Close Encounters of the Wool Kind
technical design document

This technical design document (TDD) outlines the technical details of all of the State and Script Machines in the game, making special note to document where custom events exist and when they are referenced during gameplay. This does not need to be read from start to finish, but rather should be used as a reference point as you work on your port. In Unity, all Script Graphs have descriptive summaries, and where needed, are grouped and commented to clarify what each section does. Select each group to see notes about the functionality in the Graph Inspector.

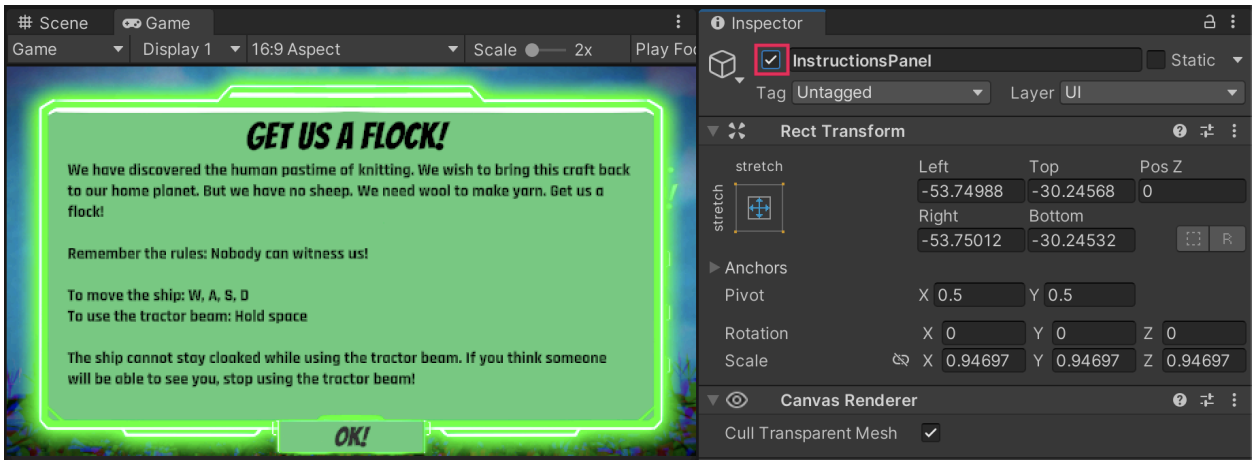
StartScene

The start scene is the first scene that the user experiences when they launch the game. There are three buttons in the scene that include most of the functionality:

- **Start!**: This button triggers a UFO flying animation, fades to black, and then loads the Main Scene.
- **How to Play**: This button activates an instructions panel that sits on top of the scene contents.
- **Quit**: This button will quit the application, or if it is running inside the Unity Editor, will log a message to the console.

StartScene: The InstructionsPanel GameObject

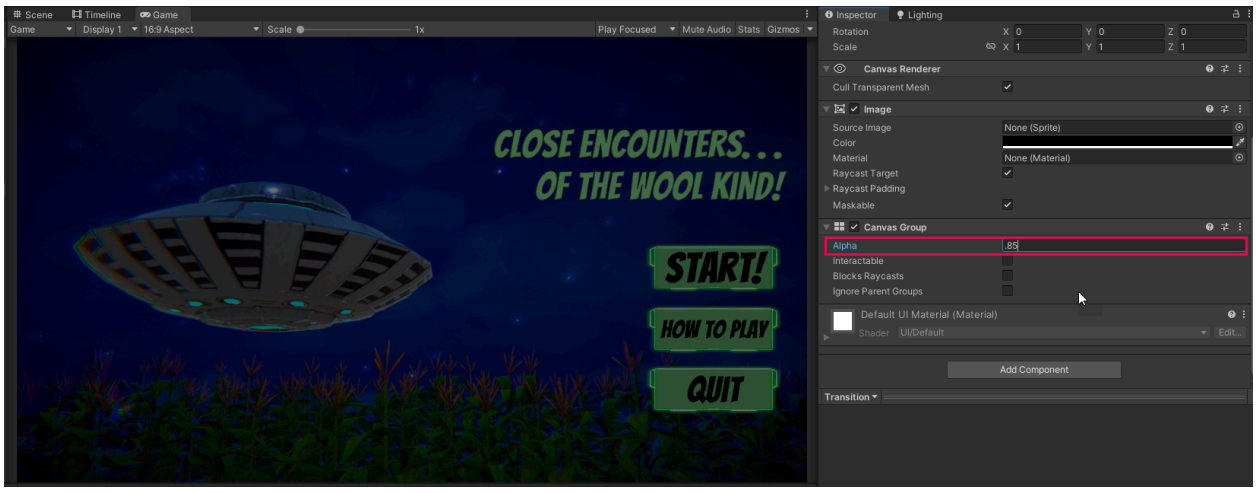
The **InstructionsPanel** GameObject is activated when the user selects the **How To Play** button.
The **InstructionsPanel** GameObject is a child of the **Canvas** GameObject and is inactive by default.



