**MEMO**

**TO:** Kelvin Sung

**FROM:** Jonathan Earl

Jason Herold

**DATE:** July 25th, 2016

**SUBJECT:** Capstone Progress Report: Week 5 Textures

|  |
| --- |
| **PROGRESS** |

## Textures

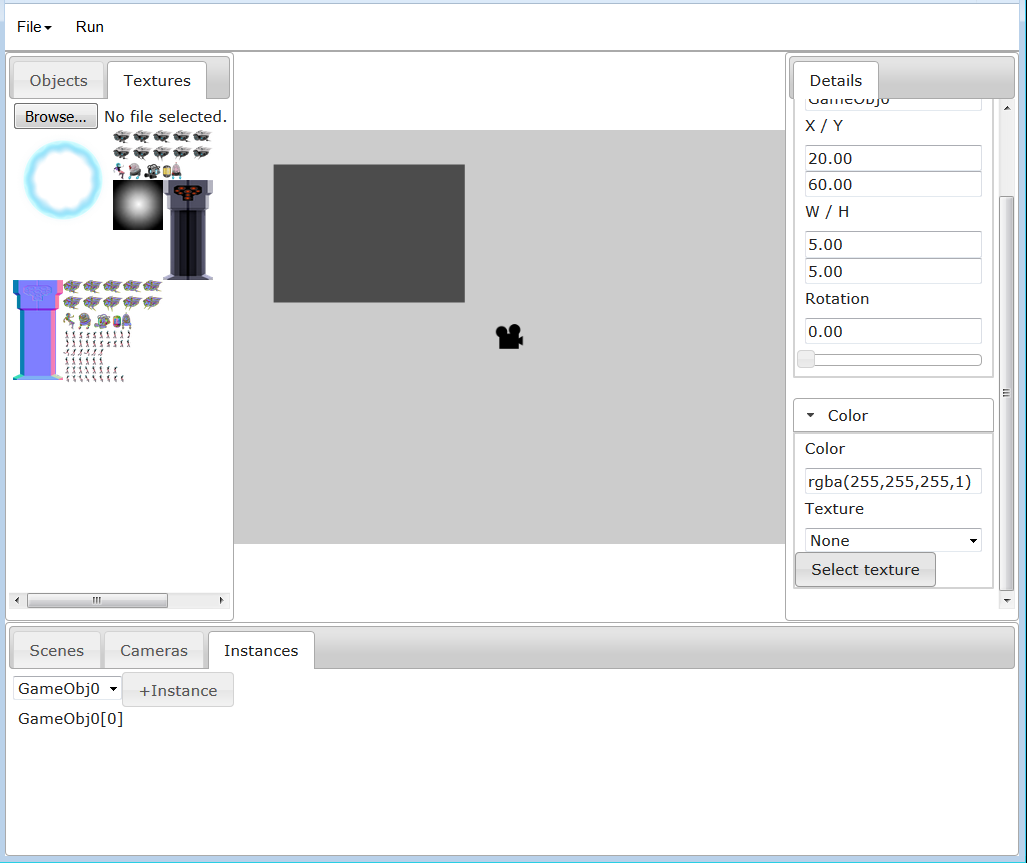
Last week we worked on completing adding textures support to the GUI as well as fixing the selection box to properly rotate with game objects. Below is a list of completed tasks. Below that is a few images showing the progress.

