**MEMO**

**TO:** Kelvin Sung

**FROM:** Jonathan Earl

Jason Herold

**DATE:** August 1st, 2016

**SUBJECT:** Capstone Progress Report: Week 6 Camera

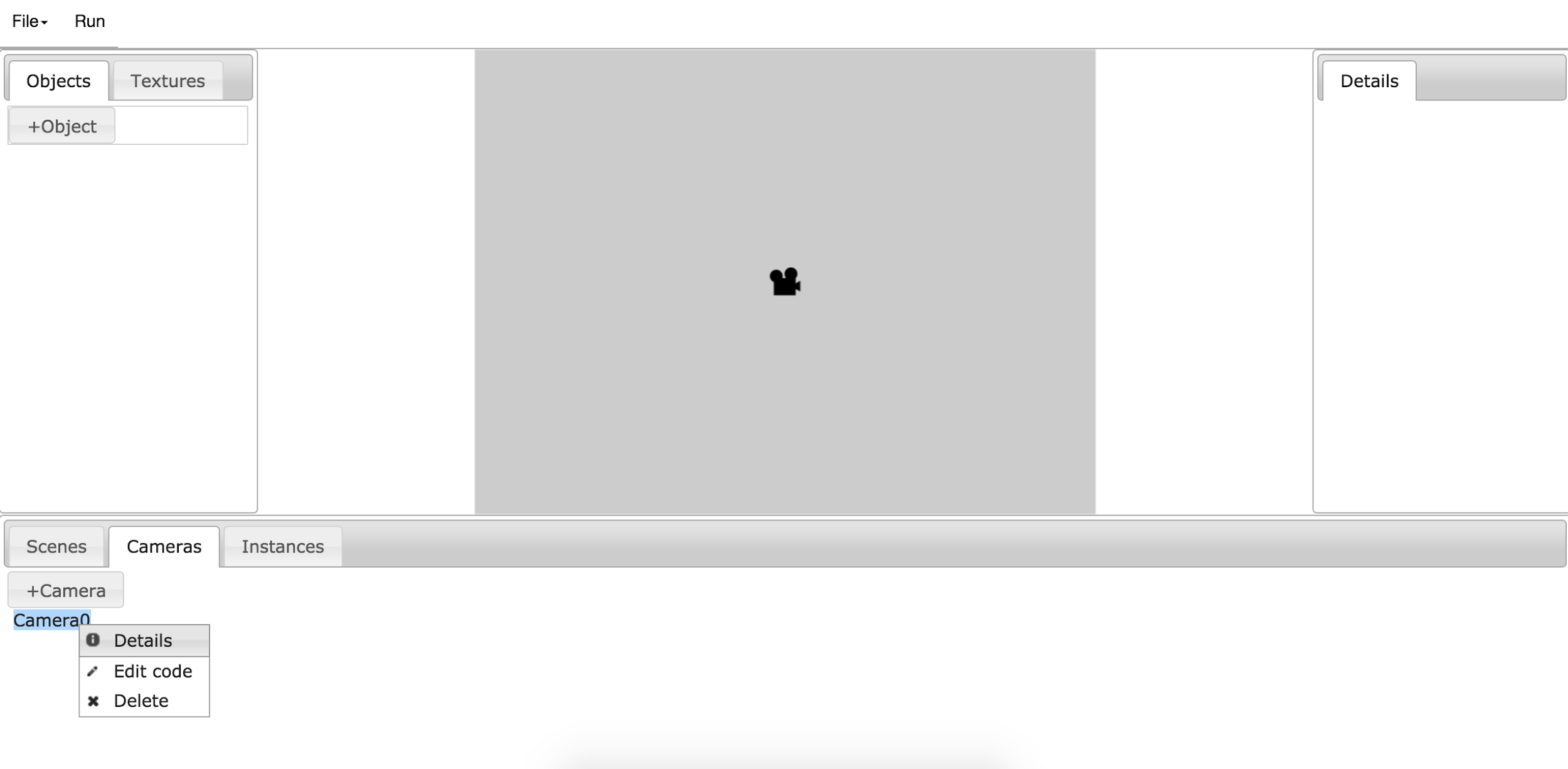
|  |
| --- |
| **PROGRESS** |

Work Completed

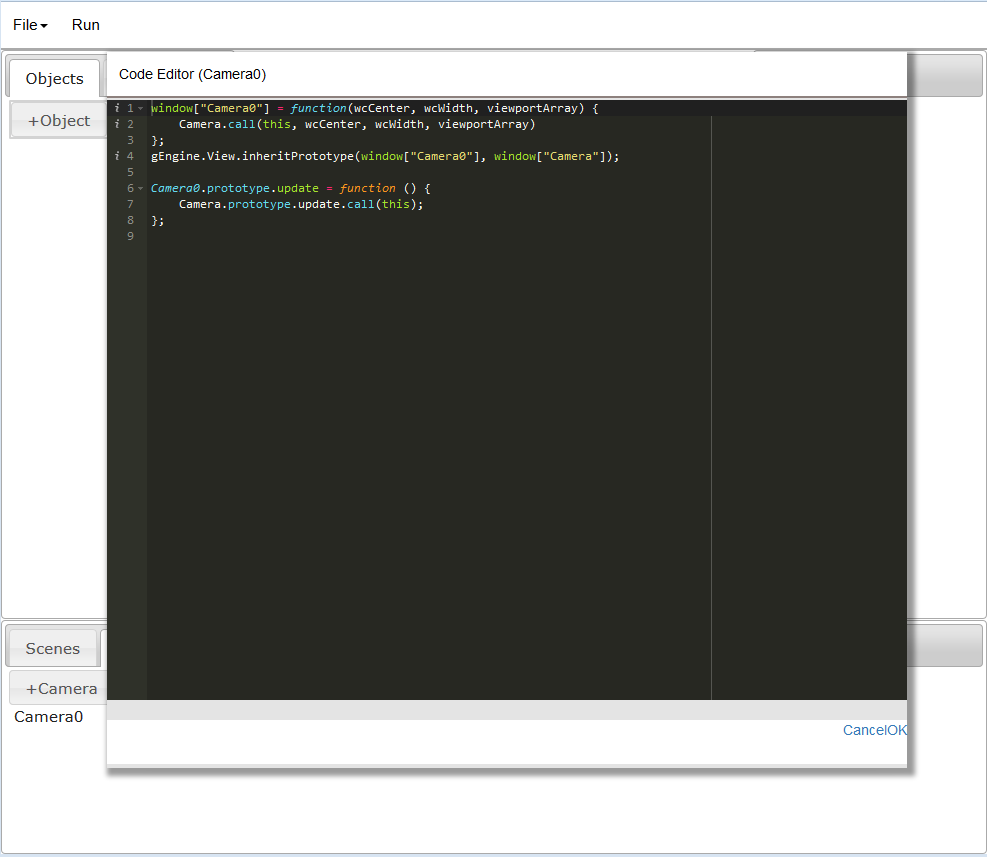
1. Added scripts to cameras
   1. Scripts can be saved/loaded
   2. Scripts work like the GameObject scripts except we do not currently support instances being created from the scripts
2. Cameras now support Layer settings
   1. These settings allow you to select which camera is displayed on top
   2. Layers are numbered 0-5, 0 is the top layer, 5 is the bottom layer
3. Instances now support order in layer
   1. Instances can be reordered by setting a text field in the details tab
   2. No set layers like with cameras
4. Direct manipulation refactor
   1. Changed long boolean checks into states, moved some functions into other objects
   2. Still long, but more readable
5. Texture deletion
6. General bug fixes (name conflict check wasn’t working), code cleanup/comments
7. Began work on lighting (started today)
   1. LightSet is populating and light details can be changed
   2. TextureRenderables can be swapped out for LightRenderables (not well tested yet)
   3. Next step is adding the actual Lights to any GameObjects that have LightRenderables