This functionality is controlled in the **How to Play button** group of the **StartMenu** graph, which can be found on the **SceneManager** GameObject.

StartScene: The Transition GameObject

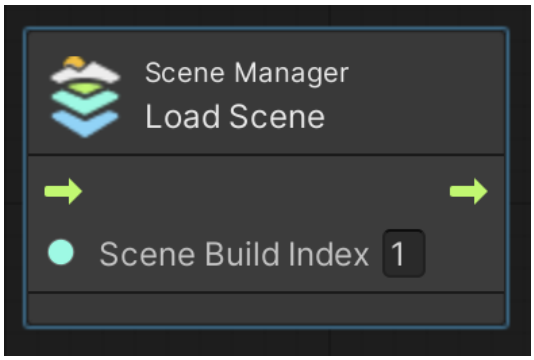
The **Transition** GameObject is used to fade the screen to black when the **Start** button is selected. The **Transition** GameObject is a child of the **Canvas** GameObject.
In the Canvas Group component, if the **Alpha** value changes from **0** to **1** and back to **0** again, the screen fades to and from black.



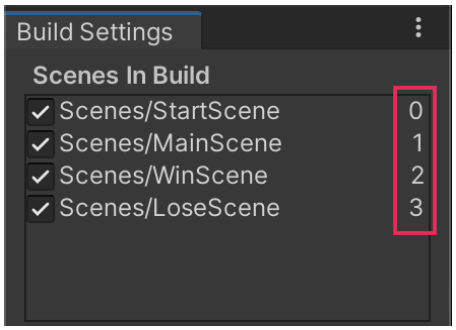
All fades to and from black in the game are handled this same way. For a fade to black, the Alpha value transitions from 0 to 1. For a fade from black, the Alpha value transitions from 1 to 0. In the Start Scene, this transition is controlled in the **StartMenu** graph, which can be found on the **SceneManager** GameObject.

StartScene: The scene build indexes

When the user selects the Start button, the MainScene loads. In this game, scenes are loaded according to their **build index**, which is an integer corresponding to the scene's rank in a list. The first scene in the list has an index of 0, the second scene has an index of 1, and so on. In the Script Graph, the **Scene Manager: Load Scene** node can load the scene with a given index.



In the Build Settings window (from the main menu, select **File** > **Build Settings**), all four scenes are listed in the **Scenes in Build** panel, each with a unique build index value.

