Project Process Documentation

Team Members:

Anis Setoodeh (101278866)

Elias Erek (102578318)

Kyle Gibbs (102107602)

Wei Yik (103838398)

Last Updated: 07/08/2023

Table of Contents

Table of Contents 2
Process Documentation Template 3
Trello Board Process (Outdated) 4
FMOD Audio Setup Process 10
Github 22
Adding New Deliverables 26

Process Documentation Template

Description

One line about what process is being documented

Written By

Who wrote this

Last Updated

When was this updated

Overview

Summary:

This is what process is being documented in more detail. Help to understand what the business process is.

Purpose:

Describe the ideal outcome from following this process

Step 1:

Describe what is happening, why it is happening, and how to do it all.

Trello Board Process (Outdated)

Description

How to use the Trello Board Effectively

Written By

Kyle Gibbs

Last Updated

17/05/2023

Overview

Summary:

This is a brief description of how to use the team Trello Board. This will showcase the process for adding a new item to the Trello board, as well as moving it through all the processes before it makes it to the end.

Purpose:

The ideal outcome would be for the Trello Board to remain tidy while being used, and allow all users to better see how to use it effectively.

Step 1:

Summary:

Start by adding a new item to the Trello Board.

Why?

To allow us to keep track of what work needs to be done.

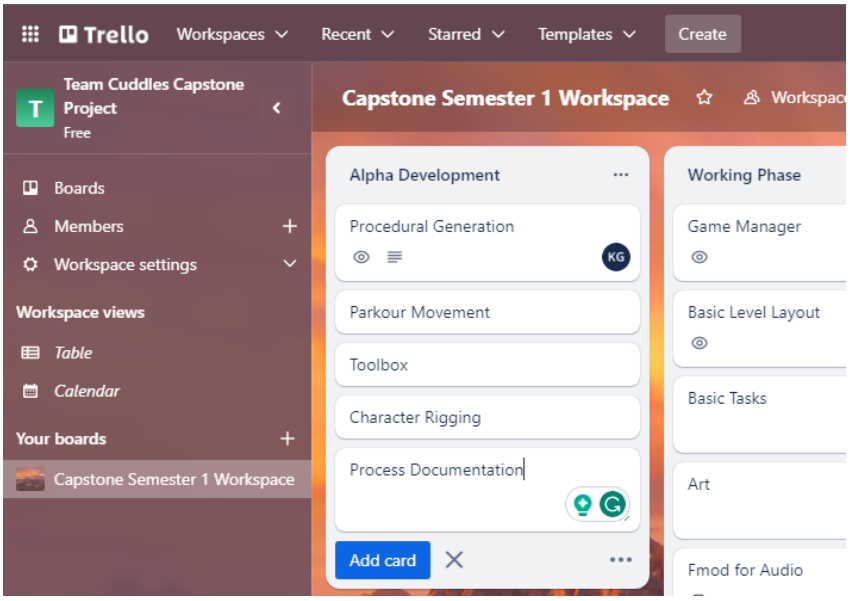
Who Does this Step

Anyone who is looking to add a new item to the Trello Board

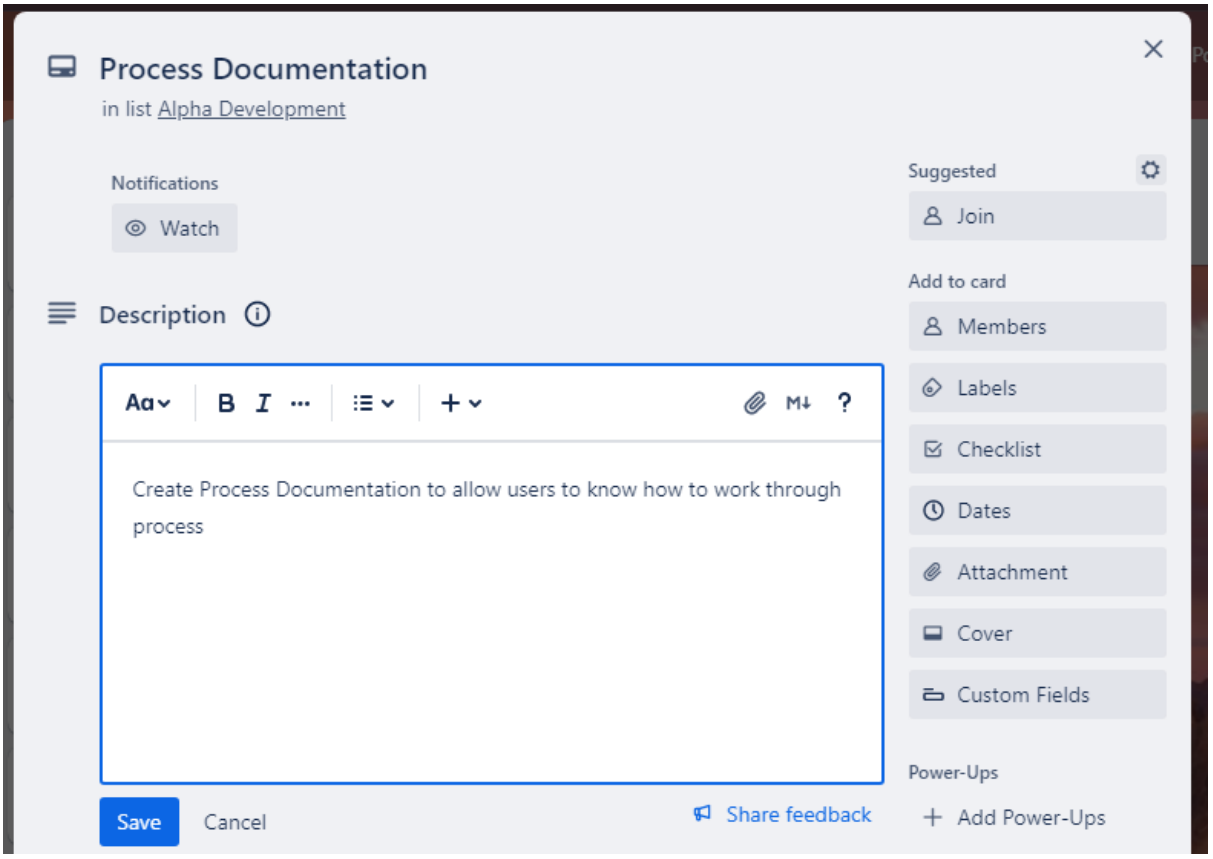
How?

Once logged into the Trello Board (link provided below image), to add a new Trello Board item needing to be completed, you simply click on the “+ Add a card” button in the Alpha Development board. This will add the new card to that section. From there you will add the

title of the card, which should relate to what the task is.



Once the card has been created, you will click on the card and edit the Description to something that will briefly let others know what the task is.



Resources:

Trello Link: https://trello.com/b/BKNBVgYj/capstone-semester-1-workspace

Step 2:

Summary:

Assign yourself to a Trello Task.

Why?