|  |
| --- |
| **Examples: Camera Scripts** |

Example of how to open Camera Code Editor

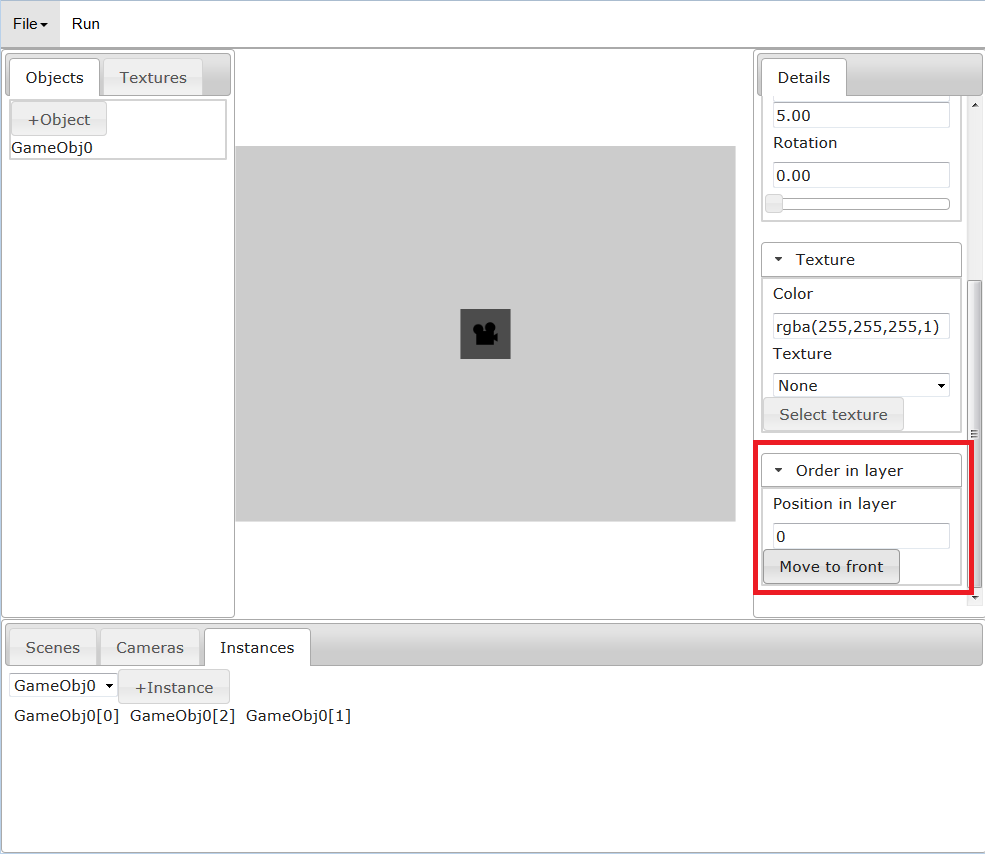
Right click the camera’s code you want to edit and select edit code 

Select OK to save, or Cancel to discard changes.