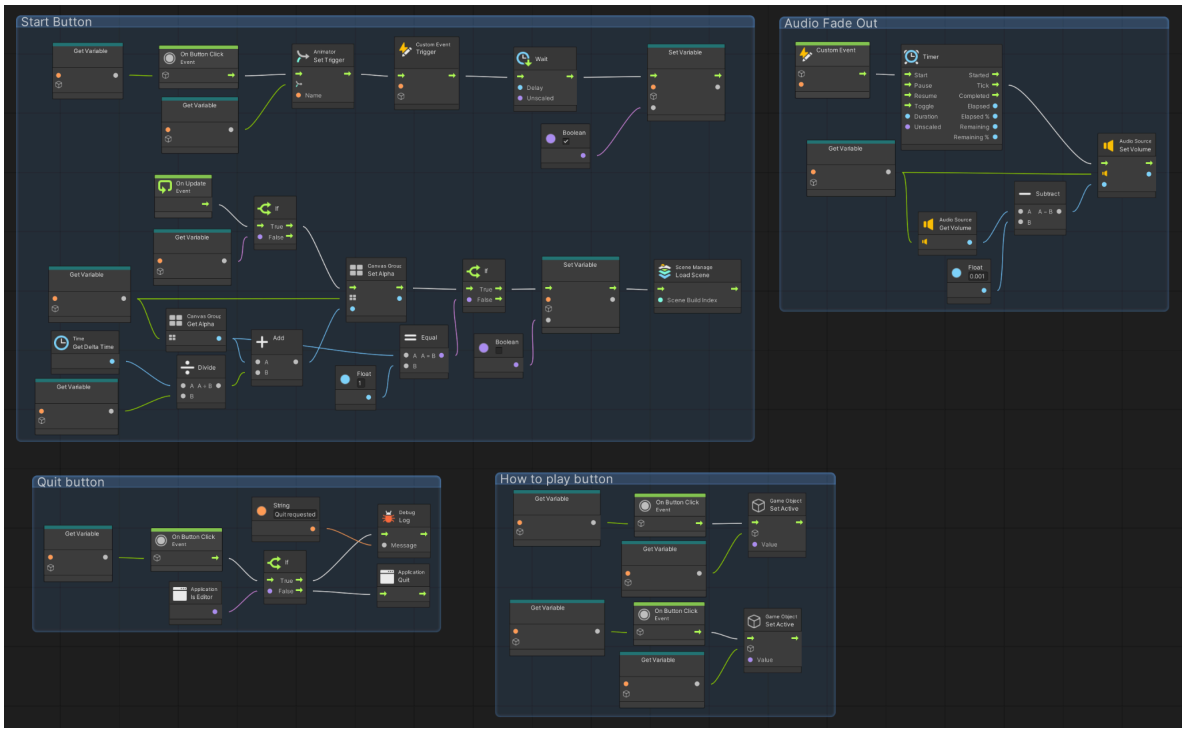


StartScene has an index of 0, MainScene's index is 1, WinScene's index is 2, and LoseScene's index is 3.

Loading the MainScene from the StartScene is done in the **StartMenu** graph, which can be found on the **SceneManager** GameObject.

StartScene: The Start Menu Script Graph

There is only one script machine in this scene that controls all of its functionality: the Start Menu graph. It is located on the **SceneManager** GameObject.
Each button's functionality is organized into a group, with a comment explaining the node sequences.



Win and Lose Scenes

The win scene and the lose scene are the simplest scenes and include almost identical functionality.

The following two buttons exist in both scenes:

- **Restart**: This button loads the Main Scene again.
- **Quit**: This button will quit the application, or if it is running inside the Unity Editor, will log a message to the console.