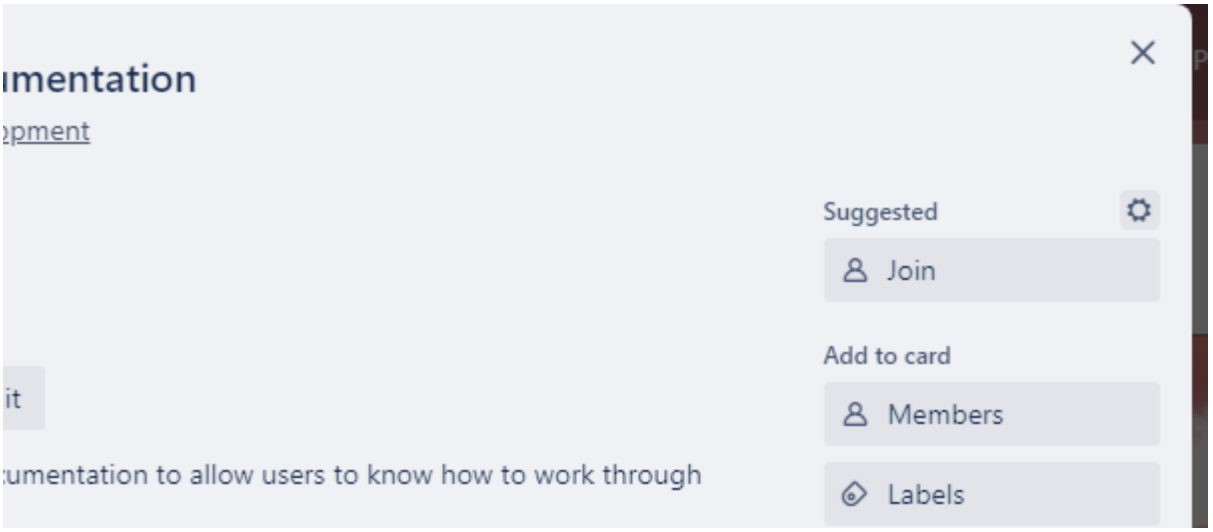
To allow us to keep track of who is completing what work.

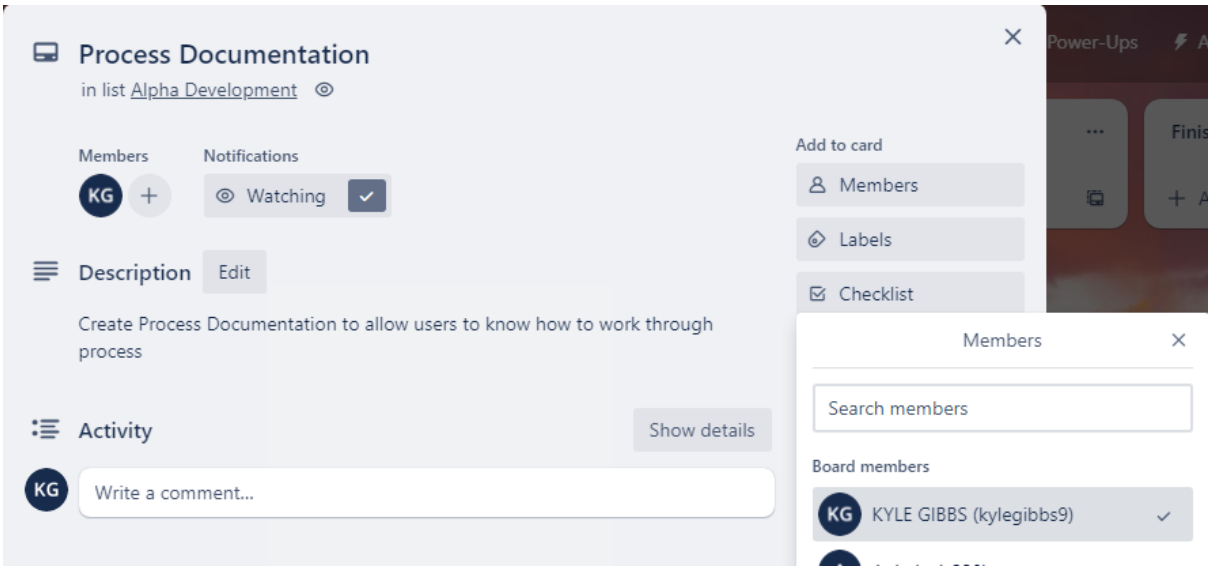
Who Does this Step

Whoever is assigned to complete that task, or whoever decides they want to complete that task.

How?

Once on a chosen Trello card, if you wish to assign yourself to the task you simply click the Members button on the right-hand side, located below the Suggested section. If you are looking to assign yourself, you simply click on your name to add it to the board.





Resources:

Trello Link: https://trello.com/b/BKNBVgYj/capstone-semester-1-workspace

Step 3:

Summary:

Moving a task from Alpha Development to Working Phase.

Why?

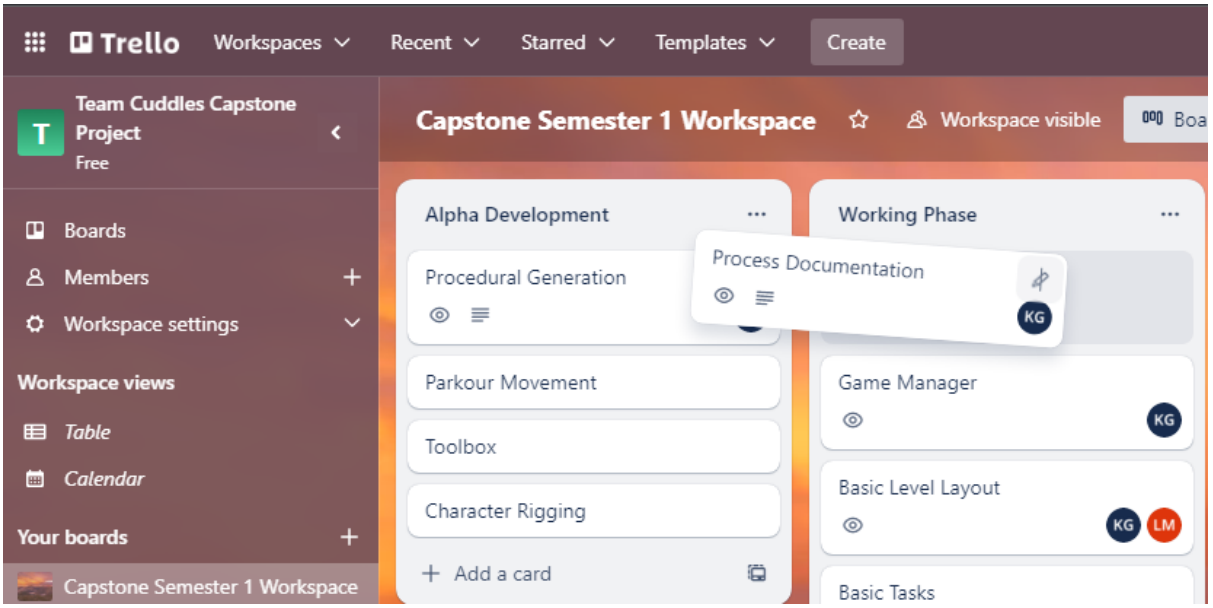
To allow us to keep track of what work is currently being worked on.

Who Does this Step

Whoever is assigned to complete that task.

How?

Once it's decided that a task needs to be worked on, you will drag that task from the Alpha Development section to the Working Phase section. This can be done by holding down the left mouse button and dragging the desired card to the Working Phase section.



Resources:

Trello Link: https://trello.com/b/BKNBVgYj/capstone-semester-1-workspace

Step 4:

Summary:

Moving a task from the Working Phase to the Editing Phase.

Why?

