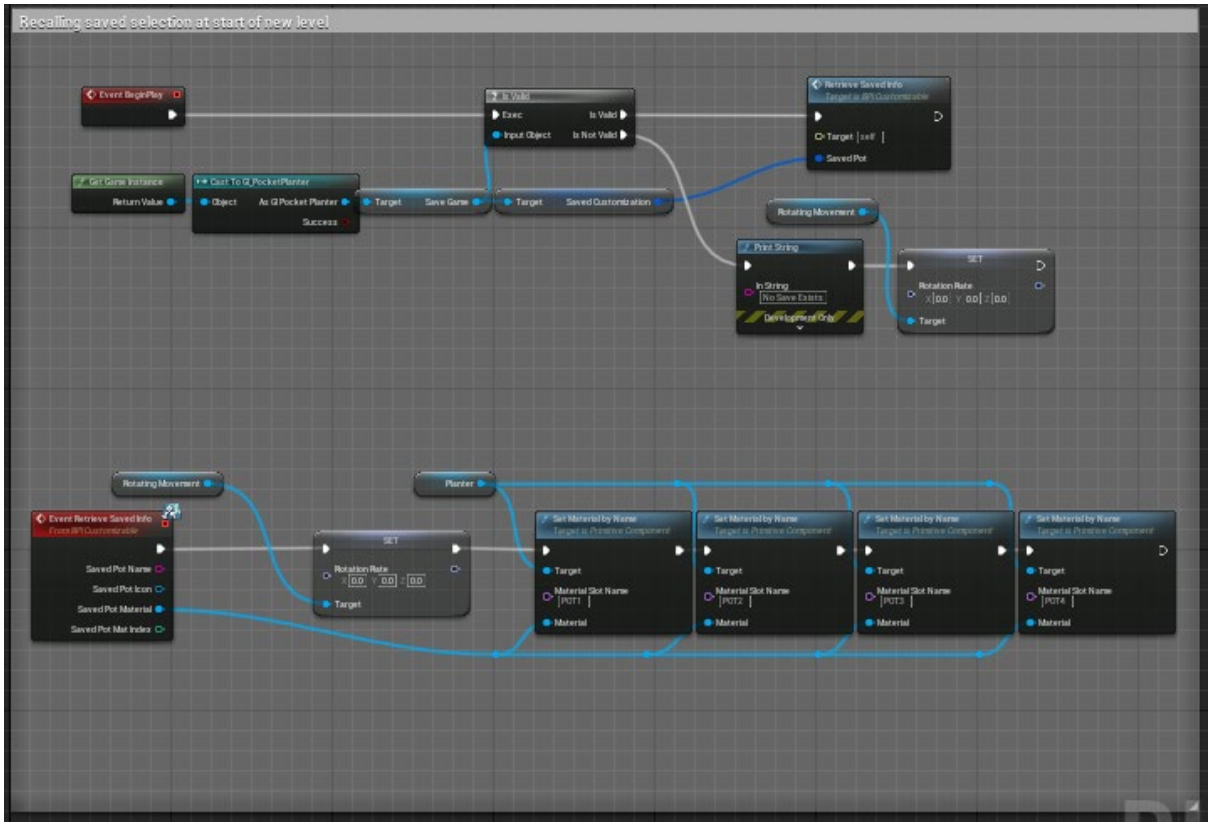


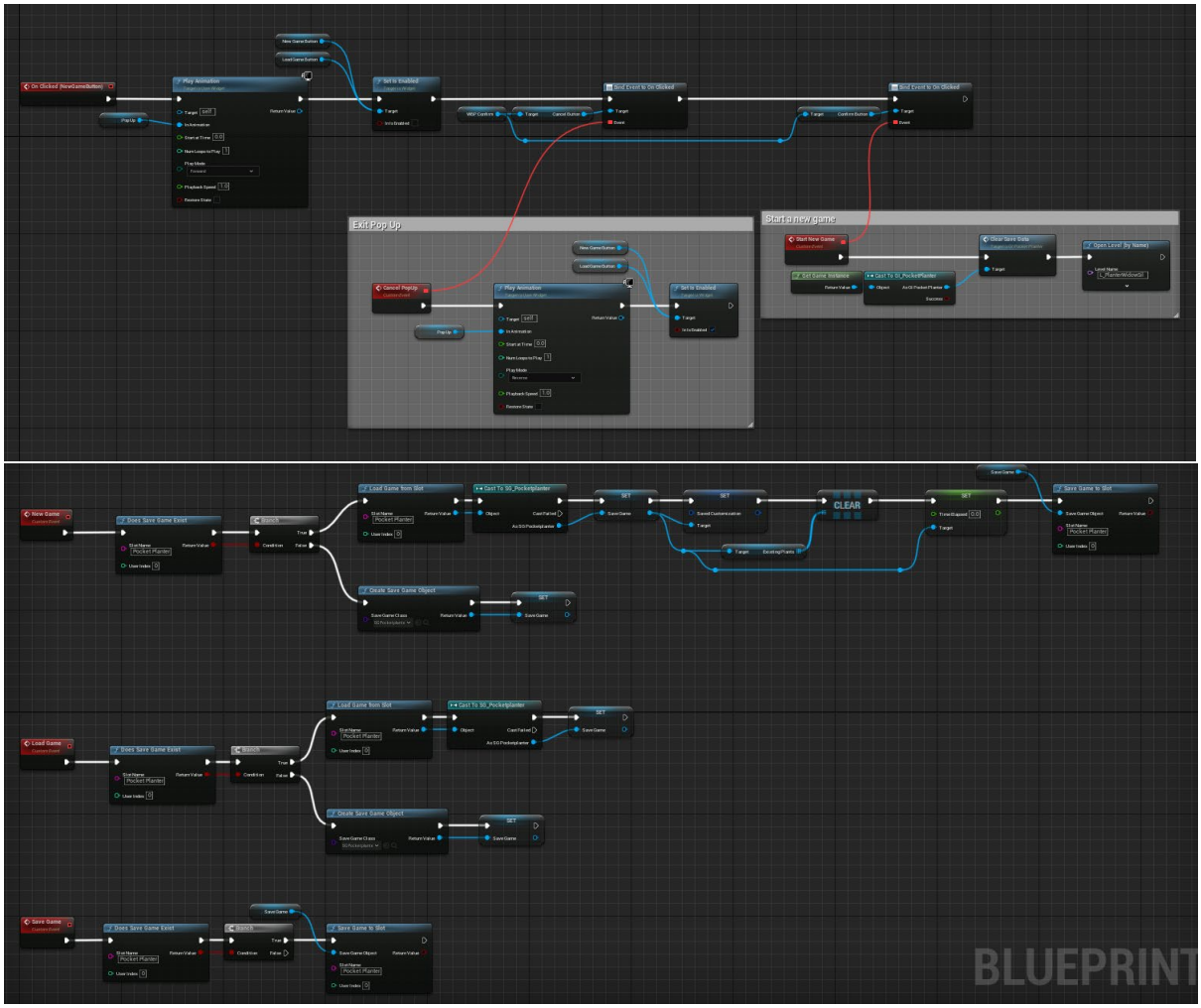
I now just want to add a confirm button and save the selection to be called when the main level is loaded. I added this functionality into an interface for customizable objects in case I ever want to expand on the customisation.



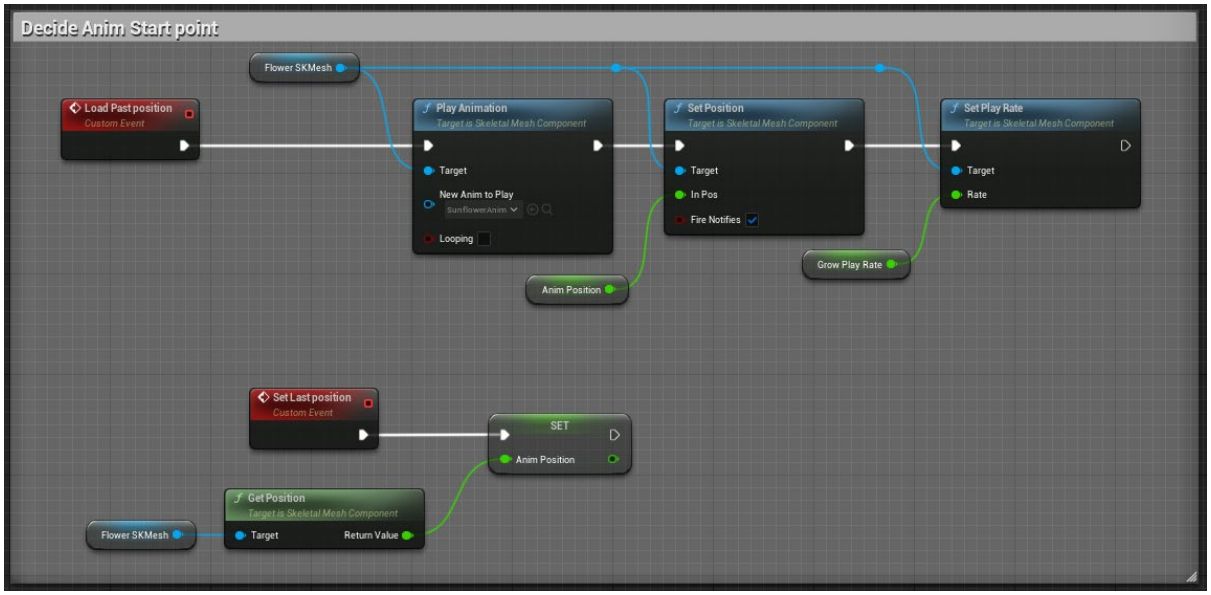
To this I then worked on some more UI Functions, adding stored information to a save file and connecting this to the menu safe/load function. I also added a pop up that notifies the player that starting a new game will erase the old one. Changing the above function a little to check for a save file before applying the saved pop selection.