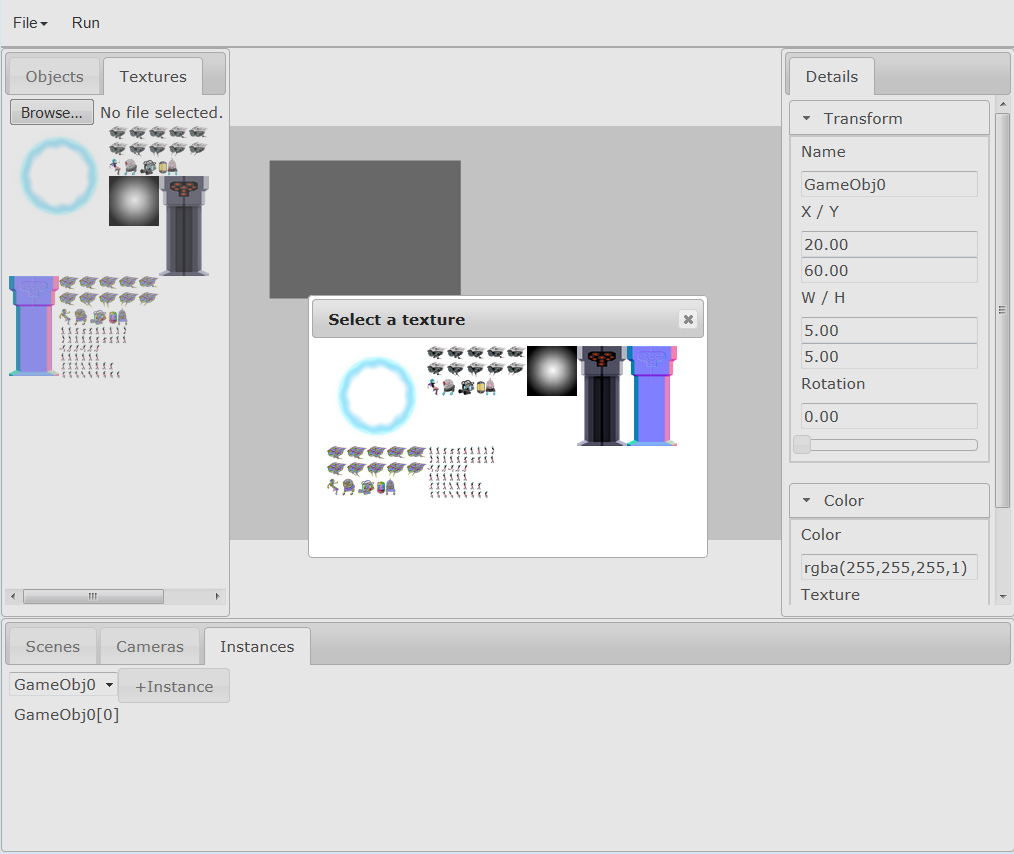
Completed work:

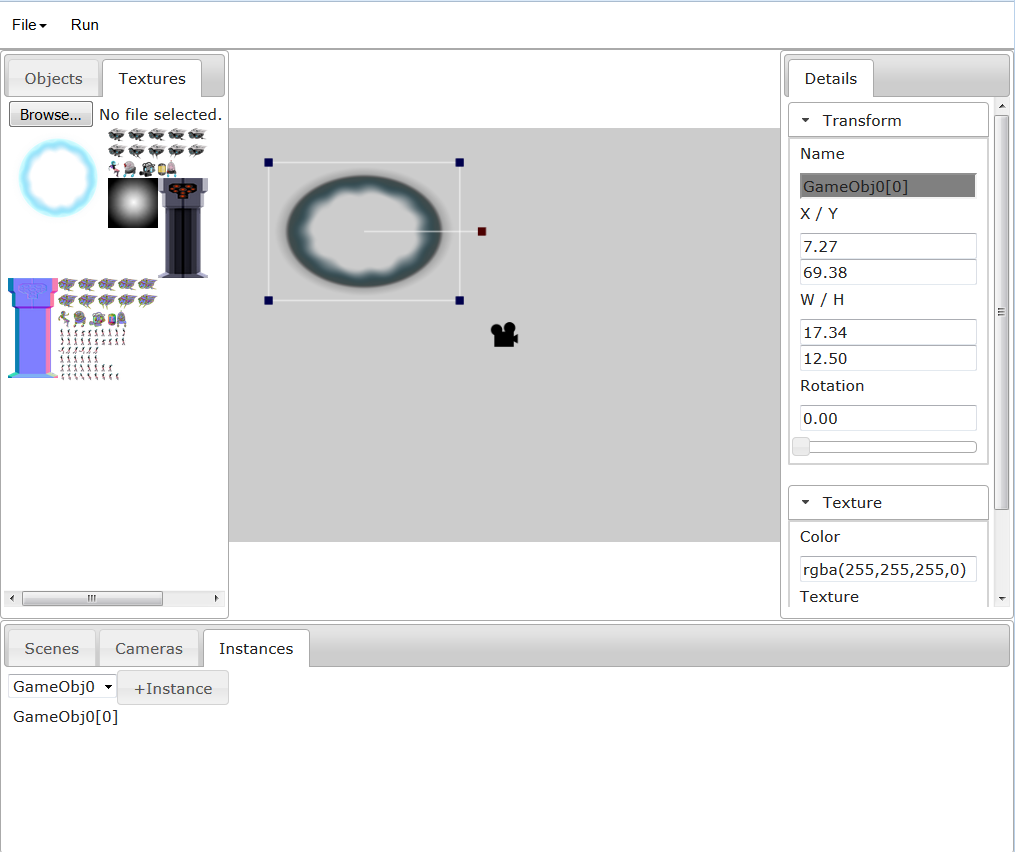
* Added the ability to create texture renderables
* Added a method to set game objects renderable (So that textures can be changed)
* Saving/loading textures
* Upload textures via file browser
* Corrected selection box on game objects
* Select texture via image drop down menu
* Select texture via dialog