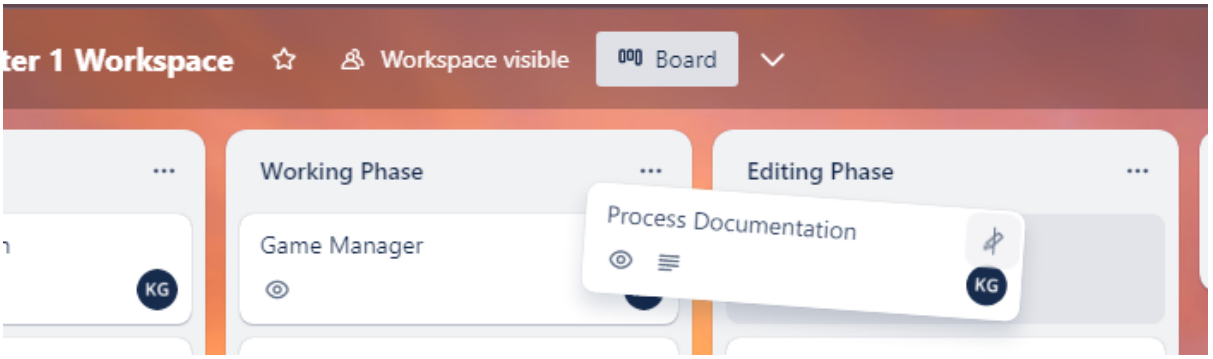
To allow us to keep track of what work is currently left in the working phase and is looking to be approved by the group.

Who Does this Step

The individual who completed the task and is looking for others' approval for it to be considered.

How?

Repeat the same steps as stated in Step 3, except between the Working Phase section and the Editing Phase Section.



Resources:

Trello Link: https://trello.com/b/BKNBVgYj/capstone-semester-1-workspace

Step 5:

Summary:

Moving a task from the Editing Phase to the Finished

Why?

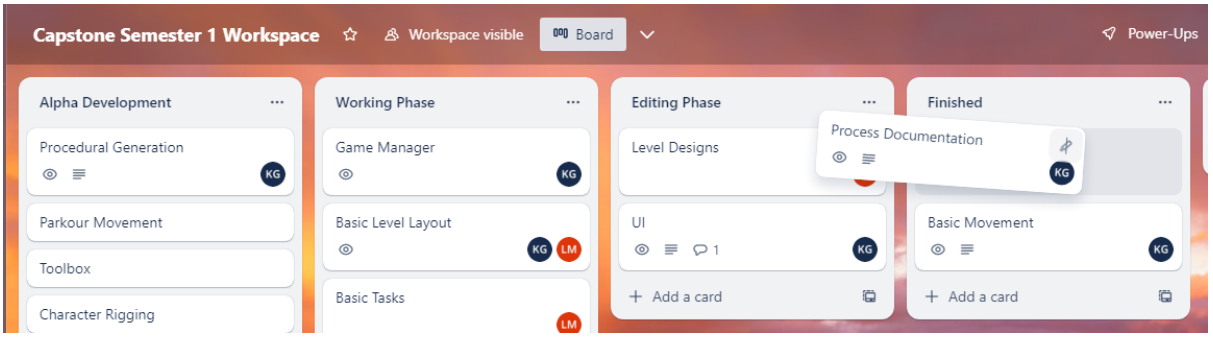
To allow us to keep track of what work is currently completed and no longer needs active changes.

Who Does this Step

The group when it has been decided that that task is completed to a satisfactory level.

How?

After a group meeting where the current state of the task has been discussed, if it has been agreed that the task is completed, the Card will be moved from the Editing Phase to the Finished section.



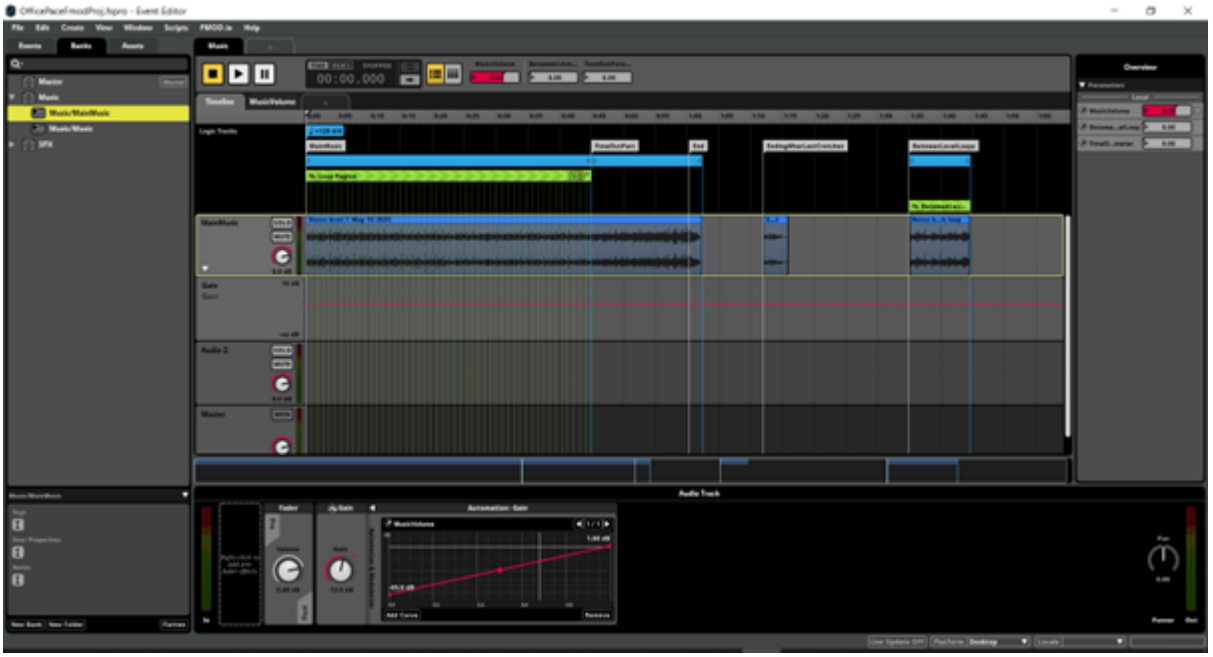
Resources:

Trello Link: https://trello.com/b/BKNBVgYj/capstone-semester-1-workspace
Swapped from GitHub Scenes to GitHub Branches

Keeping track of Tutorial learning

Useful Links for things:
Process documentation Template
https://bit.ai/templates/process-documentation-template#Template-Embed

FMOD Audio Setup Process



Description

How to use
the FMOD studio and control audio through the Unity game engine

Written By

Anis
Setoodeh

Last Updated

25/08/2023

Overview

Summary:

This is a brief description of how to use the FMOD studio. This will showcase the process for adding audio files, as well as modifying them In FMOD studio and Unity engine.

Purpose:

To set up and use Audio/Music, for video games and have great control over different parameters and effects of the audio files.

Step 1:

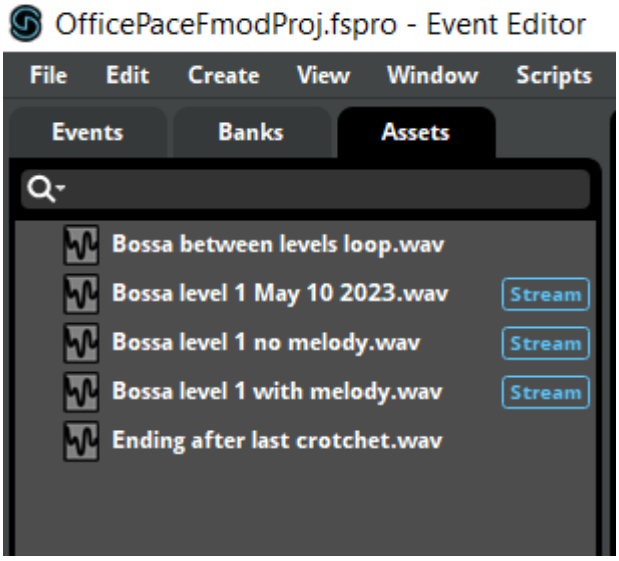
Summary:

Add new Audio track/tracks.

