Unity Tutorial Report

Amaryllis Argueta
aargueta@oxy.edu
Occidental College

1 Introduction

Since I am working in Unity and writing code in C#, a language I had never used before beginning my senior comprehensive project, I have had to look through and follow a lot of tutorials for various aspects of my project. I spent time looking at tutorials for the Unity editor itself but I also had to look through tutorials for learning and writing scripts in C# to make my program actually interactive.

In Unity, scripts control all the functionality of a project. Scripts can be used to control character movements, take user input, and control/change levels and scenes. Therefore, given my interest in UX/UI and the importance of user experience in determining the success of this project, I chose to focus on UI scripting and scene changing in Unity for my tutorial report.[2]

2 Methods

The goal of the tutorial[3] I followed was to teach users how to write scripts to change scenes in a game or project. This was done through implementing UI elements and canvases to create a user interface that users can interact with and have it go to another scene. In this particular tutorial, the creator of the video demonstrates this function by creating a main menu screen and a main menu button to switch between scenes. I enjoyed the way this function added a sort of linear nature to a project and the ability to click back and forth between the main menu and the game scene reminded me of being able to flip through a physical document. I felt as though this feature would be reminiscent of going through the pages of the disorientation guide and knew that I wanted to incorporate this into my project.

Following along with the tutorial, I began by creating a canvas to house UI elements. The creator of the tutorial already had a user interface set up, but through previous tutorials I had looked at I was able to build and setup my canvas before beginning to implement the scripting components. After creating the canvas and buttons for my home screen, I created a new scene with a canvas to have multiple scenes to switch between/manage.

After setting up my scenes layout, I followed along with the tutorial and added the scenes to the editor build settings and configured my settings so that I could access and change these scenes from within my script. Once I had configured the editor, I was able to start my first script, the ”ScenesManager” script. The creator of the tutorial indicates that it is important to distinguish between ScenesManager and SceneManager because SceneManager is actually used as part of the Unity.SceneManagement library. In the script, an Scene enum is declared and in the method the scenes from the project are included, so that they may be called upon from different methods and scripts. I then created four methods:

LoadScene(Scene scene);
LoadNewGame();
LoadNextScene();
LoadMainMenu();

In each of these methods, I used the SceneManager.LoadScene to load the corresponding scene. Once I wrote these methods, I created a gameobject called ScenesManager (the same as my script name) in both of the scenes I was using to house the corresponding script. By adding this gameobject to both scenes, I was able to then attach the ScenesManager script to a gameobject and actually have it be read by the engine. Once I had created my ScenesManager script and applied it to my project, I was able to move onto the UIMainMenu and UIGameMenu scripts.

The tutorial for the UIMainMenu and UIGameMenu scripts were essentially the same, just with some slight changes in the names of the buttons being used. For the UIMainMenu script, I followed the tutorial to declare a button object within the script that will be used to make the button in our canvas functional. After initializing the variable, I wrote a line of code to add a listener that would load the main menu when the button was clicked. In order for this code inside of the start function to work, I needed to create another function called StartGame() that called on the LoadNewGame function from the ScenesManager class.

In order to write the UIGameMenu script, I copied the code from the UIMainMenu class - as the creator of the tutorial instructed - and changed the name of the button as well as the function that was being called on button-click. Once I created both of these scripts, I placed them into the corresponding scenes on the canvases I created and then configured the settings so that the UI buttons would be active. Once I had followed this last portion of the tutorial, I had a working user interface where I could click back and forth between my main menu and my project screen.

3 Evaluation Metrics

I think this tutorial was quite useful in learning how to create working UI and how to change between scenes which is an important part of any Unity project. As I mentioned, once I had finished the tutorial, I had working UI elements in my canvas and while I did not have much added to the scene besides the buttons and some text, I did have a working project that successfully implemented the scripts I had created.

When thinking about evaluation metrics in regard to a tutorial, I think a way to evaluate whether or not a tutorial is useful and successful is reliant on three things. First, the clarity in a tutorial and how the creator is able to explain certain concepts. Second, user feedback, which in this case is my feedback as well as some comments on the YouTube video of the tutorial. Finally, whether or not the tutorial - when being followed exactly - produces a successful outcome and all of the code runs the way it is meant to.

Given these evaluation metrics I have established for the tutorial, I believe that this tutorial was both useful and successful. The video itself was only about 23 minutes long, and though I had to pause and rewind a few times to make sure I wrote the code correctly, it was very clear and easy to follow along with. The creator, Dani Krossing, also did a good job of providing further explanation on some aspects of the code. For example, indicating how to load the next scene from the current one even though he was only using two scenes. Which would be useful for applying this code to projects with many scenes. I think this also shows how Krossing did a good job of presenting the code in a way that can be built upon and altered based on the different needs of different projects. Based on the comments on the tutorial video, many of which are positive, user feedback would indicate that other people were also able to successfully follow this tutorial, with only a handful of people commenting that they ran into errors when writing their code. Despite this - given the last evaluation metric I defined - I was able to successfully follow the tutorial and produce the desired outcome from the video.

With working scripts and buttons, I am not only able to use that in my senior comps projects, but I can easily transfer this code and use it as a template to control the UI of other potential projects. It is also important to note Krossing created this tutorial on a 2D game, whereas my project will be in 3D. Despite this, the code worked on his project and mine, and I think that also speaks to the success of the universality of this code.

4 Results and Discussion

Though I spoke on this in the previous section, I believe that this tutorial was very useful and successful in the way Krossing displayed the information to follow. I have gone through many Unity tutorials in the process of creating my Comps project and though many of them have been helpful, tutorials that use video to show what is happening on the screen in real-time are, in my opinion, some of the best ways to learn about Unity specifically. Since there are a lot of moving parts when testing a game, it is nice to be able to visually see a video of what should be happening instead of just reading it or seeing screenshots.

I also think a big part of what made this tutorial successful was the ability to use this code as a template for other projects. Though I only created two simple scenes to test this code, I can easily see myself implementing these scripts in my Comps project. I also think this feature will be useful when doing user testing. By creating an interactive start screen, and having code that will allow me to move linearly from one scene to the next, it will allow me to streamline the large amount of information that I am using from the Disorientation guide. This will allow me to mimic users going from one page to the next and will allow me to display certain information on the title screen and other canvases before the user enters the virtual space.

One critique I have for the tutorial is that some of the decisions in naming certain objects and files were slightly confusing. I think this is also probably what caused errors in the code of people commenting on the video that they ran into problems with the tutorial. With such similar names for methods and classes (ie; having to distinguish between SceneManager and ScenesManager) it was tricky to distinguish between which name to use when going through the tutorial the first time. Despite this, and despite the fact that this initially led to a few errors, I was able to go back in the video and clarify which variable to use. But I do believe it would just be easier to alter the names so that they are more easily distinguishable.

Overall, I successfully followed this tutorial and was able to elevate the level of interactivity of my project which is one of the main goals for my Comps - to present the disorientation guide in a more interactive way. [1]

References

[1] Argueta, Amaryllis D. *Amxryllis/tutorialreport: Comp390 Tutorial report repo*. May 2023. URL: `https://github.com/amxryllis/TutorialReport`.

[2] Argueta, Amaryllis D. *Overleaf link to view this tutorial report*. May 2023. URL: `https://www.overleaf.com/read/qdfgtyqtjvnm`.

[3] *CHANGE SCENE WITH BUTTON IN UNITY — Scene Manager in Unity — Learn Unity*. YouTube, Oct. 2022. URL: `https://www.youtube.com/watch?v=jrPTpD2eAMw`.