Questions:

* Currently when you change the texture of an instance, the gameobject texture changes as well (which changes the other instances as well). Does this behavior make sense?
* We ran into some issues with dragging the textures into the scene (To create a gameobject and instantiate an instance of it). The JQueryUI selectable interface doesn’t seem to work with draggable -- they overwrite eachother. There’s probably a workaround, but is this kind of functionality something we should work on?

Texture display and texture selection display

Texture select dialog (User clicks on one of the images to select)

Post texture selection

## Backend Modifications

**Engine.Resources.Textures.js:** We modified this file in order to load textures which are uploaded via the browser to our GUI. As well as to load textures from memory, and from file into the resource manager.

**GuiBase\_TextureSupport.js:** for support of creating a texture renderable, adding a texture renderable class definition to the window, removing a texture renderable, getting a texture renderable. This class also supports replacing renderables within a gameobject. Also tracks the Image objects used to create each texture, so that the textures can be recreated when the game is loaded.

**GuiBase\_SaveLoadSupport.js** for support of saving the texture to file, loading the texture to file, saving the texture to memory, loading the texture from memory.

**GuiBase\_TextureSupport.js** was added for tracking which texture is currently selected, which textures have been added to the gui, as well keeping references so that the game engine may access these textures.

## Frontend Modifications

To allow the user access to the texture renderable functionality we had to modify the details panel so that a texture can be selected and added to the gameobject, we will also have to modify the texture panel so that texture files can be selected from disk and added to the Gui. The following files will need to be modified:

**TextureContent.js** was altered to show which textures are present in the GUI, it also allows gameobjects to be instantiated from this panel via right clicking an image.

**ColorTextureContent.js** was modified to allow the user to select textures via image selector.

**ImageSelectList.js** was created for displaying the image list

**TextureSelectDialog.js** was created for displaying an ImageSelectList in a dialog window

|  |
| --- |
| **NEXT STEPS** |

## Camera

This week we intend to add camera functionality to the GUI. For camera extension we would like to add the ability to control the camera with a camera script. A second feature we would like to add is the ability for selecting which camera is displayed on which layer. The addition of a third feature is to have more user feedback to zooming in and out of the scene view camera so that the user knows how zoomed in or out they are relative to the game camera. For these changes we will need to modify or add the following scripts.

## Backend Modifications

**GuiBase\_CameraSupport.js:** If we add the ability to create camera scripts we will need to create a new camera class that supports the creation of a new camera class like the way we create gameobject classes from scripts. This will require both storing the camera scripts and instantiating an instance of the new camera class.

**SubClass from Camera.js:** This will also need to be modified in order to define which layer the camera is on.

**GuiBase\_EditorSupport.js:** This will need to be modified to accept and allow editing of camera code

**GuiBase\_LoadSaveSupport.js:** This file will need to be modified to both save the code for cameras as well as save the instances of cameras.

## Frontend Modifications

**GuiBase\_View.js:** We will also need to be able to edit the camera code, so we will need to add a drop menu to right click on a camera list object.

We will need a way for the user to interact with the zoom other than hotkeys. We may do this through either added buttons as well as through hotkeys

**CamerasContent.js:** Will need to be modified to allow differentiation between instantiated Cameras.

**CameraLayerContent.js:** will need to be created to allow the user to define which layer the camera is placed on through a details panel

**ClientScene.js:** The draw function will need to be modified to display the cameras in the correct order.

## TimeFrame

This table is an estimation of the times needed to complete each objective. We expect the majority of the time to be taken up by integrating the cameras into their scripts.

|  |  |
| --- | --- |
| TimeFrame | File |
| July 25th | Add a create Default Camera method  Add a default Camera Code getter  Modify CamerasContent |
| July 26th-27th | Integrate Cameras into editor  Add ability to instantiate Cameras from View  Add ability to edit code from View |
| July 28th-29th | Integrate Cameras into LoadSaveSupport  Testing |
| July 30th-31st | Fix any bugs, tidy up code |