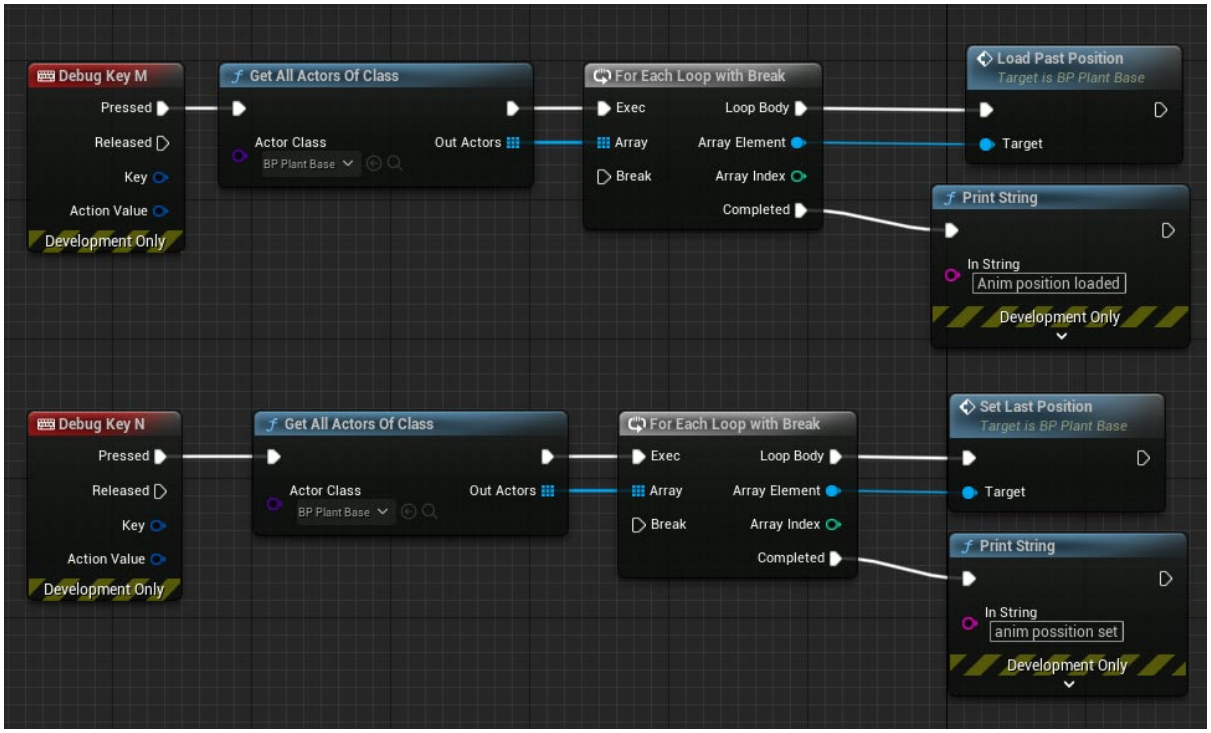
I held all the save/load/new game information in the game instance with a reference to the save game being created after the player selects wither load or new in the main menu.


Recalling saved selection at start of new level
Event BeginPlay
Is Valid
Exec
Input Object
Is Valid
Is Not Valid
Retrieve Saved Info
Target
Saved Pot
Get Game Instance
Return Value
Cast To GI_PocketPlanter
Object
As GI Pocket Planter
Success
Target
Save Game
Target
Saved Customization
Rotating Movement
Print String
In String
No Save Exists
Development Only
SET
Rotation Rate
Target
Rotating Movement
Planter
Event Retrieve Saved Info
Saved Pot Name
Saved Pot Icon
Saved Pot Material
Saved Pot Mat Index
SET
Rotation Rate
Target
Set Material by Name
Target
Material Slot Name
POT1
Material
Set Material by Name
Target
Material Slot Name
POT2
Material
Set Material by Name
Target
Material Slot Name
POT3
Material
Set Material by Name
Target
Material Slot Name
POT4
Material


Exit Pop Up
Start a new game
CLEAR
BLUEPRINT

Curresntly I have only the pot colour being saved in the save game, but I wanted to create a way for the animation to be saved and recalled too. After some research on epic forums wich pointed in all directions. I played around with a few of the suggestions. Most of the suggestions were assuming the use of an animation montage but I was working with an animation blueprint. Eventualy I decided on this simple set position right after playeing the montage, I also reset the play rate in case the player had affected this. Then I just tested it out with a button press and everything seemed to be working okay.



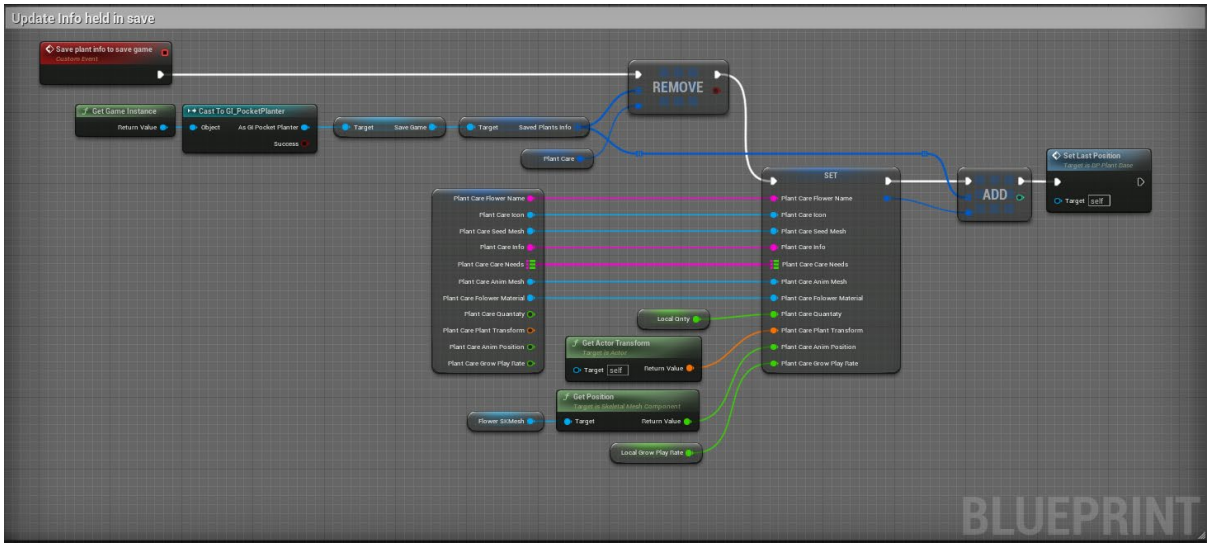


I thought the best way to save and recall all this information would be a structure, Sine the flower already has a structure, I could add to this then save the info in the save game. I will add the actor transform, grow rate, and animation position. Up to this point I've been testing this on one flower type, so before I add all this information to the save, I first created some different colour variations to represent different plant types.

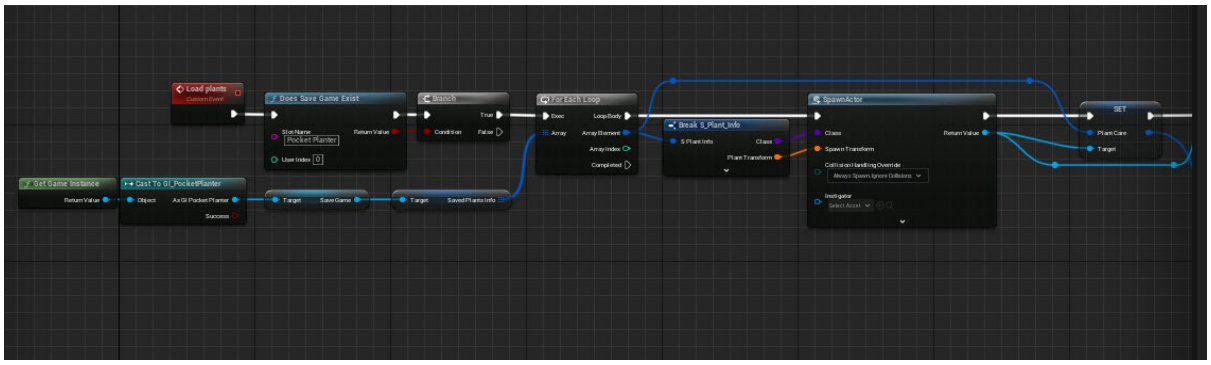
Save File Drama

Adding all the extra Information was a little finicky. I set up all the information to be updated on a button press for now as I intend the save to happen on existing the game.

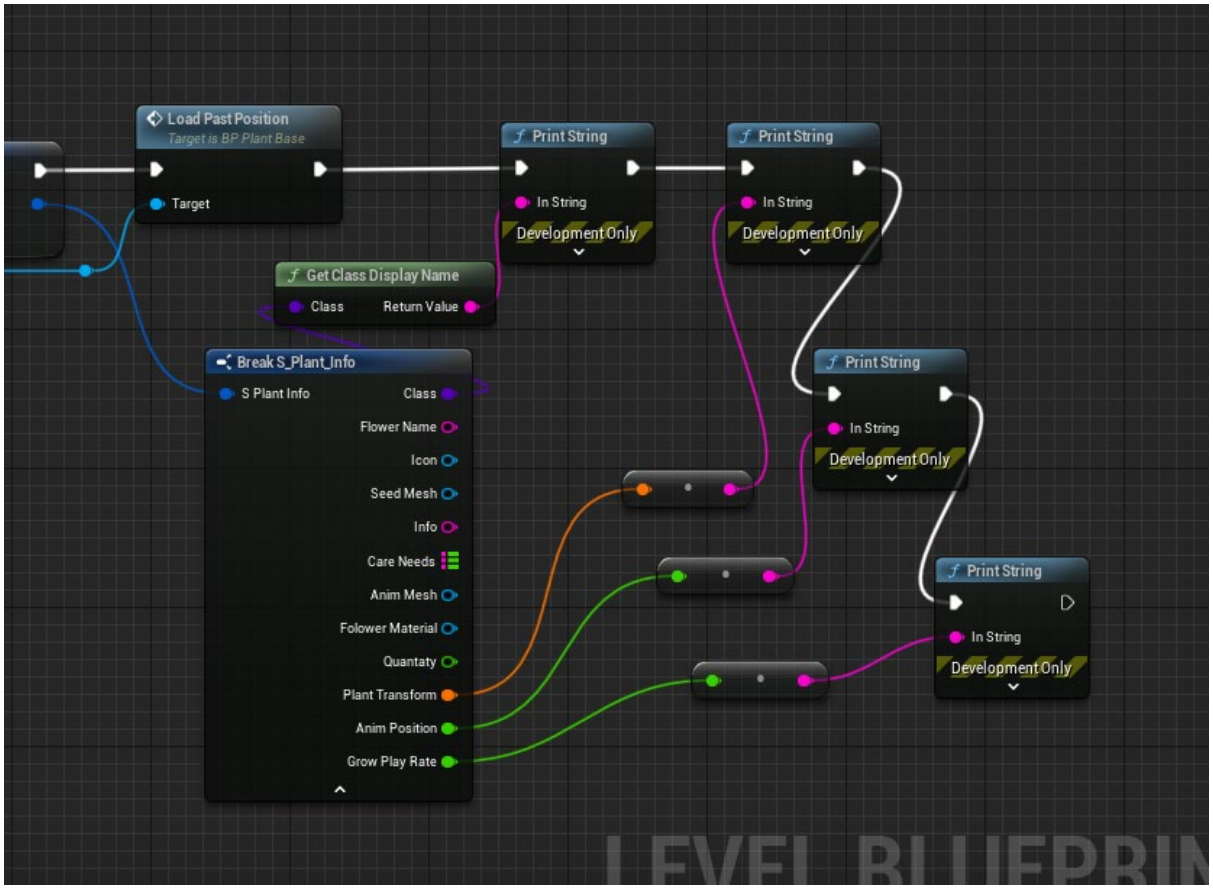
First, I created some child blueprints of the base plant in different colours for the different plant types. Within this blueprint I set an event via the plants interface to update the plant information. This information would update every time the game is saved (currently on a button press.)