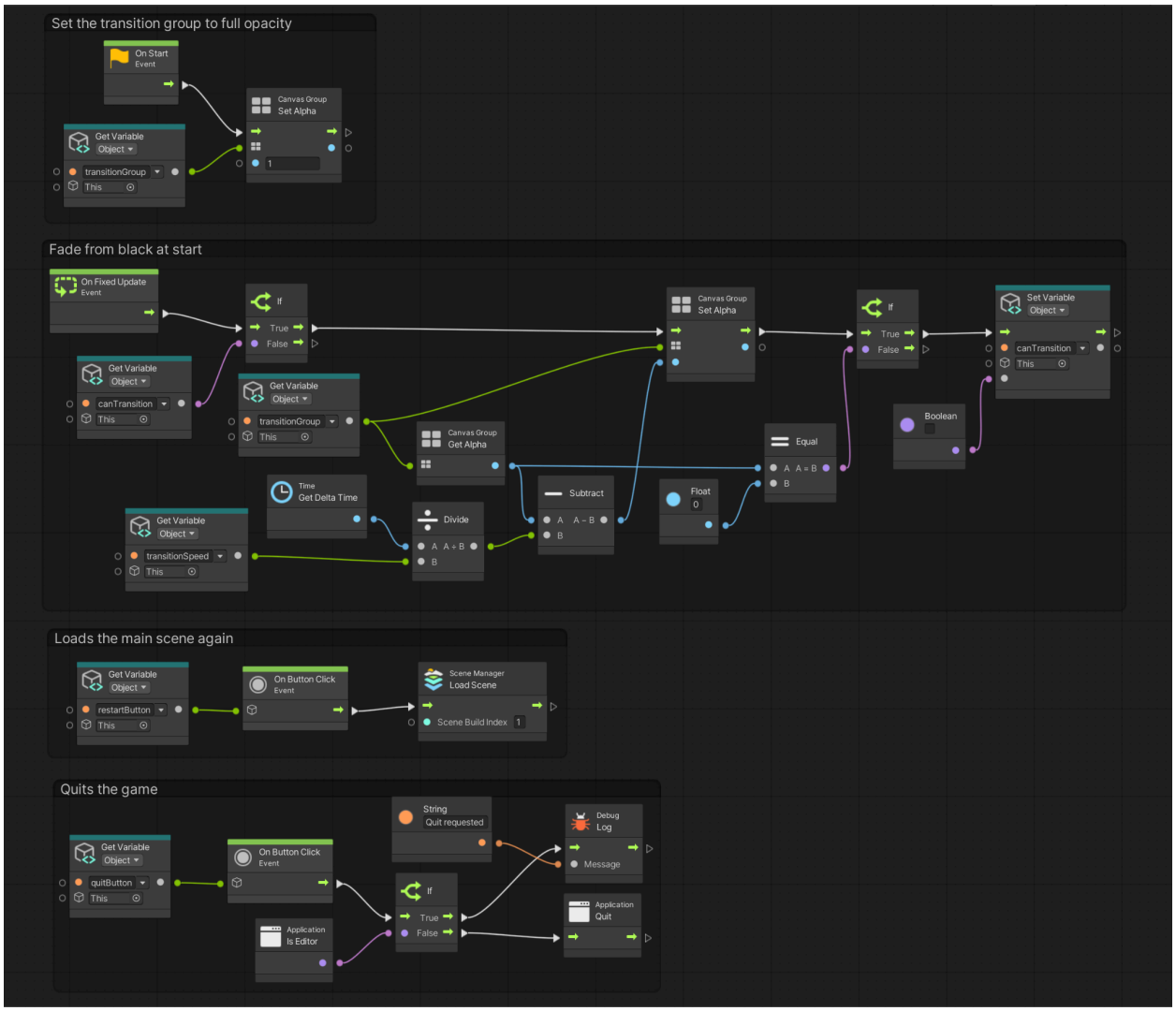
The sections below will highlight some of the interesting, relevant, or complex features in the win and lose scenes.

The Game Over manager

Since the two buttons on the win and lose screens have identical functionality, both scenes share the same **GameOverManager** Script Graph. This graph fades the scene in from black, then manages the Restart and the Quit buttons.

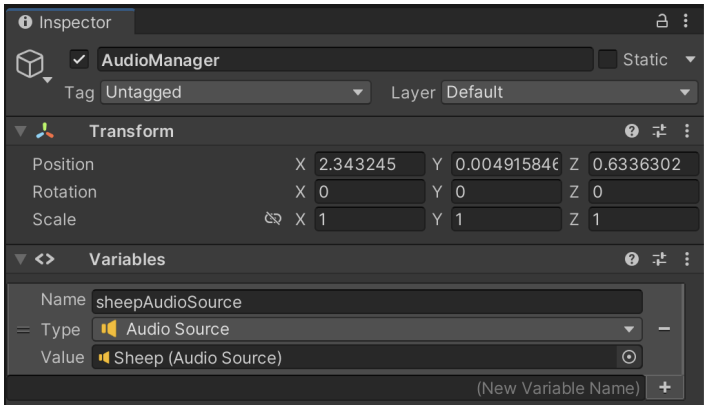
The **GameOverManager** Script Graph is located on the **EndSceneCanvas** GameObject. It has four node sequences, each of which is grouped and commented:

- Set the transition group to full opacity
- Fade from black at start
- Load the main scene again
- Quit the game



Lose scene audio manager

In the LoseScene, a sheep sound plays at random intervals. The **EndSceneAudio** script graph is located on the **AudioManager** GameObject. The sheepAudioSource variable holds the specific sheep sound effect, which can be replaced with other sounds.



In the **EndSceneAudio** script graph, the **Random: Range** node controls the minimum and maximum time between sheep sounds.

Win scene animation manager

In the WinScene, sheep float up and down and orbit around the UFO. For a more dramatic effect, the sheep are also using their abduction animation.

The **WinAnimationManager** graph is located on the **SheepGrp** GameObject (short for “Sheep Group”).

Main Scene

Game Flow

Refer to this section for a high level overview of what happens in the MainScene as soon as it loads in game.

