Artifact One Narrative

Daniel Dennington

Southern New Hampshire University

CS - 499

The enhanced artifact is a simple game made a few days ago using Unreal Engine, the C++ programming language, and Unreal’s Blueprint visual scripting system. In the game there are two characters and a simple object for the player to interact with. I have implemented the ability to interact with these characters within the Engine’s editor, along the ability to move the player’s character using C++ . There’s also an item for the player to interact with and pick up and these goals were met using the Blueprint system.

There are many reasons I chose to include this project in my ePortfolio. For one, this artifact enhancement is a simple video game, and one of my interests as a software developer is video game development. I plan on releasing a video game onto a popular indie gaming platform, such as itch.io or perhaps even Steam sometime in the near future. So, in turn this enhancement will serve as great learning experience to help me realize my future goal.

Another reason for this inclusion, is that developing a video game allows me to better understand OOP principles such as using classes to take advantage of polymorphism and inheritance to work on an object-based piece of software more effectively- be it a video game or something else entirely. Understanding these facets of Object-Oriented Programming are essential to becoming a better software developer.

When it comes to meeting my course objectives, this project acted as a perfect foundational piece for furthering my knowledge of not only video game development, but video game design as well. Using tools like Unreal Engine 4, Blender, and Megascans serves as a great demonstration of my ability to use modern and innovative techniques, skills, and tools to develop and design a piece of software.

When I was creating this project, I learned many new things about the proper workflow when it comes to developing games, such as when it’s the appropriate time to use tools like Blender to create custom assets, or Megascans to import premade assets, or to prototype an asset using Unreal’s in-engine editor. I also learned that creating most game assets straight from C++ code isn’t entirely necessary and could actually become a hindrance further on down the line, as the project becomes more complicated over time. However, I would have to say that the most important aspect of game development I learned was how essential it was to approach a project with a specific goal in mind- reducing the chances I would get sidetracked tackling other problems that weren’t a part of the initial scope of the project.

I didn’t face that many challenges when it came to creating the code or the Blueprints to this short little game, mostly due to the relatively small scope of the project. One challenge I did face though, was to get the importing and exporting of premade files from Megascans to work correctly and the Send2Unreal tool in Blender to function as well. These fixes are relatively easy to remedy, so within a day or two I can start importing and exporting custom assets between all of the tools in my workflow.

This is a basic piece of software, and in my opinion cannot even be considered a game yet. However, with the way this project is structured, it could function as a springboard that could be used to create even more complicated, interesting, and fun experience to not only play but to develop as well. Another further enhancement that will increase the scalability of the project, and that is to connect this to a version control system before I make any major changes like Perforce or Git.

An example of some further enhancements I can make in the future would be to implement AI behavior to the NPC in the project which could then become an enemy. I could also use what are referred to as “Instances” within the Blueprint system to create multiple but distinct player characters that could be toggled between, and a variety of NPCs that can be easily edited within the engine. Thus, creating a simple party system and adding a layer of interactivity between the game and the player. Another example of potential enhancements would be to use “Instances” to create different enemy variations to help spice up the gaming experience and make it more fun.