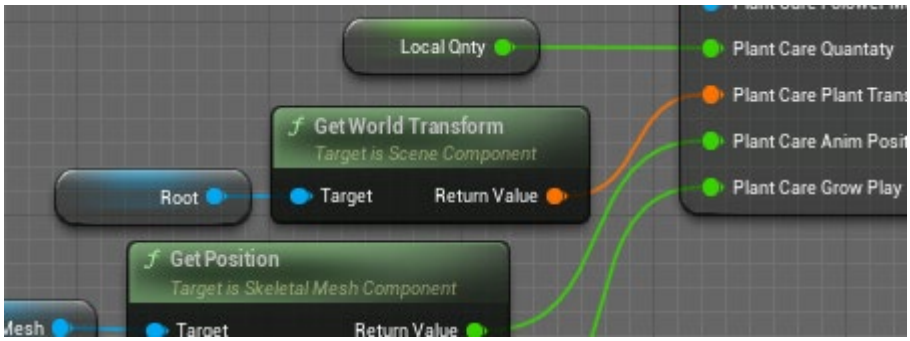
Then within the planting area blueprint I added any new plants to the save game as an array. And finally in the level blueprint I called a function to load in all plants in that array, as well as a clear array function on starting a new save file.



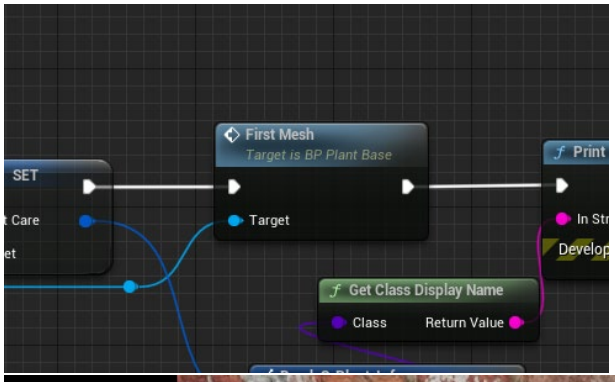
On first test, nothing loaded in, and I got errors. This was caused by several issues that I discovered in stages. The first was that I was that I was spawning in the base plant which held no class information. After correcting this, still nothing spawned in. I set some print strings to show information on screen to see what was missing.

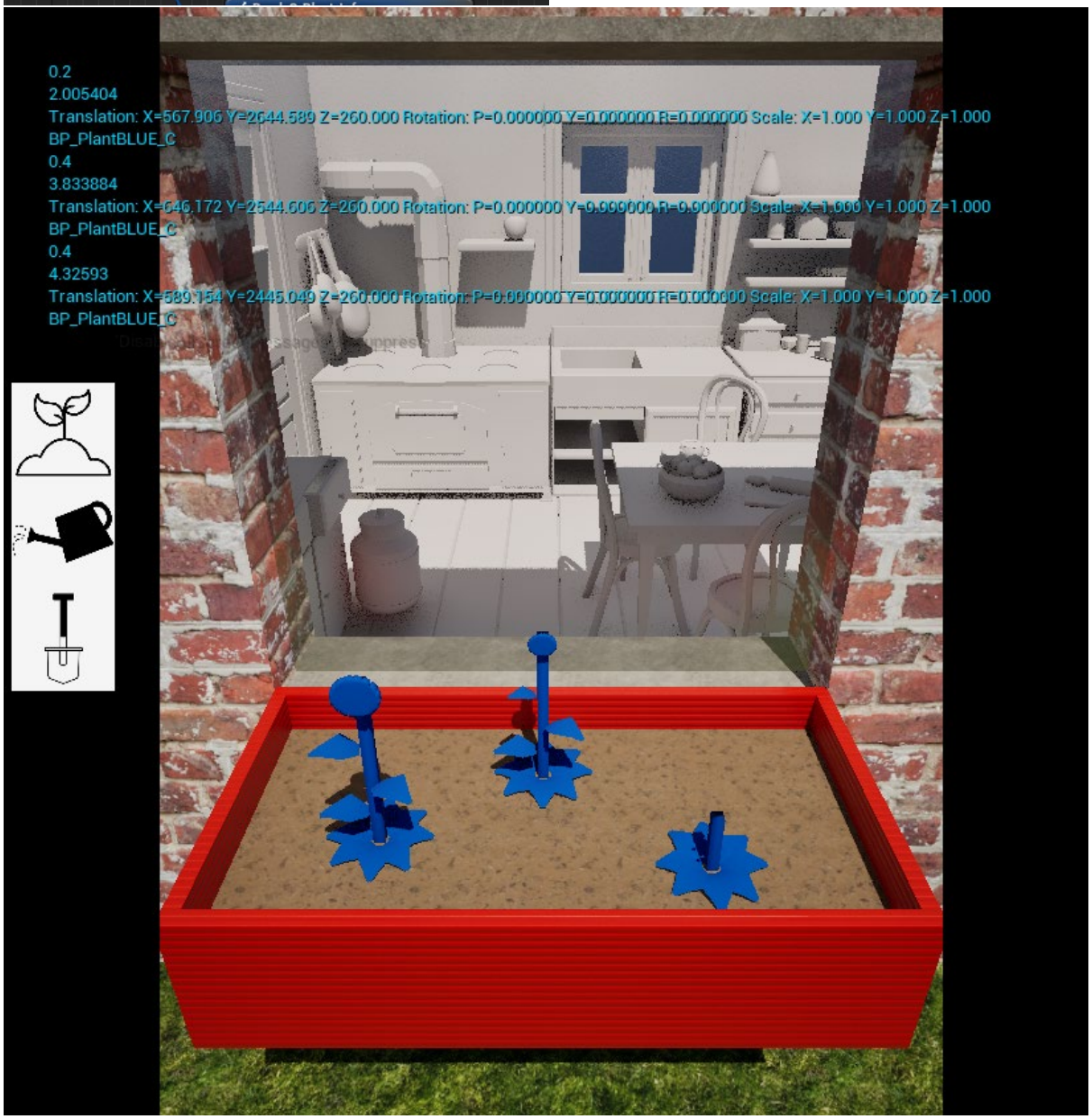


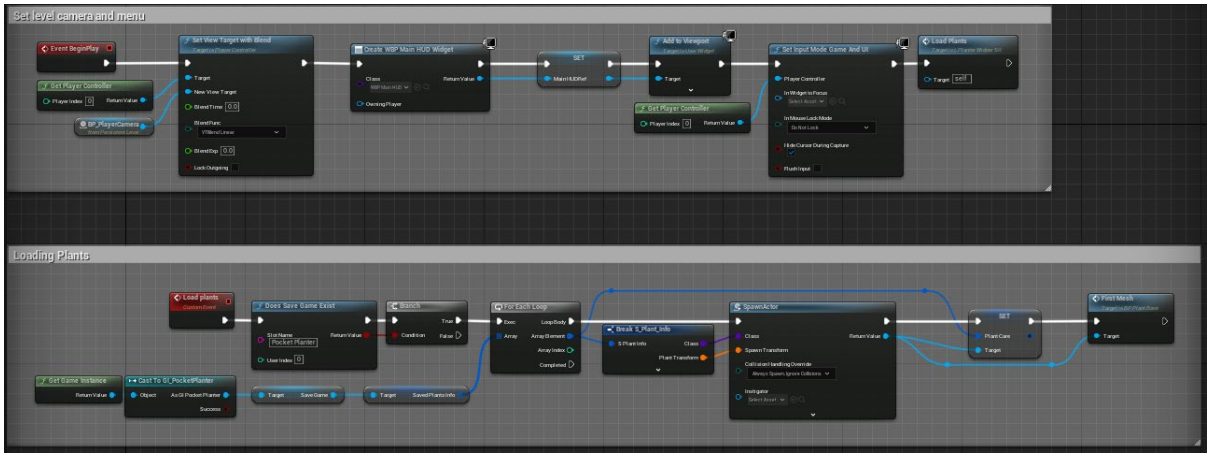
This showed that the transform location was at 00.00.00 due to the fact I had called the wrong transform data within the plant. After setting this to get world transform of the root, the data printed to screen seemed okay, but I still had no plants.



The last issue was that on calling the event to set the animation I was inadvertently skipping the part where I set the visibility. The function I had called was only intended for testing the animation logic, So I moved the logic in to the correct event line with the rest of starting actions were being called, and everything worked as intended.


SET
Care
First Mesh
Target is BP Plant Base
Target
Print
Get Class Display Name
Class
Return Value

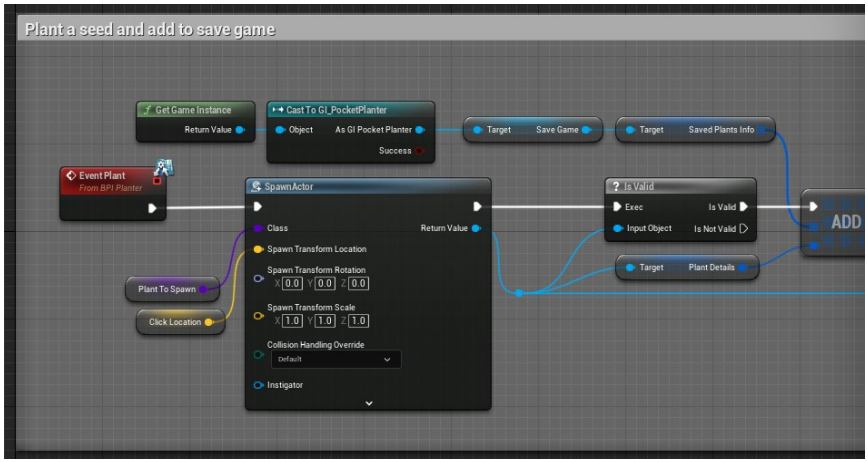

0.2
2.005404
Translation: X=567.906 Y=2644.589 Z=260.000 Rotation: P=0.000000 Y=0.000000 R=0.000000 Scale: X=1.000 Y=1.000 Z=1.000
BP_PlantBLUE_C
0.4
3.833884
Translation: X=646.172 Y=2544.606 Z=260.000 Rotation: P=0.000000 Y=0.000000 R=0.000000 Scale: X=1.000 Y=1.000 Z=1.000
BP_PlantBLUE_C
0.4
4.32593
Translation: X=589.154 Y=2445.049 Z=260.000 Rotation: P=0.000000 Y=0.000000 R=0.000000 Scale: X=1.000 Y=1.000 Z=1.000
BP_PlantBLUE_C