Why? To have audio files for later modifications.

Who Does this Step: The person responsible for audio design and audio setup, to add new Audio files/files to the FMOD assets section.

How? After Installation of the FMOD studio, to add a new Audio track in FMOD studio you just have to drag and drop the track or tracks into the assets folder of the FMOD studio.



Step 2:

Summary:

Create music banks.

Why?

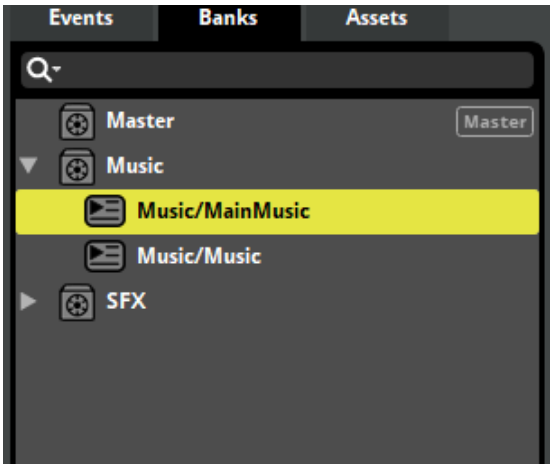
To create a library of sounds.

Who Does this Step:

The person is responsible for audio design and audio setup.

How?

To add a new Audio bank in FMOD studio, select the ban's window and right-click on the page to create a new bank.





Step 3:

Summary:

Set up music events.

Why?

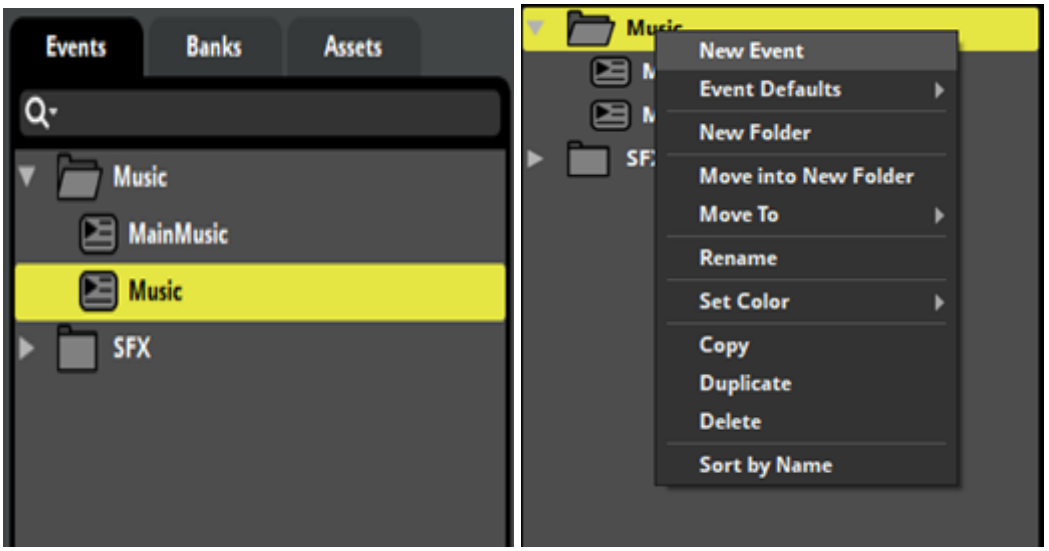
To add new events.

Who Does this Step:

The person is responsible for audio design and audio setup.

How?

To add a new Event in FMOD studio, select the Events tab and right-click on the audio folder to create a new event, to create a new folder repeat the same steps but instead choose a new folder.



Step 4:

Summary:

Add the tracks to the Timeline.

Why?

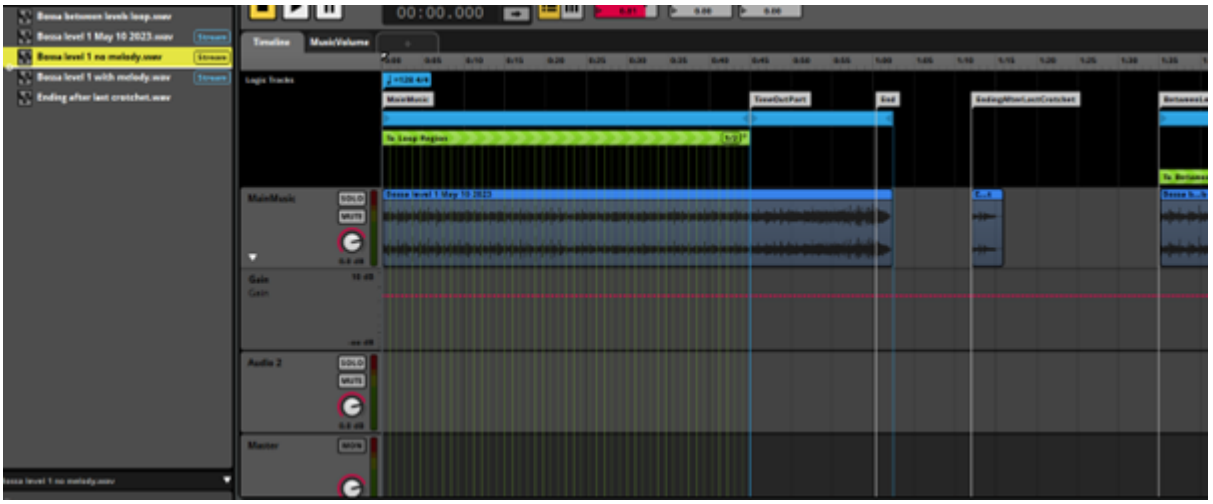
To add audio files into the timeline and prepare them for processing.

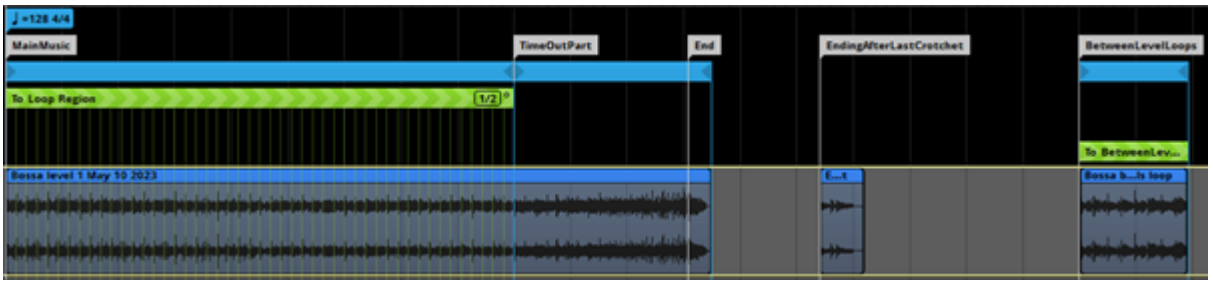
Who Does this Step:

The person is responsible for audio design and audio setup.