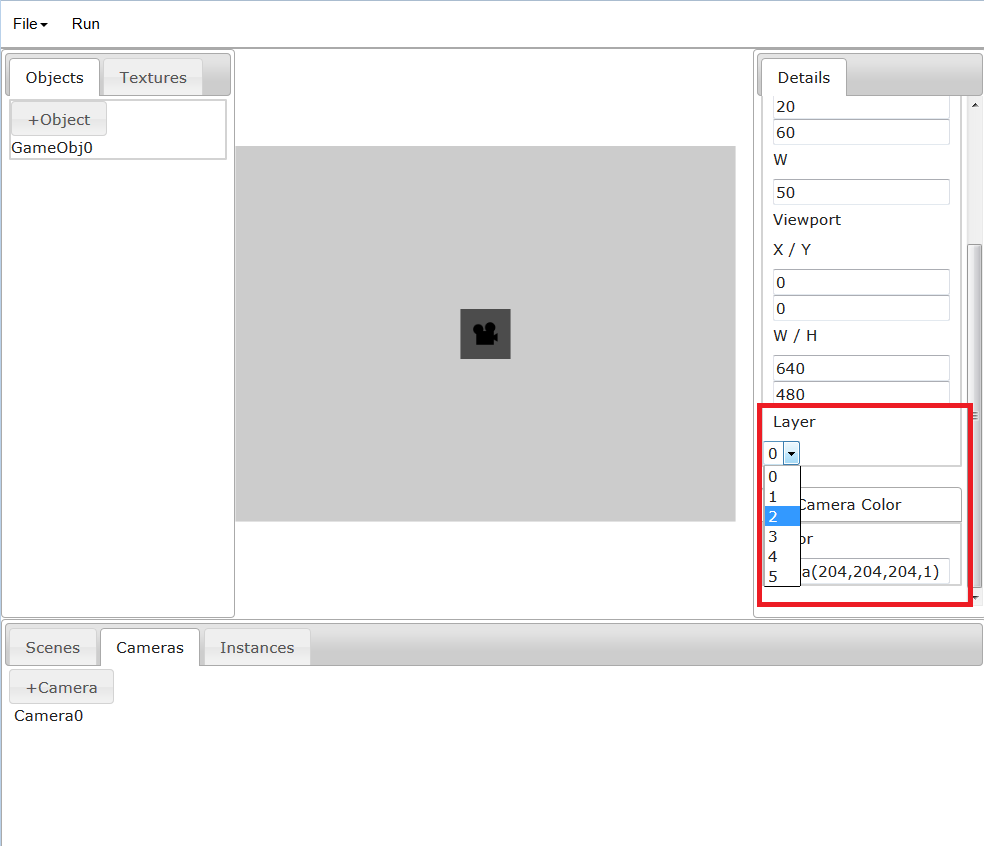
Example of Object Layers

Left click the instance and scroll down to the layer content. The layer may be manually typed or select Move to front.