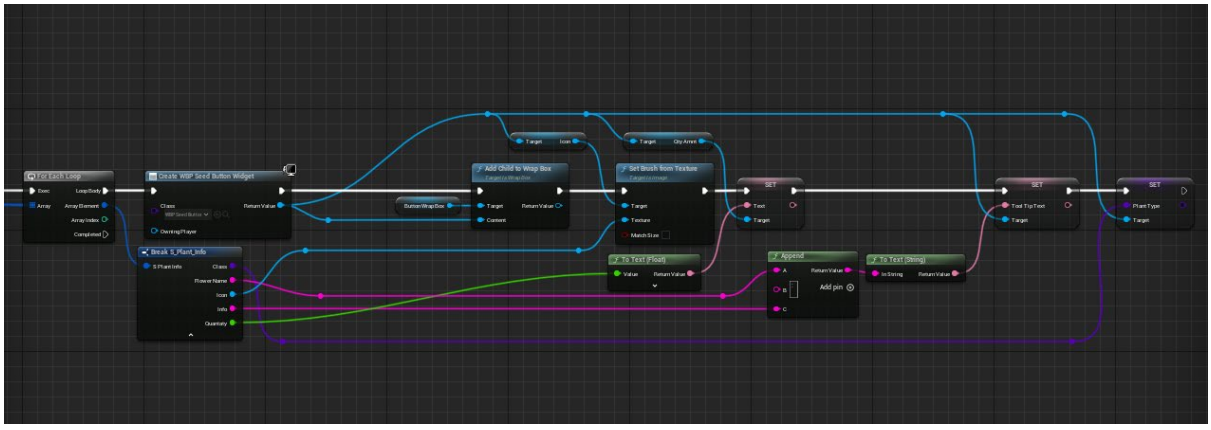


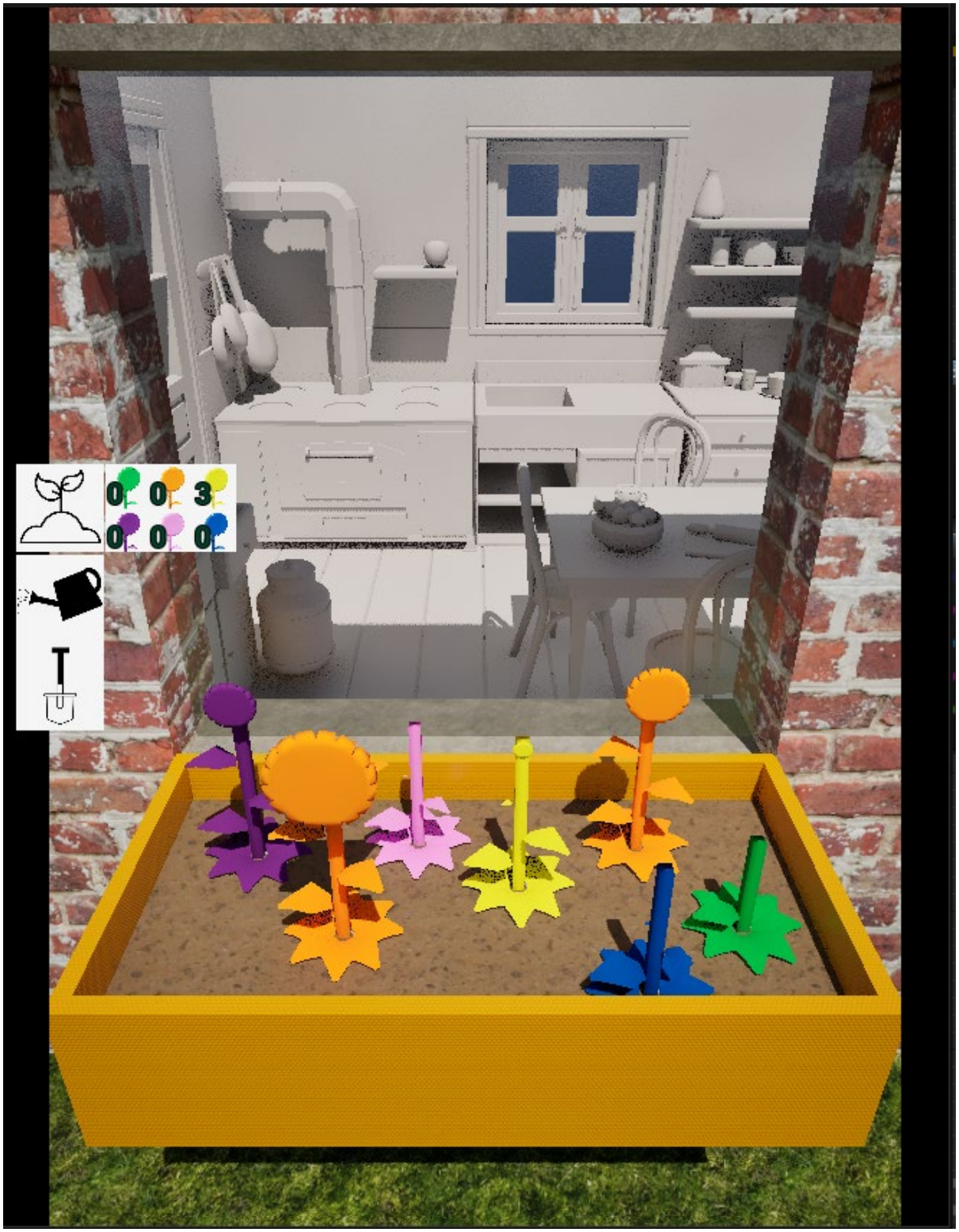
Pop-ups and Pop-outs everywhere

I started on creating the pop out menu to select the type of flower to plant. First thing I did was make a map of the available plants just to hold the information in. I held this information in the save game to recall easily. I then made a menu with all colour options and in the widget blueprint for the pop out menu, used a for each loop to fill in the information. This worked well, then I made a short animation to show and hide the menu.

To get the correct plant to spawn I changed the span actor to a variable and used the menu to send that variable information to the Planter bluprint before it called the spawn function.






0 0 3
0 0 0

I tried to make the menu a little more appealing by adding colour and realise that the pop-up menu also needs to be hidden when clicking away from the menu.

I spent an ungodly amount of time trying to get the pop up to hide when clicking away and ended up using a variation of this from a forum:

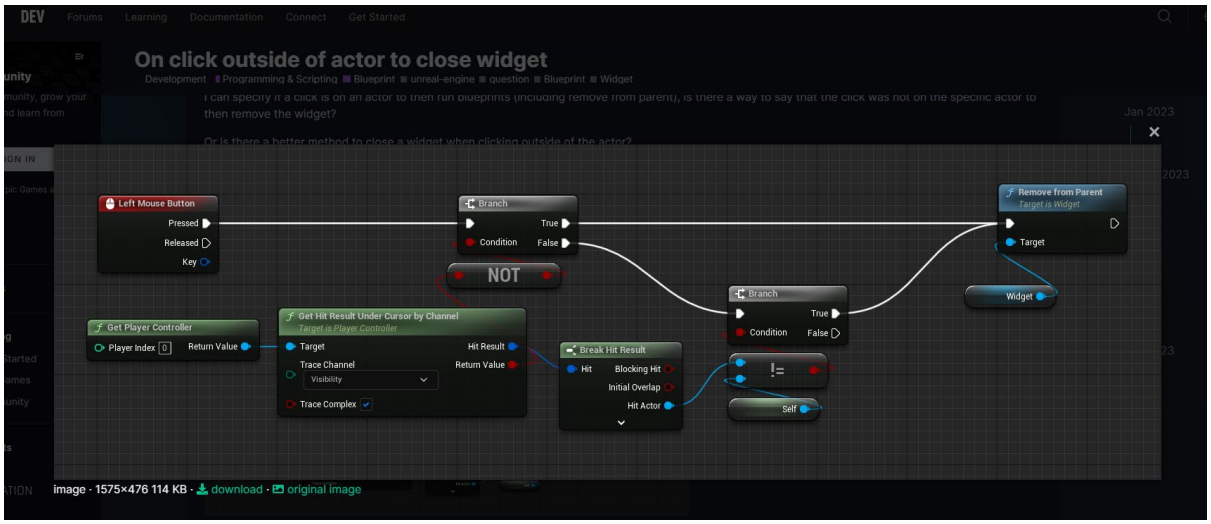


Figure 1: Forum version.

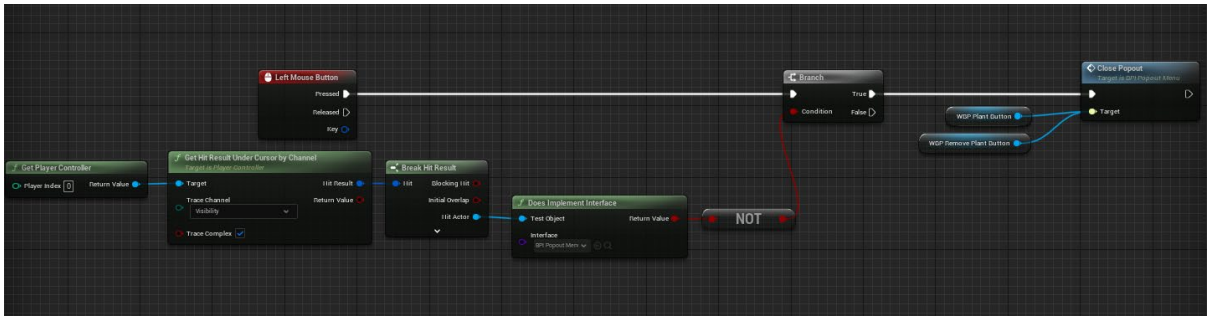


Figure 2: My version.

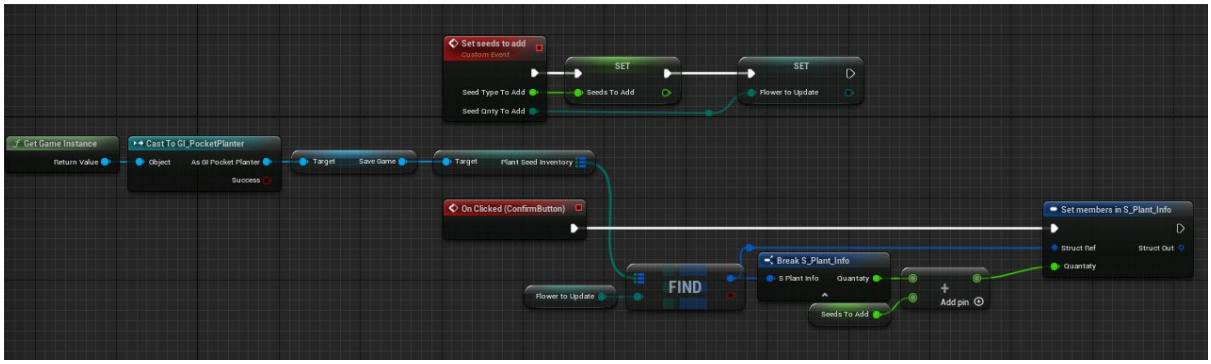
Only this worked when clicking somewhere else in the world, not when clicking other menu elements. For now, I left it like this, because I actually spent about 3 hours on this one little thing, and I cannot face sinking more time into it for such small refinement. For the sake of getting the project finished on time, this is something I can come back to if I have time.

In the menu I already prepared the numbers and hooked up the quantity to the pop out, and I'm still unsure how the player will earn new seeds, or how they will earn money to plant seeds. Looking at other games for inspiration there are a few directions games tent to go, purchasing seeds, collecting one a freebie once a day, or earning money though achievements to buy with in game currency.