How?

To add a new track to the timeline, select the desired track and drag and drop the audio track(s) to the timeline.





Step 5:

Summary:

link the event to their bank.

Why?

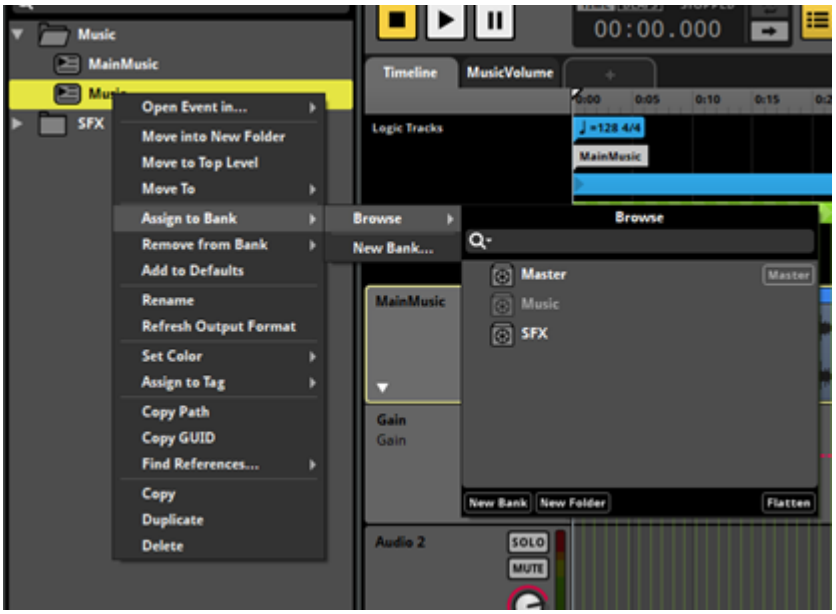
To assign events to their specific banks.

Who Does this Step:

The person is responsible for audio design and audio setup.

How?

To assign an event to their bank, select the desired event and right-click on the event from the menu choose to assign to Bank option and from the browser select the bank you are looking for to link the event to it.



Step 6:

Summary:

Build the project.

Why?

To build the project so Unity can use the elements of the FMOD studio.

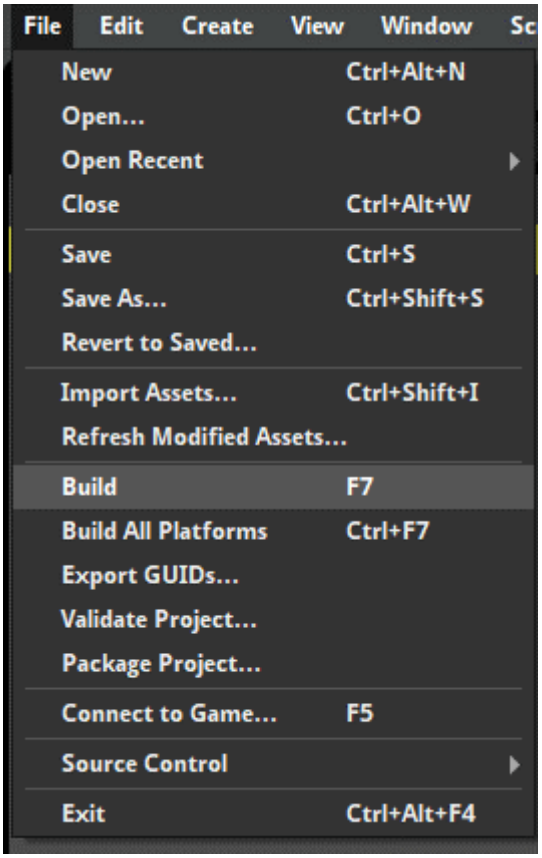
Who Does this Step:

The person is responsible for audio design and audio setup.

How?

To build the project click on the file tab from the top left of the screen and then from the drop-down menu choose the Build option to build the project, alternatively, you can use the hotkey ‘F7’ to build the project.

This is important because by saving the project itself we are not able to see the changes in the unity engine.



Step 7:

Summary:

Add modulation and automation parameters in FMOD and how to use them in unity.

Why?

To add modulation parameters that we can adjust using code to the unity engine for our game.

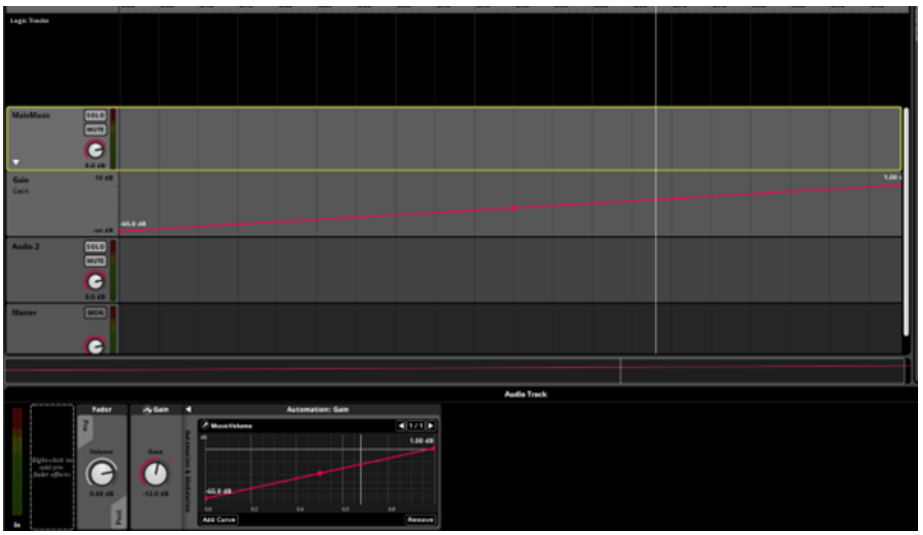
Who Does this Step:

The person who is responsible for audio design and audio setup.

How?

To add Modulation or Automation properties, you should first select the audio track by left-clicking on the desired audio track In the timeline, it will display the master settings for that audio track, by clicking on the window next to the fader you will get an option to Add Effect to your audio, choose Add Effects from the menu and then you can choose your preferred effect, for example you can choose ‘FMOD Gain’ to change the loudness of the track.

These effects can be modified inside the FMOD and from the unity engine using scripts.


Logic Tracks
MainMusic
Audio 2
Master
Audio Track
Fader
Automation: Gain


File Edit Create View Window Scripts FMOD.io Help
Events Banks Assets
Master
Music
Music/MainMusic
Music/Music
SFX
Timeline MusicVolume
Logic Tracks
MainMusic
Audio 2
Master
Fader
Add Effect
Add Send
Add Sidechain
Add Effect Chain
FMOD 3-EQ
FMOD Channel Mix
FMOD Chorus
FMOD Compressor
FMOD Convolution Reverb
FMOD Distortion
FMOD Delay
FMOD Flanger
FMOD Gain
FMOD Limiter
FMOD Multiband EQ
FMOD Panner
FMOD Pitch Shifter
FMOD Reverb
FMOD Transceiver
FMOD Tremolo
FMOD Deprecated
FMOD Spatializer
FMOD Object Spatializer
FMOD Loudness Meter


MusicVolume
0.68
BetweenLeve...
0.00
TimeOutPara...
0.00

Step 8:

Summary:

Play and modify sounds in unity using the script.

Why?