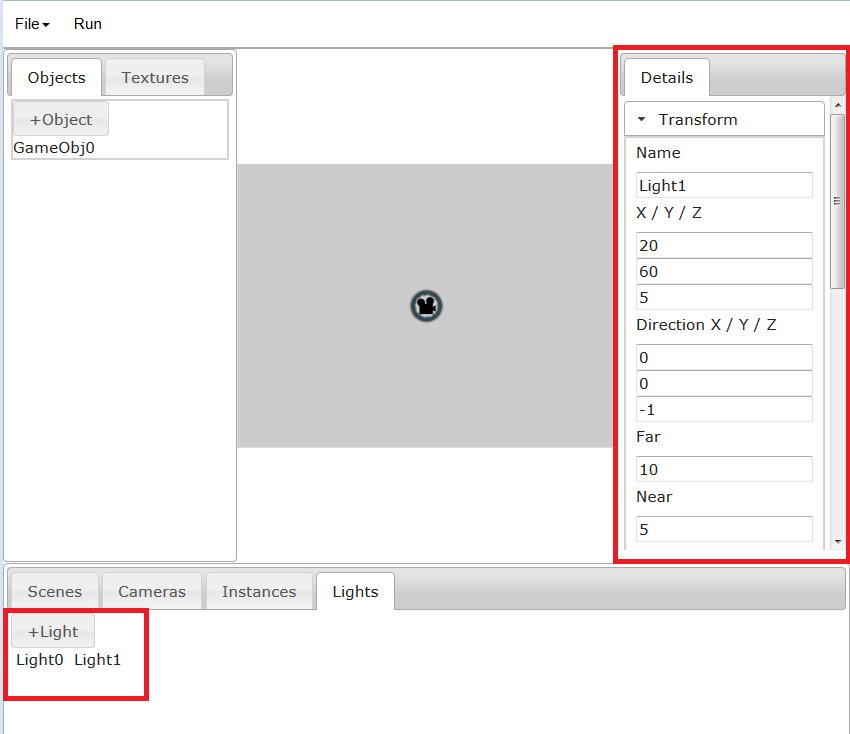


Example of Camera Layer functionality

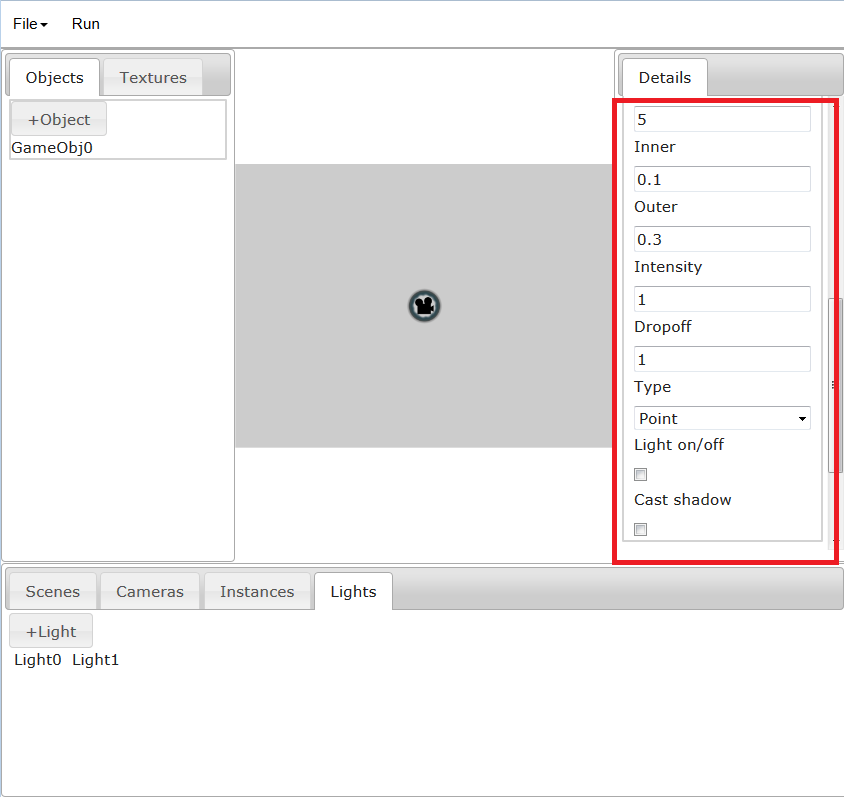
Left click to the camera in cameras panel to open its details. Then in the details panel left click the layer dropdown to select which layer you would like to display it on. 0 is considered top.