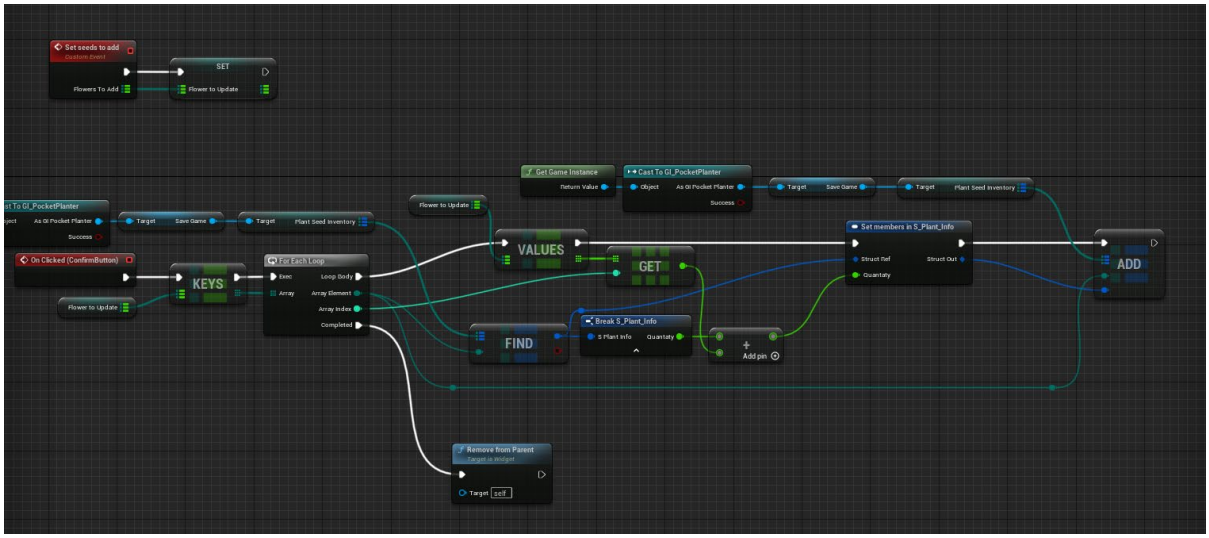
For now, I just want to implement another pop up that notifies the player that they have new seeds and use this as a starter pack of seeds at the beginning of the first play and create something to expand on later.

Pushing My Knowledge

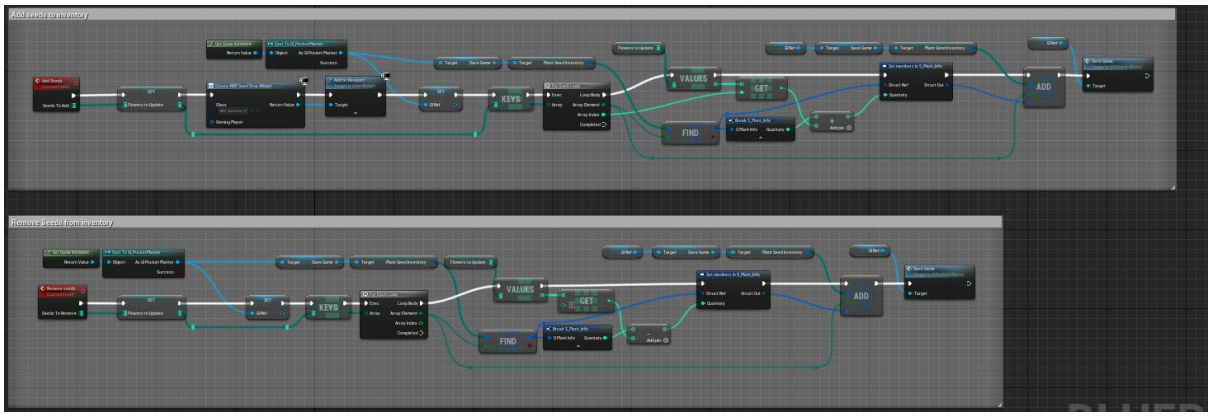
I made the pop up and in the first attempt I had put the logic for adding the seeds inside the button. This aside, my logic was full of issues. I'm not actually even sure how to describe the headache that followed, but I think I was really bushing the boundaries of my knowledge, and I *know* my code is a mess! I had a map within the save game, with a Enum of the flower types and their quantities and I had no idea how to pull the type I wanted and change just that one value. I tried first with a custom even being called in the level to add a starter amount of seeds.



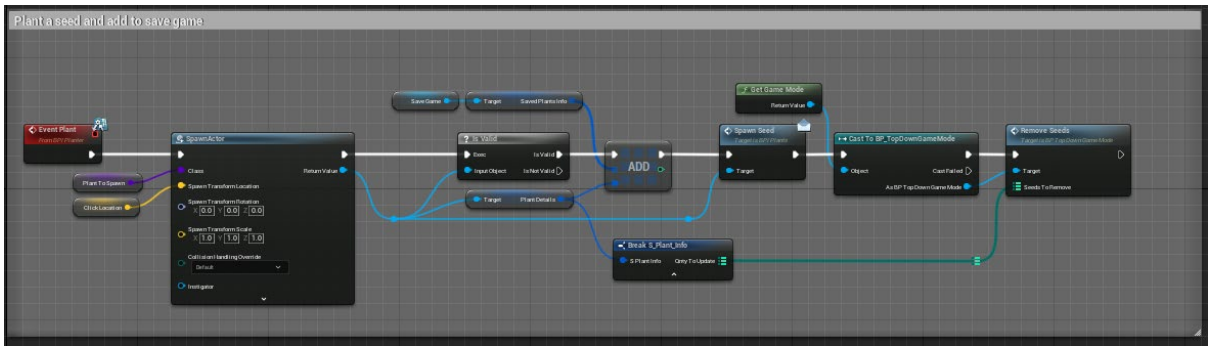
But I didn't know what to do with the Set member node. I did some research on the epic forums and managed to figure it out after lots of trail and error.



This was, more or less working, but all the logic being in the widget made no sense. I started to move this all over to the game mode as two functions (add and subtract.) so that the same logic could be used for getting seeds from any source in any quantity. I also during this made the quantity an integer as it had been a float the whole time.

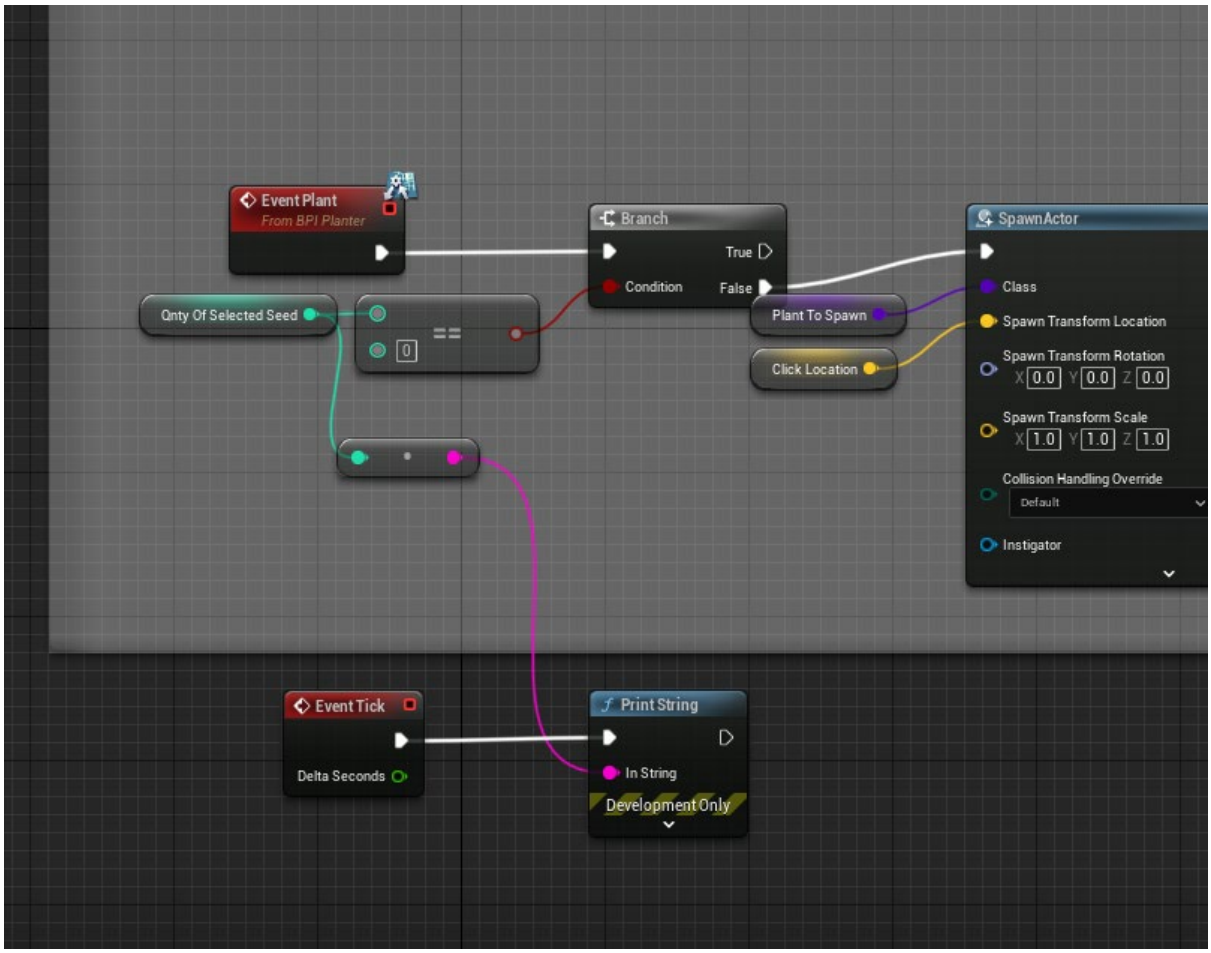


After this I used the function to remove seeds right after spawning a new plant.

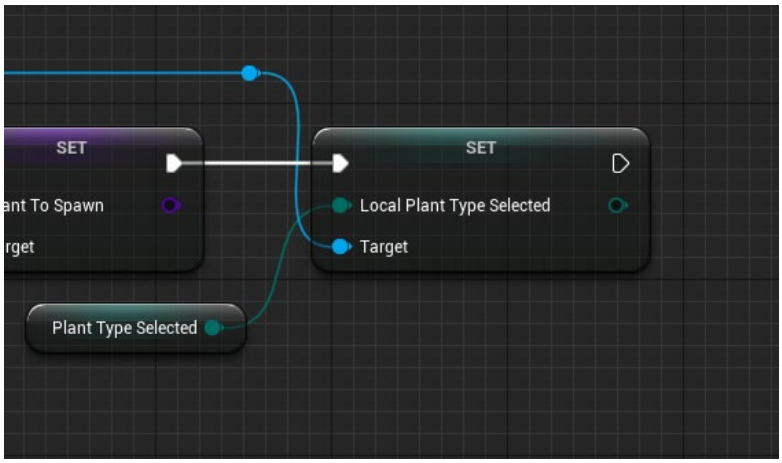


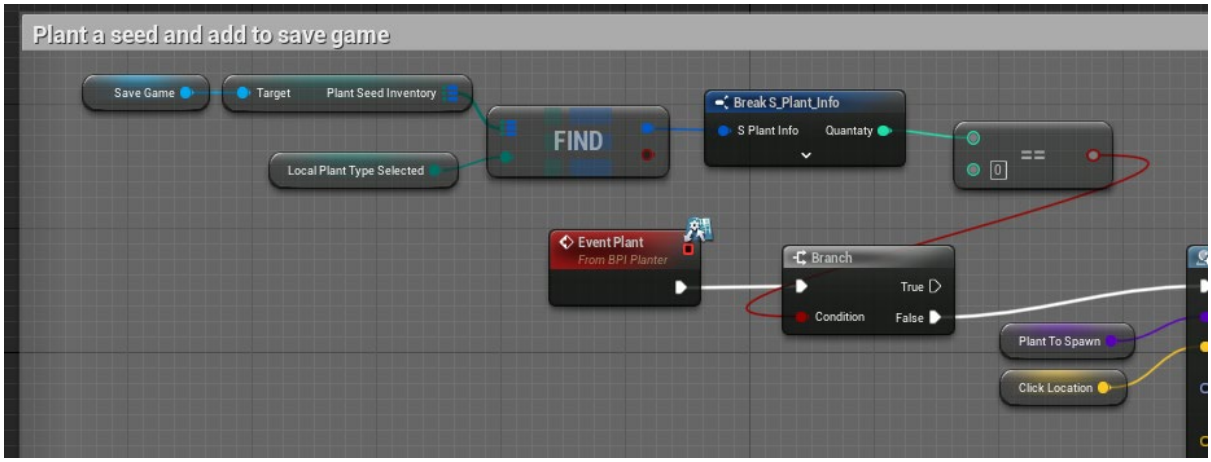
Things got messy when I realised that the selection of the seed has no idea what type of seed that it is trying to spawn before it's clicked. So, all the information I wanted was too late in the sequence of events. I considered spawning to 0.0.0 and then moving the mesh on a click, but this felt unnecessarily complicated and like a bad work around. I tried a few things, trying to copy the quantity data over from the button selector, but the way I had set it up meant that the quantity was being called only as the button was pressed, it's not bound to the save data. I used a print string on a

tick a number of times to get a better look at what was happening and just kept testing things like this.