To add scripts that enable us to play the music tracks or sounds when we want, and change parameters related to FMOD in unity such as volume.

Who Does this Step:

The person who is responsible for audio design and audio setup.

How?

To play music we can create a function to play music and call an event reference that holds the reference data to the events inside the FMOD and start that instance by calling the function inside the start method.

We can set parameters such as music volume and sound volume in the script and like how we Implement it for playing music, we can call and adjust those parameters inside functions such as the Update function.

Further, we can create empty objects and assign our scripts to them, so we can play them in our scenes.

```
private EventInstance musicEventInstance;
```

```
1 reference
private void PlayMusic(EventReference musicEventRefrence)
{
    musicEventInstance = CreateEventInstance(musicEventRefrence);
    musicEventInstance.start();
}
```

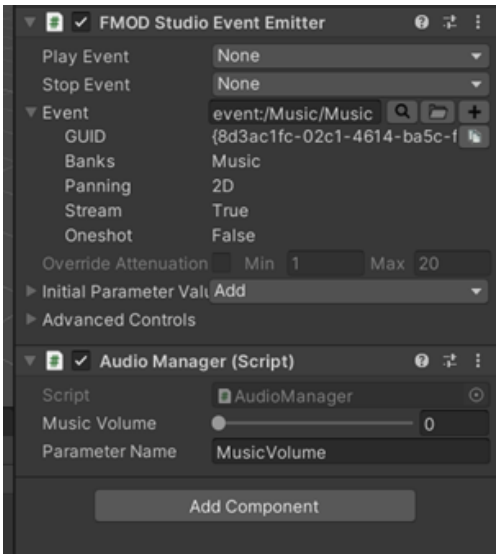
```
Unity Message | 0 references
private void Update()
{
    SetMusicVolume(parameterName, musicVolume);
}
```

```
Unity Message | 0 references
private void Start()
{
    PlayMusic(FMODEvents.instance.mainMusicTrack);
}
```

```
1 reference
public void SetMusicVolume(string parameterName, float parameterValue)
{
    musicEventInstance.setParameterByName(parameterName, parameterValue);
}
```

```
[Range(0f, 1f)]
[SerializeField] private float musicVolume;

[SerializeField] private string parameterName;
```






```
[field: Header("MainMusic")]
1 reference
[field: SerializeField] public EventReference mainMusicTrack { get; private set; }
```

Step 9:

Summary:

Play different sounds in different places.

Why?

To add ways to modify and play sounds in different places

Who Does this Step:

The person who is responsible for audio design and audio setup.

How?

To play different sounds in different places we can create a parameter in FMOD studio, same as mentioned before for the volume parameters and then adjust the value of those parameters through unity using the script, for example, main music can have a parameter that when is set to 1 then plays a different track such as end music.

In unity we can create those parameters and set them to different values 0f for off and 1f to set them off, then whenever the player timer runs out we can set the parameter to 1 and play the end music.

Resources:

1. Fmod Tutorial:

https://www.youtube.com/watch?v=nnT2-ldNEHs

2. Fmod Tutorial:

https://www.youtube.com/watch?v=DI72MHPfh0k&list=LL&index=4

Useful Links for Programming:

Mesh generation(Procedural generation):

https://www.youtube.com/watch?v=yxd8_BHxlmY

Github

Description

How to use Github, how to pull and push branches, create merge requests, and solve conflicts.

Written By

Elias Erek

Last Updated

2/06/2023

Overview

Summary:

GitHub is a tool that allows us to work on the same project but minimises conflicts by allowing us to work in branches, offering version control, as well as other useful tools that are beneficial when multiple members are working on a single repository.

Purpose:

To explain the process of setting up your GitHub repository on your local computer, as well as pushing changes and handling Pull Requests into your main branch.

Step 1: Cloning

Summary:

Download the GitHub Desktop App click “Clone Repository”, and clone the game’s repository.

Why?

To Allow us to work on the repo via the GitHub Desktop application.

Who Does this Step

Anybody who wishes to make changes to the repository.

How?

Click “File” at the top followed by “Clone Repository”. You will be prompted with a window that displays your Github repositories for you (If you are logged in - otherwise, do that first). Click on the repository you want to clone.

Step 2: Pulling

Summary:

Pulling the project to ensure your project is up to date.

Why?

If a team member makes changes to the main repository, the files on your computer become outdated. This can be solved by “Pulling”, which will bring your local files up to date with the repository.

Who Does this Step

Anybody making changes to the repository should Pull it first

How?

Click “Repository” at the top, and then click “Pull”. You may also use the Ctrl+Shift+P shortcut.

Step 3: Branching

Summary:

This step will explain the process of creating your branch.

Why?

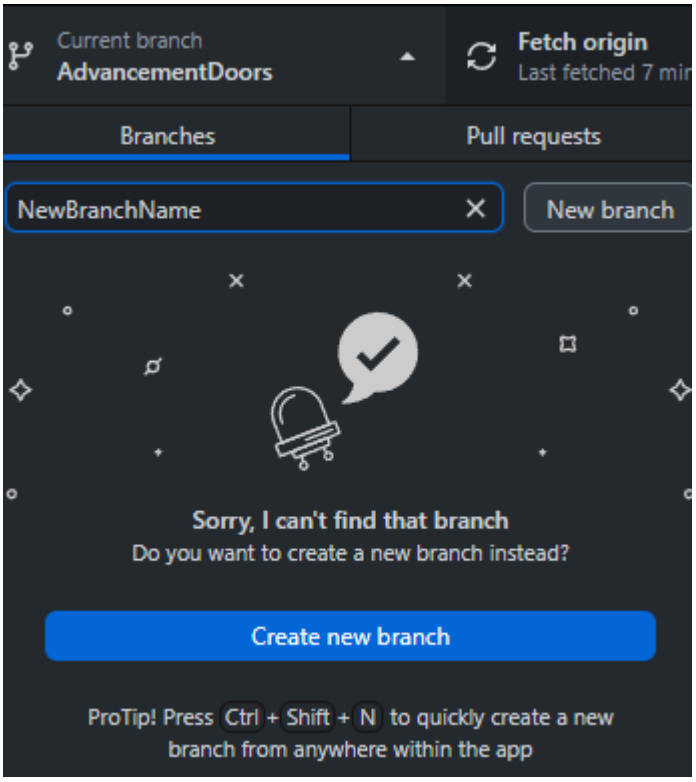
Multiple people working on the main branch of the repository may cause issues such as conflicts. It also ensures that if a change corrupts the project, the corruption is only happening inside a specific branch.

Who Does this Step

Anyone working on the repository should use branches regardless of how small the changes are.

How?

Press “Current Branch” which should be set to main by default, then type in the name of your new branch, and press “Create New Branch”. The “Current Branch” UI should change to show your branch name - This means any changes you make now are made to your branch.



Step 4: Committing and Pushing

Summary:

Pushing your changes from your local files to the repository on GitHub.

Why?

When making changes to your files, the changes are still only local, even if you were working on your branch. You must commit and push your changes to ensure the repository is updated.

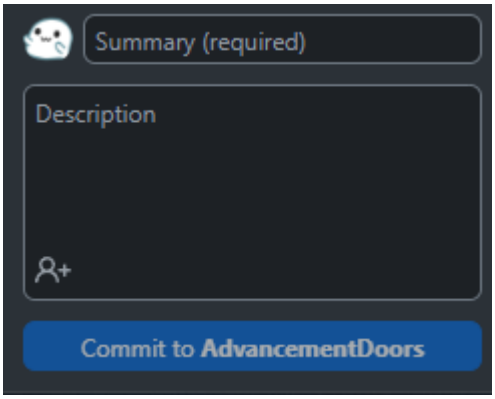
Who Does this Step

Anyone wishing to make changes to the repository.