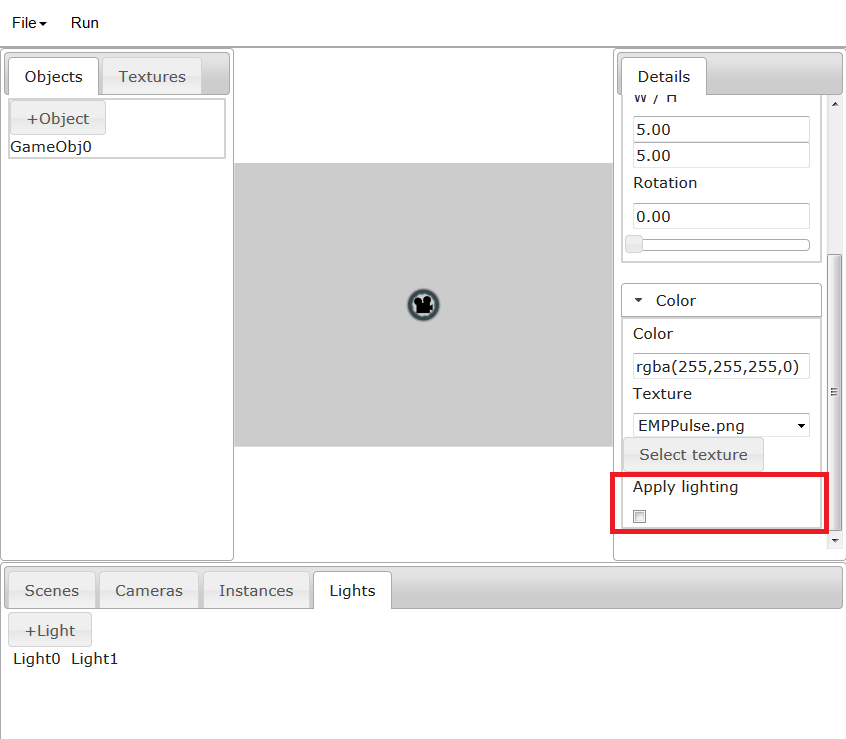
Light Support



LightContent and LightTransformContent



More of LightTransformContent



UI for allowing user to apply lighting to a gameobject. Currently this does switch the TextureRenderable to a LightRenderable and vice versa, but does not actually give the LightRenderable the lights.

|  |
| --- |
| **Code Modifications** |

## Backend Modifications

**GuiBase\_CameraSupport.js:** Now stores both cameras for look up, as well as their scripts

**SubClass from Camera.js:** On creation of a new camera, a new class is created with its own scripts

**GuiBase\_EditorSupport.js:** Minor modifications in order to differentiate between a camera and a gameobject

**GuiBase\_LoadSaveSupport.js:** Camera scripts had to be saved, and new camera classes had to be instantiated when loading cameras from file.

## Frontend Modifications

**CamerasContent.js:** Was modified to allow for right click editing of the camera script

**CameraTransformContent.js:** Currently the layer selection is placed inside of this content panel

**ClientScene.js:** Modifications to how the camera is drawn and updated in order to support layer ordering. mAllCamera now first references which layer then which camera (ie mAllcamera[layerIndex][CameraIndex])

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

## Lighting

We would like the ability to add lighting sources to the game via the GUI. This may be done by creating a light source and adding it to the scene. We would then like to be able to modify the light source via the details panel in the GUI. This would involve creating a light source, adding it into the scene and keeping a reference to it in order to modify its settings and position.

Steps:

1. Give the user the ability to generate Light objects through the UI, similar to what’s been done so far. LightSet will be in ClientScene. (Done)
2. Give instances a checkbox that sets them as a LightRenderable instead of a TextureRenderable. May need to disable this checkbox if no texture is selected. When the instance is turned into a LightRenderable, add the scene’s light list to the instance (Partially done)
3. Save/load lights. Will need to create new functions for saving/loading lights as well as modifying the object/instance save/load (Because some of them have lightrenderables)
4. Light removal. Lights will need to be removed from every gameobject/instance
5. Direct manipulation for lights.

## Backend Modifications

**GuiBase\_LightSupport.js:** This file will need to be created in order to create, modify, and remove light sources.

**ClientScene.js:** Will need to be modified to store light sources, as well as add and remove them from scenes

## Frontend Modifications

**LightContent.js**: will need to be created in order to store the tab information. This would be used to organize and access which light source you would like to edit.

**LightControlWidget.js**: add the ability to control the type, radius, distance, intensity of a light source and add this widget to the details panel when the light source is selected.

## TimeFrame

This table is an estimation of the times needed to complete each portion of the Light source.

|  |  |
| --- | --- |
| TimeFrame | File |
| August 1st | Modify ClientScene.js to accept adding/removing LightSources  Create LightSourceContent.js, LightControlWidget.js |
| August 2nd-3rd | Create GuiBase\_LightSupport  Attach GuiBase\_LightSupport to the Gui |
| August 4th-5th | Integrate Lights into LoadSaveSupport |
| August 6th-7th | Fix any bugs, tidy up code |