John Boezeman (jlboezem)

Assignment 4

In this assignment, I implemented Scripting functionality in my engine using a new Property, Scripted. My implementation binds a GUID to a script location and allows the previously named scripts to be loaded and executed. Essentially, this is used to give a Game Object a Script that it can execute. I never actually ended up using it as such, instead creating new Game Objects to represent the scripts themselves as it seemed to make more sense to do so for this assignment but could be fairly easily used to do so by the use of Scripted.INSTANCE.loadScript(GUID); Scripted.INSTANCE.executeScript(params) in an event where it makes sense to invoke an object's script. I am somewhat happy with the implementation of the scripting system but do have some things I would like to change.

My system did permit me to do basic scripting, including handling events. I had to make a new Processor to use scripts for event handling, the ProcessorScripts which is used by creating a new instance of it by providing a Script and what the Script was interested in. This Processor could then subscribe and receive events in the same way as other Processors, then invoking the Script with the event it receives, allowing the Script to handle events. This is a simple way that works within my existing model but seems to be pretty inefficient as it forces the system to do file system I/O on every event the script is interested in due to scripts needed to be loaded then executed. I would change this to allowing scripts to specify whether they should be loaded at definition in Scripted or just have the path saved to allow the correct option to be selected to maximize space/time instead of forcing one version on all Scripts. Even though the current model is somewhat wasteful, it can still support most things with Scripts. Currently, I could still implement the vast majority of the game with scripts, including everything from movement to damage but it would be far less efficient than the "native" code.

I implemented three scripts across my two games, each doing a different thing. The first script simply moves the player a bit when a key is pressed, and was accomplished using my engine's Event System and existing Processors and triggered when a certain key is pressed. The second script prints out a message to stdout when something dies by handling events within the Script itself using a new Processor class to enable the script to receive the events. These two show the two major ways events and Scripts interact, the "native" handling and the Scripted handling. The final script is similar to the first in that it triggers on key press but instead of moving the player a small amount, allows the player to move very fast on normal move commands.

Ideally, I would like to implement functionality to allow Scripts to have various functions that can be called at the appropriate time when the GUID they are tied to does something requiring the Scripted functionality. Even without these changes, a game could implement similar functionality by just invoking the script of a GUID when correct and passing in the parameters required though it would be not as nice as the proposed changes. In my second game, I just used one script as much of the functionality I required was already written in native code but almost everything in it could have been implemented with scripts, from movement to shot spawning.