How?

At the left-hand side of the GitHub Desktop application, there should be a UI that shows all current changes, and on the bottom left, there should be a UI that allows you to enter a name and description for your commit. Once added, you may click “Commit”, and then head to “Repository” in the toolbar above and select “Push” to push your changes to your branch.

Ensure that your branch is selected when committing and pushing! **Remember to only push changes to your branch.**



Step 5: Pull Request

Summary:

Time to merge your branch with the main branch.

Why?

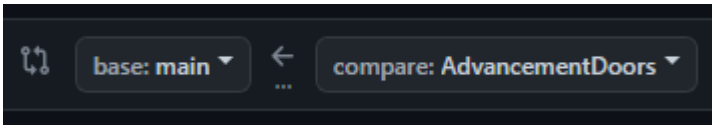
As explained before, the changes were made to your branch, meaning they are only present on that branch. If all is working fine, it's time to push the changes onto the main branch and integrate them into the main part of the repository.

Who Does this Step

Anyone wishing to make changes to the repository.

How?

On the GitHub site, find the repository, and select "Pull Requests" followed by "the New Pull request" button. You will see a UI displaying two branches, a base branch and a compare branch. The base branch is the branch you want to merge your changes into. In most cases it will be the main branch, so we can keep it the same. The UI on the right-hand side (your compare branch) should be the branch you have pushed your changes to. It should look something like this:



After doing so, it will display all the changes you are attempting to merge into the main. When all looks good, create the pull request and ask your team members to review it. Once reviewed, you may open up your Pull Request by going to its link and selecting "Merge and Close", which will merge the changes onto the main and close the Pull Request.

Adding New Deliverables

Description

The necessary parts of adding new Deliverables to unity.

Written By

Kyle Gibbs

Last Updated

07/08/2023

Overview

Summary:

This will list the process of adding a new deliverable task to the unity project. This will allow for automatic generation into the project for testing.

Purpose:

This should allow the existing infrastructure to implement the new deliverable task.

Step 1:

Summary:

Begin programming a new Deliverable based on the design

Why?

To continue expanding our current library of deliverables

Who Does this Step

Anyone programming in a new Deliverable

How?

Program the deliverable according to the design specifications

```
Unity Script (1 asset reference) | 7 references
public class ShredPaper : MonoBehaviour
{
    [SerializeField] bool isPickedUp = false;
    [SerializeField] bool pickedUpCheck = false;
    [SerializeField] GameObject paperStack;
    [SerializeField] Shredder myShredder;
    [SerializeField] public bool allShredded = false;
    [SerializeField] TextMeshPro command;

    [SerializeField] public bool activeTask = true;
    [SerializeField] private bool swapCheck = false;

    [SerializeField] GameObject player;

    // Start is called before the first frame update
    Unity Message | 0 references
    void Start()
```

Step 2:

Summary:

After programming in the base information for the deliverable, code in an “initialState” function. This function will be used to reset the deliverable back to its base values and conditions.

Why?

This is used whenever you start a new run or a new day, making sure that the task can be put back to its default settings

Who Does this Step

Currently, whoever is programming the deliverable.

How?

Create a public initialState() function, and parse in the initial values.

```
2 references
public void initialState()
{
    isPickedUp = false;
    pickedUpCheck = false;
    allShredded = false;

    paperStack.SetActive(true);

    myShredder.resetTask();
}
```

Step 3:

Summary:

If the deliverable has a component carried between destinations, run the newPickup() function from the player class to register that pickup into the hands. The name of the object that the deliverable is being run on needs to be parsed as a variable.

Why?

This is needed to check if another item is currently being carried

Who Does this Step

The programmer, assuming something is being picked up

How?

Get the player object, and run the newPickup() command from it.

```
Unity Message | 0 references
private void OnTriggerStay(Collider other)
{
    if(other.tag == "Player" && Input.GetKey(KeyCode.E) && !isPickedUp)
    {
        isPickedUp = true;

        paperStack.SetActive(false);
        command.enabled = false;

        player.GetComponent<PlayerController>().newPickup("StackofPapers");
    }
}
```

Step 4:

Summary:

If completed step 3, then inside of the player class, add a new case to the switch used in the newPickup function.

Why?

This is needed to drop an item

Who Does this Step

The programmer, assuming something is being picked up

How?

Inside the Player class, find the newPickup() function. From there, add a new case in the style as follows (in this instance, the ShredPaper class is being run on an object named “StackOfPapers”):

case (“StackOfPaper”):
old.GetComponent<ShredPaper>().drop();
break;

```
public void newPickup(string name)
{
    if(!handsOn)
    {
        handsOn = true;
        currentTaskName = name;
    }
    else
    {
        GameObject old = GameObject.Find(currentTaskName);

        if(old == null)
        {
            old = new GameObject();
            old.name = "ManagementThings";
        }

        switch (old.name)
        {
            case ("StackofPapers"):
                old.GetComponent<ShredPaper>().drop();
                break;

            case ("CoffeeTask"):
                old.GetComponent<CoffeeTask>().drop();
                break;

            case ("ManagementThings"):
                Debug.Log("Ignore This for Now");
                break;

            default:
                Debug.Log("No Case Found");
                break;
        }

        currentTaskName = name;
    }
}
```

Step 5:

Summary:

If you did steps 3 & 4, then you also need to program in a drop() function if the item is dropped.

Why?

This allows the player to drop unwanted items, making sure they only carry one at a time

Who Does this Step

The programmer, assuming something is being picked up

How?

Programming in a drop() function in the deliverable. This should either:

- Move the deliverable to the destination the player was in when it was dropped
- If dropping resets the task, return the values to the necessary states (possibly using the initialState() function)

```
1 reference
public void drop()
{
    Debug.Log("Drop Papers");

    paperStack.SetActive(true);
    this.transform.position = new Vector2(player.transform.position.x, player.transform.position.y);

    isPickedUp = false;

}
```

Step 6:

Summary:

Add the deliverable to the DeliverablesManager

Why?

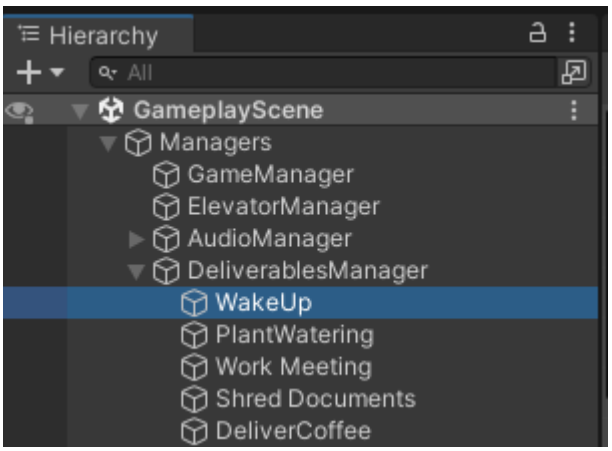
This is needed to allow the program to know what deliverables exist in the scene

Who Does this Step

The programmer

How?

From the unity hierarchy, click on the “Managers” drop-down menu, click on the “DeliverablesManager” drop-down menu, and add a new empty object



Step 7:

Summary:

Add a “Deliverables” script to the empty object

Why?