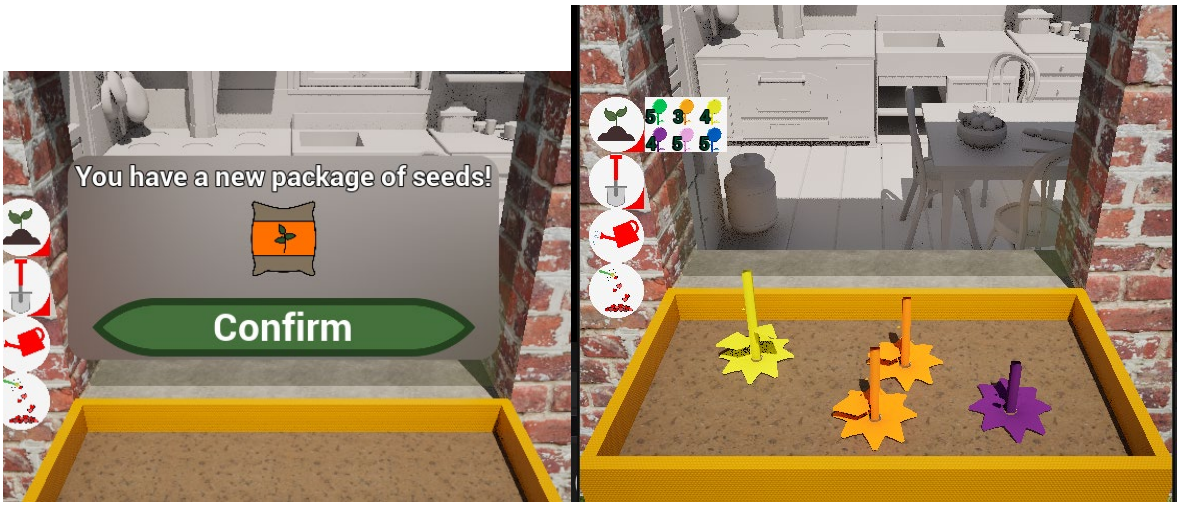


Eventually I used similar logic to how I was calling the class to spawn, adding an Enum to the existing data struct and copying the selections to a local variable within the plant spawner. Using this to select which quantity value to check before spawning.





This finally did work, although it just feels messy, I don't have an alternative solution at this point. And most importantly, it saves and loads as intended. Yey!

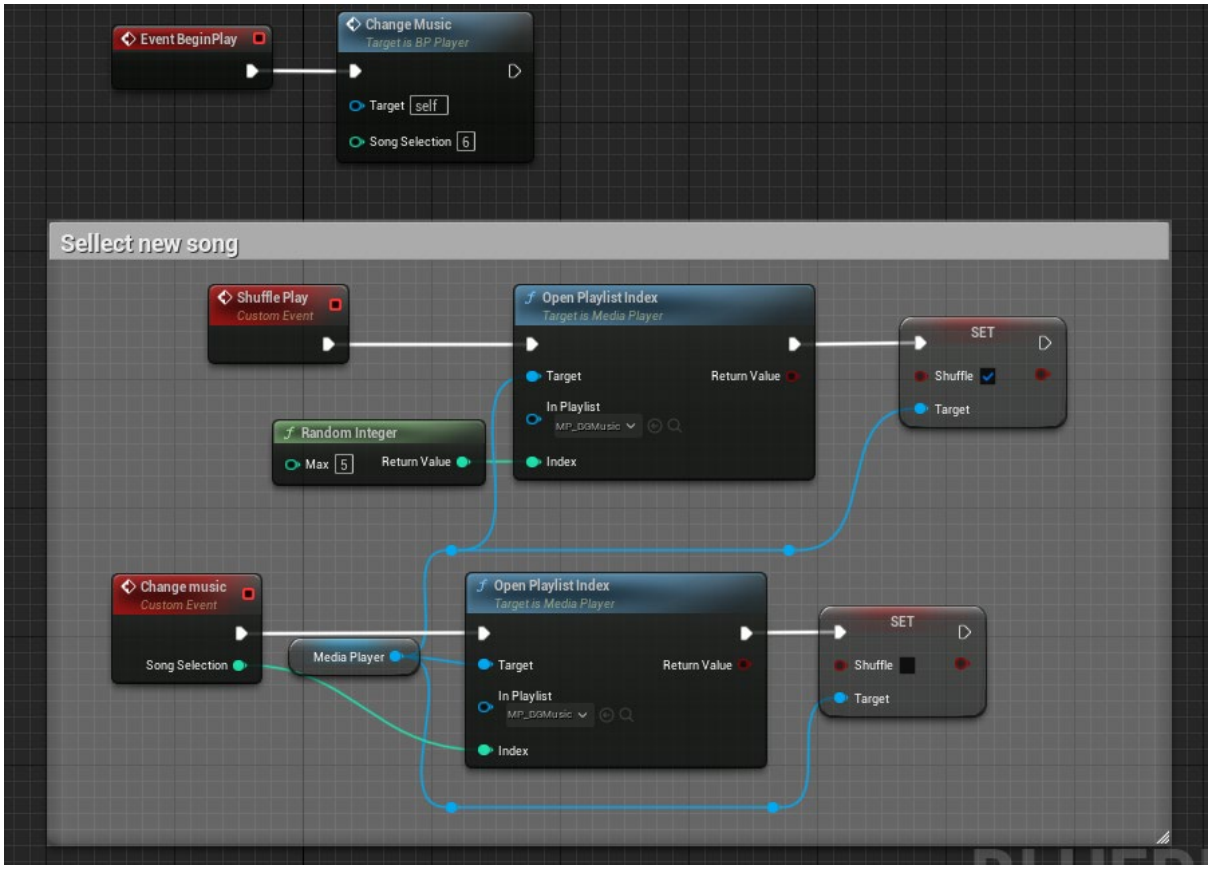


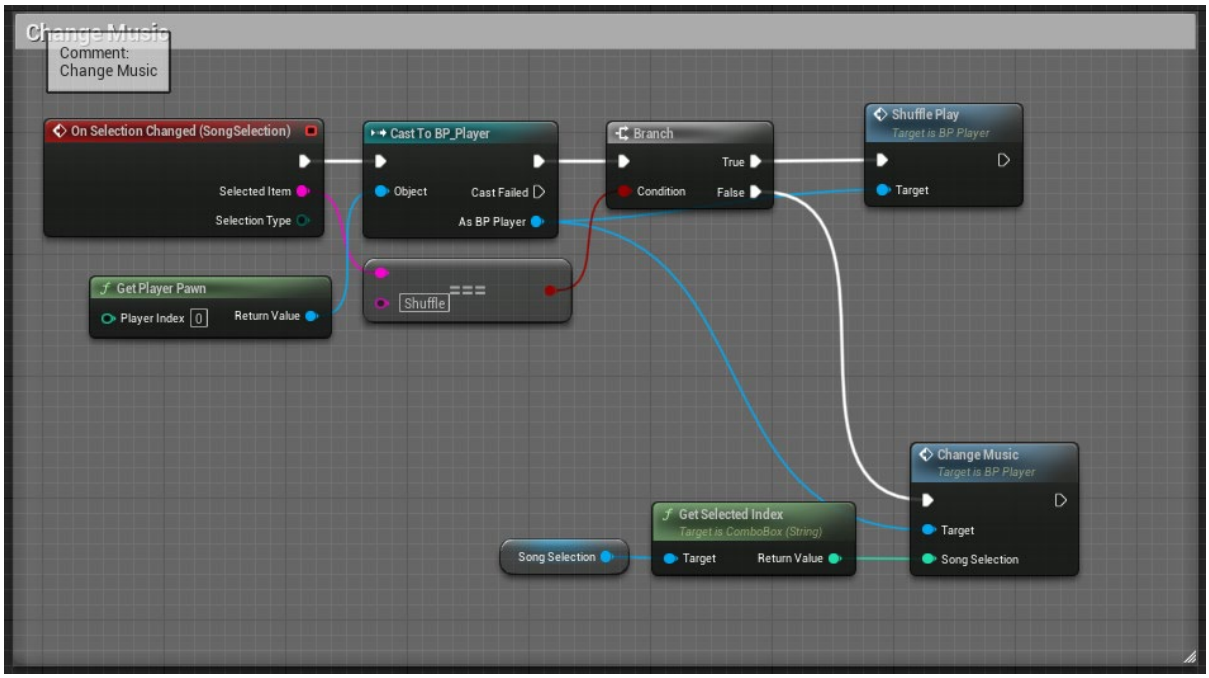
Music to my ears

I jumped the gun a little bit and decided that now was the time to add music. I was happy to learn how simple this was. Before doing any research, I set up an in-game wav sound and was going to randomise it, but I decided with this being a calming, therapy type calm, I wanted the player to be able to choose the music, as calming is different for everyone. I chose some songs from a free library of sound file and did some research on how to set up a music player.

https://www.youtube.com/watch?v=ii0r6AzXUFM

I found this really easy to follow video giving details on how to set up a radie within the level with MP3s rather than WAV files. After setting up the correct file path for each song and creating a playlist, adding the UI to switch between them was very simple.