- Scene loads
- Game Manager sets all of the required Scene variables and Object variables
- Game Manager starts the opening cutscene
    - Game Canvas Transition object fades from black to transparent
    - Cutscene UFO animates into the scene
    - Cutscene UFO is disabled, the Player UFO becomes enabled
    - GameStart event is Triggered to activate the Farmer
- Game is active, user can now control UFO
- Farmer cycles from asleep to waking, to awake based on random interval timers
- Sheep cycle through their idle animation. Each sheep’s animation has been offset randomly.
- User uses W, A, S, D to move the UFO within the bounds of the farm
    - If the UFO moves over a sheep, the sheep is assigned to the Target variable

on the UFO
    - If the UFO moves away from any sheep assigned to the Target variable, the Target variable is set to null
- User presses Space to activate the tractor beam
    - UFO material changes from the transparent cloaked material to the opaque material
    - The Visible boolean variable on the UFO is set to true
    - The tractor beam game object renderer is enabled
    - IF a sheep is assigned to the Target variable, it will begin to ascend toward the UFO
        - IF the tractor beam remains active until the sheep enters the UFO, the sheep will disappear and the sheep counter UI will be reduced by one.
            - If the sheep counter UI is reduced to 0, the WinScene loads and the game is won.
    - IF the farmer awakens, the game will end
- User releases Space to deactivate the tractor beam
    - UFO material changes from the opaque material to the transparent cloaked material
    - The Visible boolean variable on the UFO is set to false
    - The tractor beam game object renderer is disabled
    - IF a sheep is assigned to the Target variable, it will drop back to the ground
    - IF the farmer awakens, the game will continue

GameObjects in the Hierarchy

All of the objects listed below are those that exist in the MainScene Hierarchy and contain Script or State Machines.

VisualScripting SceneVariables

Variables on this GameObject can be accessed by any script graph in the scene. There are five total variables, three of which are assigned to by the Game Manager on Enable:

- **_GameManager:** Has the Game Manager GameObject already assigned. The script machines on this GameObject run the high level aspects of the game.
- **_FarmerManager:** Assigned by the Game Manager at the start of the game. The GameObject assigned is the Farmer in the Hierarchy. The state and script machines on this game object manage the Farmer’s behavior.

- **_UIManager:** Assigned by the Game Manager at the start of the game. The GameObject assigned is the Game Canvas in the hierarchy. The script machine on this object manages the UI counter.
- **_UFOManager:** Assigned by the Game Manager at the start of the game. The GameObject assigned is the UFO in the Hierarchy. This is the player.
- **globalVolume:** Has the Global Volume object already assigned. This is the post processing volume that is enabled when the UFO is abducting a sheep.

Game Canvas

Manages the sheep counter that appears in the upper left part of the screen during gameplay. Has one script machine named InGameUI. This script machine has a custom event called **UpdateSheepCount** that takes one argument. When called, the event will update the Textmesh pro text to read “Sheep remaining: [updated number sent via the event argument]” **UpdateSheepCount** is called by **Sheep Counter** on the Game Manager.

Player UFO

The UFO contains four Script Machines that manage the tractor beam, the UFO’s movement, its visibility, and its audio effects. Throughout these Script Machines, the UFO makes heavy use of calling custom events of other Script and State Machines throughout the game. Each Script Machine includes grouped sections that are fully documented for your exploration. High level notes on important features in each Script Machine are referenced below:

- **Manage Beam:** Has four grouped functionalities: detecting the space bar being pressed and released, detecting if an object within the beam’s trigger collider is a Sheep, and enabling and disabling the beam. Contains two custom events, **UseBeam** and **StopBeam,** which are called within this Script Graph.
- **UFO Movement:** Handles the movement of the UFO using W, A, S, D. Each of the four graphs in this Script Machine function the same, with only the key press, movement direction, and limits customized. The limits are set to the size of the farm.
- **UFO Visibility:** Sets the UFO back to its cloaked material, as well as disables the beam. This graph has a custom event called **GotTarget,** which is called by the sheep when they’re abducted. This graph is also called when **StopBeam** is called, and calls the **CallAudio** custom event on the UFO Audio Manager.
- **UFO Audio Manager:** Manages the audio effects for the UFO. Has one custom event named **CallAudio** which takes one argument to define which audio clip should be played.