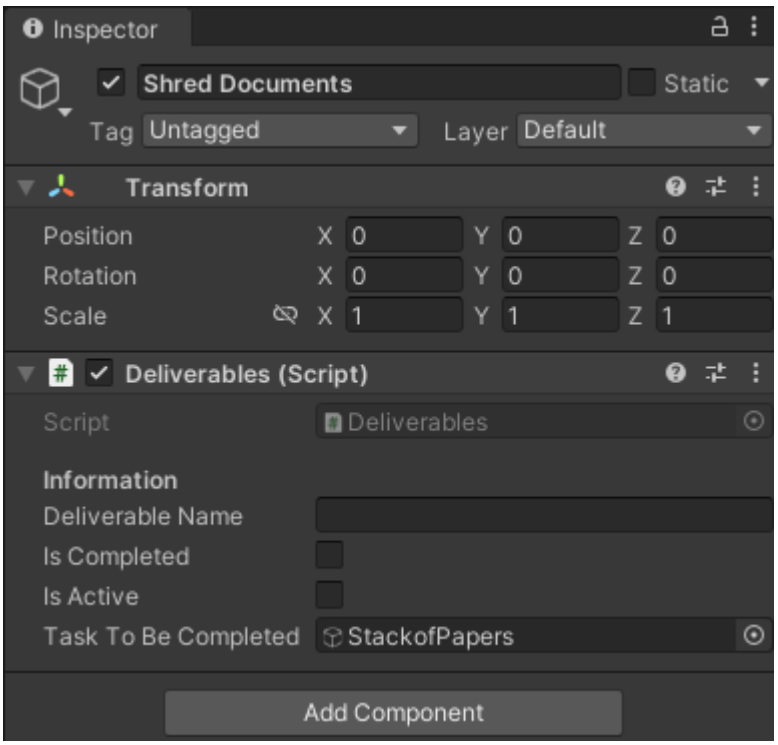
This is needed to allow the object to know what its deliverable is

Who Does this Step

The programmer

How?

Click on the empty object, and in the inspector, click Add component, and add the “Deliverables” script



Step 8:

Summary:

Add the newly created deliverables to the “Task To Be Completed” section

Why?

This is needed to allow the Deliverable Script to know what it is looking for

Who Does this Step

The programmer

How?

Drag the newly made deliverable task to the “Task To Be Completed” box.



Step 9:

Summary:

Inside the TaskController, add a new case to the switch in the AddText function.

Why?

This is needed to allow for text to be displayed based on the needed task

Who Does this Step

The programmer

How?

Inside the AddText() function, create a new case. Its input needs to be the name of the object that holds the deliverable (For ShredPaper, it is “StackOfPapers”). Add in a new string as follows:
tempstring = “Deliverable “ + i + “:
 {This is the section for instructions}”

```
2 references
public string AddText(Deliverables d, int i)
{
    string tempString;

    switch(d.deliverableName)
    {
        case ("LightSwitch"):
            tempString = "Deliverable " + i + ":<br>Wake Up Co-worker";
            break;
        case ("WateringPlants"):
            tempString = "Deliverable " + i + ":<br>Water Office Plants";
            break;
        case ("MeetingRoom"):
            tempString = "Deliverable " + i + ":<br>Attend the Meeting";
            break;
        case ("StackofPapers"):
            tempString = "Deliverable " + i + ":<br>Shred the Bosses Papers";
            break;
        case ("DeliverCoffee"):
            tempString = "Deliverable " + i + ":<br>Deliver coffee to the Boss";
            break;
        default:
            tempString = "Error: Deliverable does not have a Name";
            break;
    }

    return tempString;
```

Adding New Advancements

Description

The necessary parts of adding new Advancements to unity.

Written By

Kyle Gibbs

Last Updated

25/08/2023

Overview

Summary:

This will list the process of adding a new Advancement to the unity project. This will allow for Two-Tier capabilities and access to the upgrade system

Purpose:

This should allow the existing infrastructure to implement the new Advancement

Step 1:

Summary:

Begin programming the functions of the new Advancement based on the design

Why?

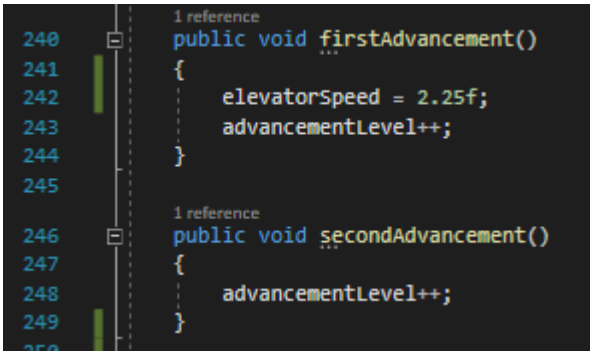
To continue expanding our current library of Advancement

Who Does this Step

Anyone programming in a new Advancement

How?

Program the Advancement according to the design specifications



Example found within the ElavatorController Class

Step 2:

Summary:

After programming the functions of the Advancements, add an integer to keep track of the advancement level

Why?

To track how many times the advancement has been picked

Who Does this Step

Anyone programming in a new Advancement

How?

Create a variable that will hold the advancement level

```
[SerializeField] public int advancementLevel = 0;
```

Step 3:

Summary:

Add the advancement to the advancement manager’s space

Why?

To allow the advancement manager to see the new advancement

Who Does this Step

Anyone programming in a new Advancement

How?

In the unity scene, add a new empty project under the AdvancementManager. Give it a meaningful name, and add the advancement class to it.



Step 4:

Summary:

Add the advancement as a case in the advancement class

Why?

To allow the advancement to know what to do

Who Does this Step

Anyone programming in a new Advancement

How?

Go to the advancement class and add a new case to the switch case found in the enableAdvancement() function. The case variable should be whatever meaningful name you used in Step 3. The case should detail if it needs to perform a function at each advancementLevel, or if it just needs to increase the advancement's level.
Example 1 is an Advancement that requires different functions to run:

```
switch (advancementObject.name)
{
    case ("ElevatorMovement"):
        if (Object.FindObjectOfType<ElevatorController>().advancementLevel == 0)
        {
            Object.FindObjectOfType<ElevatorController>().firstAdvancement();
        }
        else if (Object.FindObjectOfType<ElevatorController>().advancementLevel == 1)
        {
            Object.FindObjectOfType<ElevatorController>().secondAdvancement();
        }
        break;
```

Example 2 is an Advancement that does all the functionality within the class itself

```
case ("Multihand"):
    Object.FindObjectOfType<PlayerController>().advancementMultihandLevel++;
    break;
```

Step 5:

Summary:

Add the text for when the advancement needs to be shown

Why?

To allow the ButtonManager to correctly display text for the Advancement

Who Does this Step

Anyone programming in a new Advancement

How?

Go to the ButtonManager class and scroll down to the setUpText() function. Add a new case to the switch case, using the same name used in Step 3 and Step 4. Use an if and else if statement to show different text based on the level of the advancement.

```
case ("ElevatorMovement"):
    if (Object.FindObjectOfType<ElevatorController>().advancementLevel == 0)
    {
        button.GetComponentInChildren<TextMeshProUGUI>().text = "Elevator is Too Slow<br>Make it Faster";
    }
    else if (Object.FindObjectOfType<ElevatorController>().advancementLevel == 1)
    {
        button.GetComponentInChildren<TextMeshProUGUI>().text = "Elevator is Too Random<br>Make it Always Avaiable";
    }
    break;
```