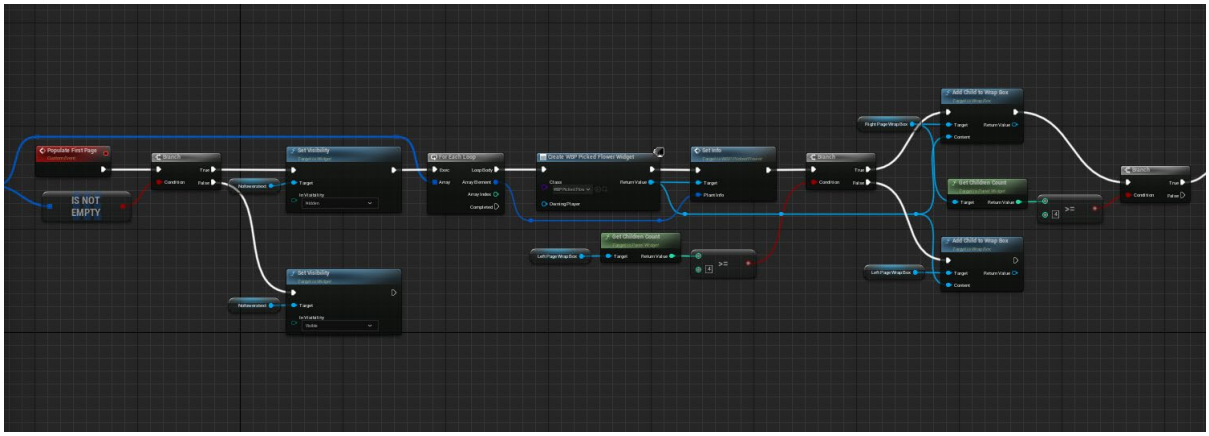

Event BeginPlay
Change Music
Target is BP Player
Target self
Song Selection 6
Sellect new song
Shuffle Play
Custom Event
Open Playlist Index
Target is Media Player
Target
Return Value
In Playlist
MP_DGMusic
Index
Random Integer
Max 5
Return Value
SET
Shuffle
Target
Change music
Custom Event
Song Selection
Media Player
Open Playlist Index
Target is Media Player
Target
Return Value
In Playlist
MP_DGMusic
Index
SET
Shuffle
Target


Change Music
Comment:
Change Music
On Selection Changed (SongSelection)
Selected Item
Selection Type
Cast To BP_Player
Object
Cast Failed
As BP Player
Branch
Condition
True
False
Shuffle Play
Target is BP Player
Target
Get Player Pawn
Player Index 0
Return Value
Shuffle
Change Music
Target is BP Player
Target
Song Selection
Get Selected Index
Target is ComboBox (String)
Target
Return Value
Song Selection

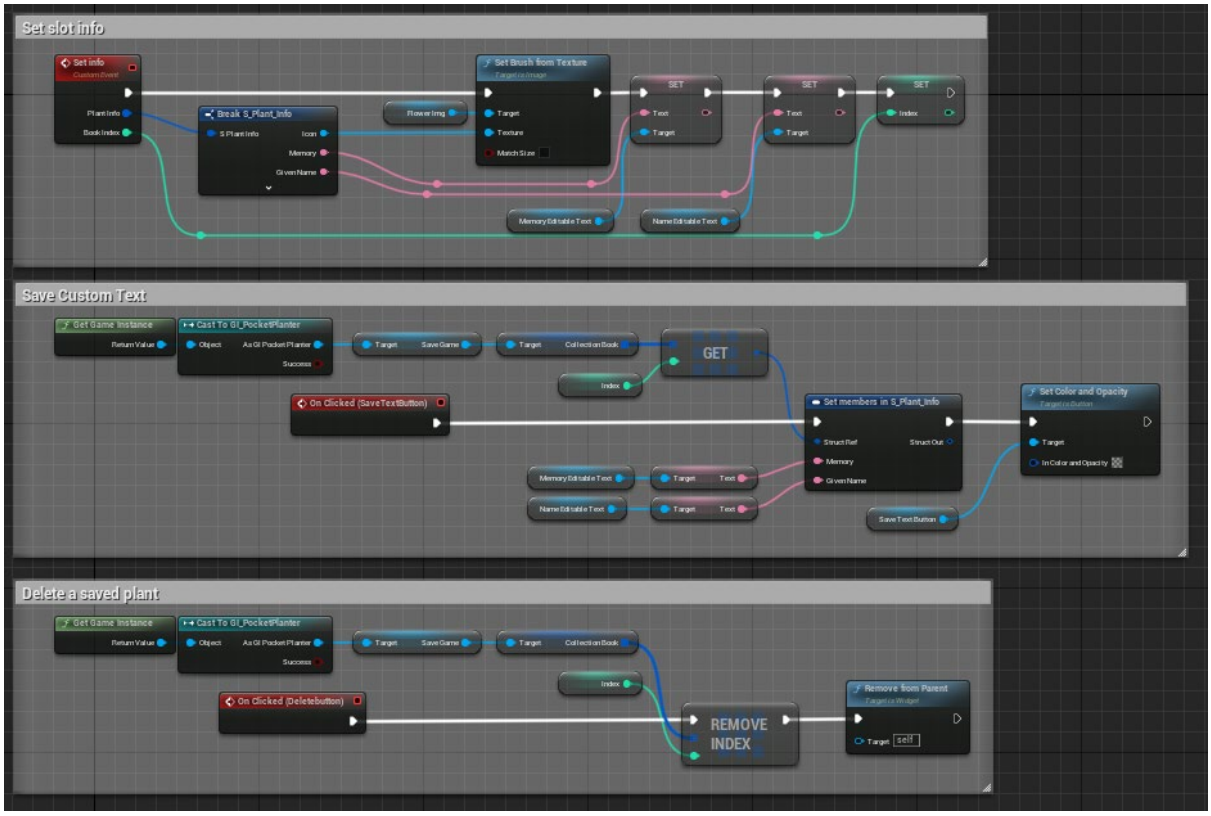
Flower Hoarder

After this quick and easy addition, I moved on to making the book menu for the player to save plants. I wanted to give the player a space to not only keep the plants, but to have the option to name them and write about them too. I added more variables to the plant info struct and used this to fill in the menu and save the text. This was a good expansion on what I had already done with a bit more complexity.

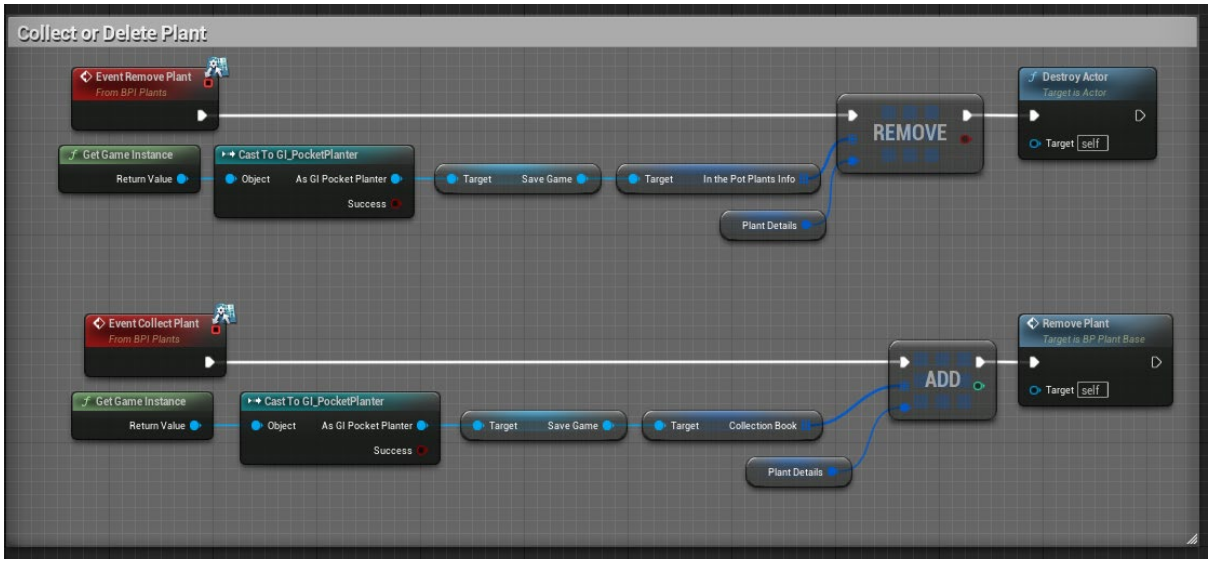
When adding all the information though, I wasn't sure how to add extra pages. I could only hold 4 on each page and had set up a Boolean to swap to the next page after 4. But I wasn't sure how to add more pages. I have capped the saved flowers to 8 for now pending more research on widget switchers, or something similar, to get the desired effect.

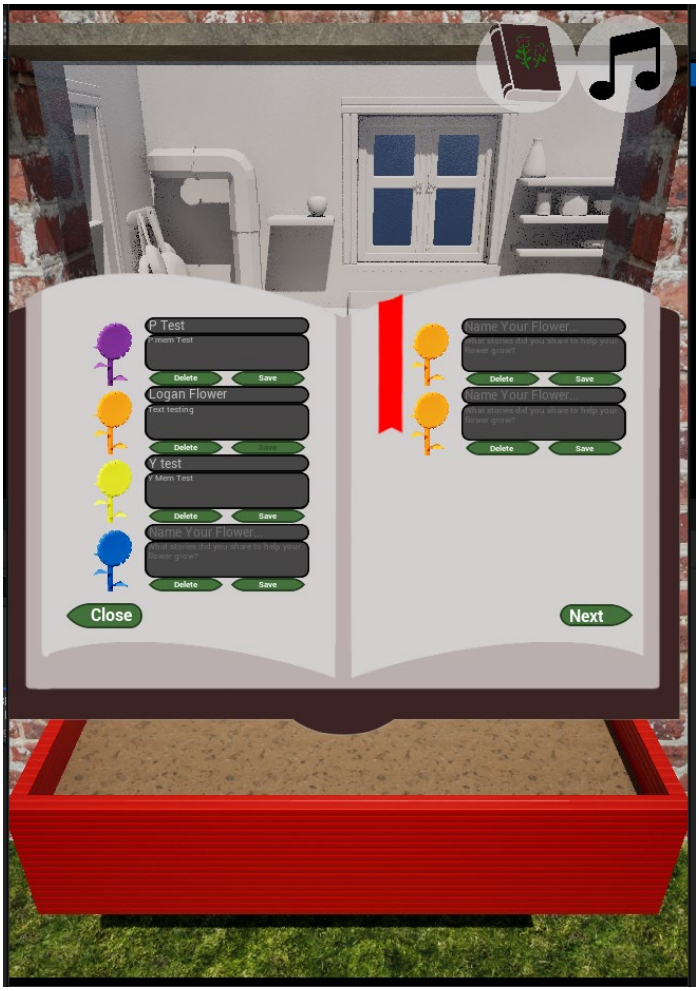


Then I set the function to save information within each slot.