Sheep

Each sheep in the scene have one state machine and one script machine to manage their behavior states and sounds:

- **Sheep States:** This state machine handles the different behaviors that the sheep have. These are Start, Idle, BeamedUp, Released, and Abducted. To see detailed descriptions of what each state does and how they interact with each other, select the Edit Graph button on the Sheep States state machine. Double click to enter each state and refer to the group names and descriptions in the graph inspector. High level notes on each behavior and where their custom events are called are noted below:
    - **Start**: Allows the sheep to become active in the game. In the transition between the Start and Idle state, each sheep instance is assigned a random value to a parameter in their animator. This value offsets each sheep's starting animation, allowing them to all look uniquely animated even though they all share the same animation sets.
    - **Idle**: This is an empty state. When in this state, the sheep simply perform the idle animation. This state is transitioned into via the Start state as well as the Released state via the **Idle** custom event, which is called by the Released state in this State Machine.
    - **BeamedUp**: Manages the physical movement of the sheep as they are abducted, along with the audio and animation changes. This state is transitioned into via the **BeamedUp** custom event, which is called by the Manage Beam Script Machine on the UFO. If the sheep abduction is completed, this state calls the **GotTarget** custom event on the UFO visibility Script Machine.
    - **Released**: Returns the sheep to their original location if abduction is not completed and calls related audio and animations. This state is transitioned into via the **Released** custom event, which is called by the Manage Beam Script Machine on the UFO.
    - **Abducted**: This destroys the selected sheep game object and triggers **UpdateCount** on the GameManager. This state is transitioned into via the **Abducted** custom event, which is called by the BeamedUp state in this State Machine.
- **Sheep Audio Manager**: Plays a sheep bleat sound effect at random intervals once the game starts.

Farmer

The farmer functions in much the same way as the sheep, with one State Machine and one Script Machine to handle behavior states and sound.

- **Farmer States:** This state machine handles the farmer's three states: Sleeping, WakingUp, and Awake. To see detailed descriptions of what each state does and how they interact with each other, select the Edit Graph button on the Farmer States state machine. Double click to enter each state and refer to the group names and descriptions in the graph inspector. High level notes on each behavior and where their custom events are called are noted below:
    - **Start**: Enables the worldspace canvas over the farmer's head at the start of the game. The transition to the Sleeping state is handled by a custom event called **GameStart** which is called by the Cutscene Manager script machine on the Game Manager.
    - **Sleeping:** Manages the sleeping and waking cycle along with associated animations. It also contains the graph that keeps the worldspace UI above the farmer's head facing the camera. A random timer is generated when this state is entered, and at the end of the timer, it triggers the **Waking** custom event on this State Machine.
    - **WakingUp:** Manages the sequence between the farmer sleeping and becoming alert. Calls the **SetAudio** event on the Audio Manager, then transitions the animation, disables the worldspace UI, then triggers the **Awake** custom event on this State Machine.
    - **Awake:** Checks to see if the UFO is currently visible in the scene. If the UFO is not visible, then it will wait for a random amount of time before calling the **Sleeping** custom event to transition to the **Sleeping** state in this State Machine. If the UFO is visible, this state calls the **SetAudio** custom event on the Audio Manager, then calls the **GameOver** custom event on the GameManager.

- **Audio Manager:** Contains a custom event called **SetAudio,** which takes two parameters. The first parameter is the audio clip to play, and the second is a bool, that if set to true, will loop the audio clip until the event is called again.

Game Manager

Responsible for the startup and management of the major game states. The Game Manager has Four script machines:

- **Scene Setup:** Finds and loads object references needed to start the game. This is the script machine that assigns objects to the VisualScripting SceneVariables object, in addition to adding object references for the Game Manager itself. This removes the need to manually add object variables to the individual graphs.

- **Cutscene Manager:** Runs the animation of the UFO flying to the farm and then starts the game.
- **Game Over**: Manages the screen fade out and loading of either the game win or game lose scenes. Contains a custom event called **GameOver** which takes one boolean argument. When the bool is set to true, the player has won, when the bool is false, the player has lost. **GameOver** is called by **Sheep Counter** (the final script machine on this game object)
- **Sheep Counter:** At the start of the game, searches the Hierarchy for all objects tagged as Sheep. Sends the total count to the **UpdateSheepCount** custom event on the game canvas. Has a custom event called **UpdateCount** that subtracts 1 from the sheep count when called by the Abducted state. If the sheep count is equal to 0, then it calls the **GameOver** custom event.