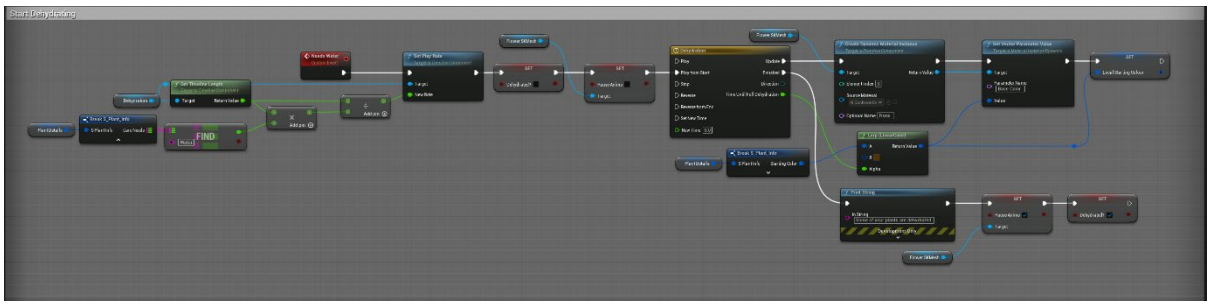
Finally, I added an on flower picked event to add the information to the save game.





Turning Brown

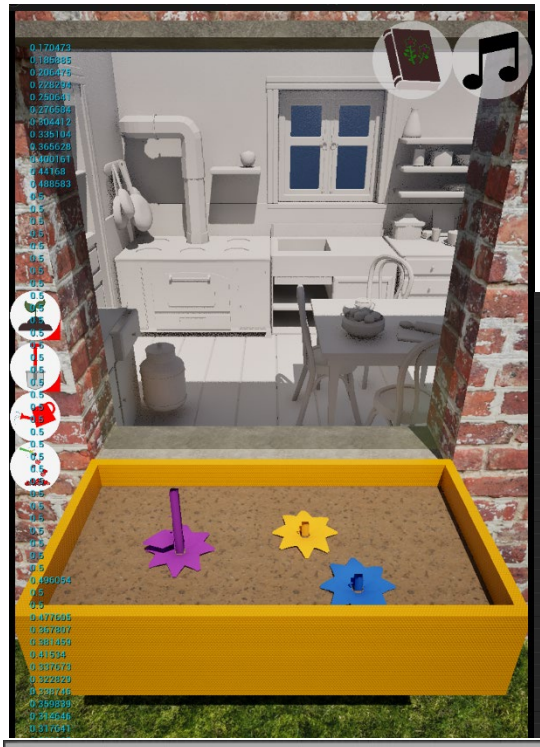
I wanted to create a visual cue for the plants needing water, without animations of the flower wilting, I decided to have the flower turn brown using a timeline, and upon finishing the timeline, the animation would pause.

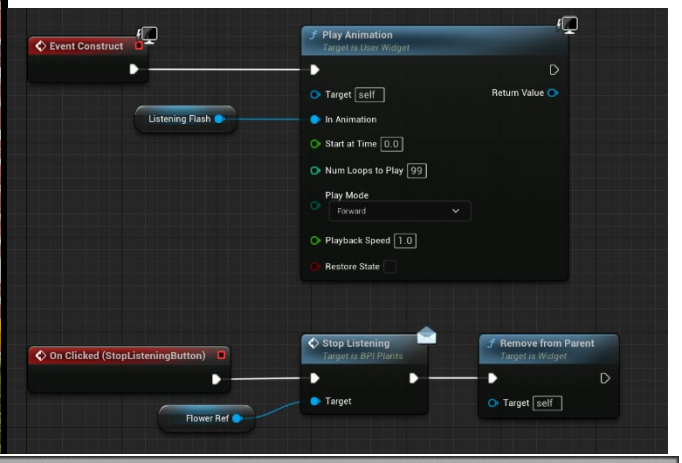


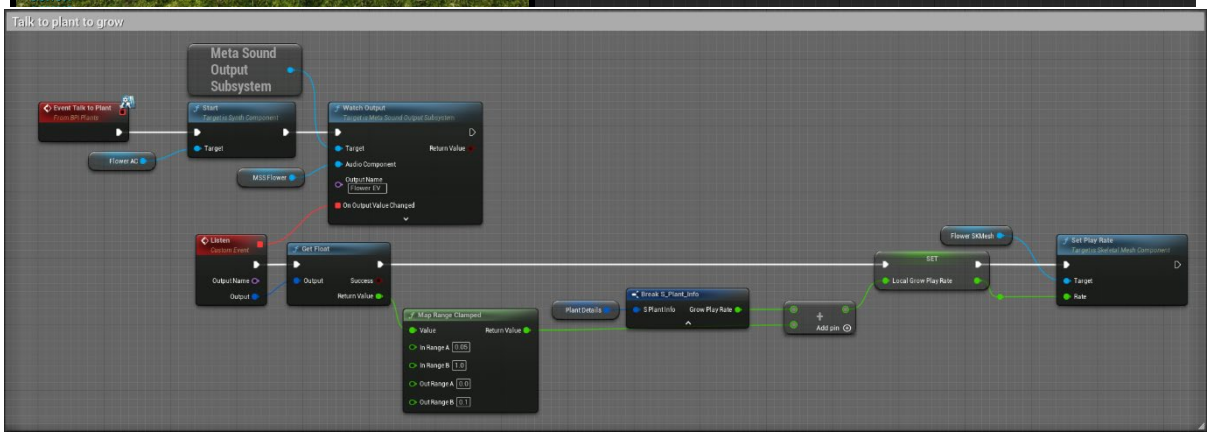
Sound... 3rd attempt!

I had to return to the microphone input as it was one of the things I was most excited about when I decided to make a mobile game and it really plays into the feeling that this game is like a therapy game that encourages players to talk more. Luckily on this attempt I found this YouTube video: https://www.youtube.com/watch?v=hek6e_mWudc This being a more sound FX focused account, it went into much more detail about how the microphone input works, and how to change different settings using meta sound. Finally, I had the knowledge needed!

I did some initial testing to get the sound input right and balance how much noise would trigger the plant growth, so the player had to be loud enough to be considered speech and not just all background noise. Then I hooked this up to the care button to trigger the sound input. I decided to do this per plant, so each plant type can have a favourite type of story, playing to them being therapy plants. I then made a small UI pop up to signal that the phone is listening and letting the player click stop when done.





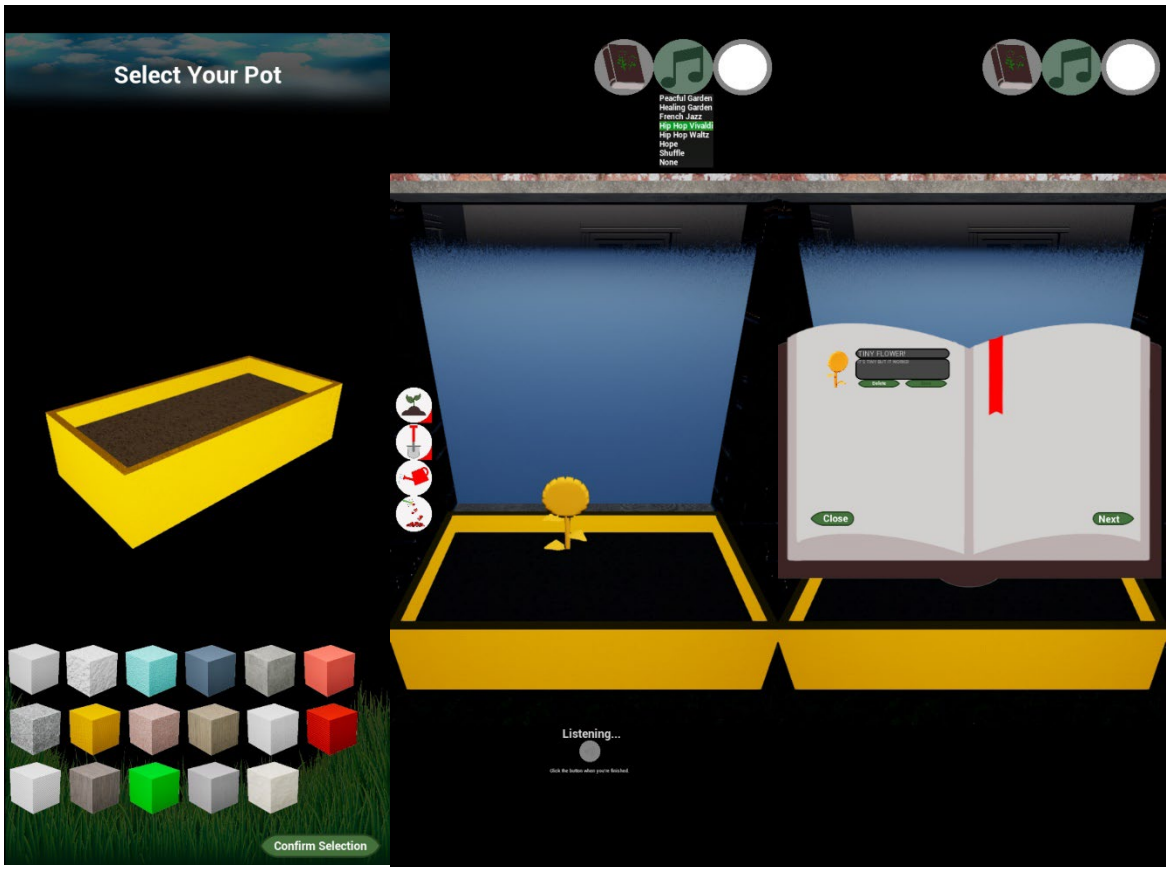


Android Packaging

With all this done and the time to package the game coming up, I decided to return to give the Android packaging another go. This time, incredibly, the software I downloaded worked! I could hardly believe it! Excitedly I put it on my phone to be met with more issues. A lot of back and forth between forums and repackaging later and I managed to get it to work on the mobile! The only issue is that I had to reduce the textures so much that some simply don't load anymore. Definitely a lesson for the future! I used a free texture pack and some of the textures were likely above what was needed. Admittedly some of the issues I didn't understand, I simply followed the advice, but I feel it's a really good starting point.

Below are some screenshots from my phone of how this looks. It's also now clear that the text needed to be much bigger! Bonus points to anyone that can read what I wrote in the flower book.

Overall, I'm happy that the functions still work, and if I was to redo this project and managed to set up the packaging earlier, these are all issues I could have spotted and solved earlier.



Reflection

Overall, I have had so much fun with this project, it's probably been my favourite to date. Though the gameplay is very simple, I think it fits within the idle game category well and does have enough features that make it fun to play. I am happy with the book collection functionality, the music player and speech, and that it does (more or less) work on mobile. This is a project I see myself returning to polish and fix when I have the time because I would like to add animations, different plants. I would have loved to add a little creature or insect that the player could tap for inspirational quotes.

Another thing that isn't available in this slice of the game is the ability to earn or buy more seeds. This is something that I planned to add but ran out of time.

I learned a lot about UI implementation, meta sound, how to work with animations in blueprint and I've created something I am really proud of.

Links To Assets Used

"Sunflower Seed" (https://skfb.ly/68WGN)

"Vintage Kitchen" (https://skfb.ly/6V8qU)

"CC0 - Drops" (https://skfb.ly/o6qEN)

open book by Rob Crosswell from Noun Project (CC BY 3.0)

bin by laili from Noun Project (CC BY 3.0)

seeds by Graphic Nehar from Noun Project (CC BY 3.0)

"Cute Spider | CCW" (https://skfb.ly/oFSyv) by Daniils Bobers is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"Bee (Low Poly)" (https://skfb.ly/o9XvJ) by EsiHere is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"Snail stylized uWu" (https://skfb.ly/owr8x) by Kyyy_24 is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).