# 

# 

# 

# **GTCS Game Engine GUI Schedule**

July 18, 2016

Jonathan Earl, Jason Herold

# 

# 

# **Plan**

The following is the schedule of the things we would like to accomplish for our capstone.

# **Week of July 18th**

## Textures

In order to support Texture Renderable in the GUI these are the modifications in which we will need to make and the time frame we aim to complete them in. We will have to modify both the Frontend and backend to support adding Texture Renderables.

## Backend Modifications

For the backend we will be required to modify the following files so that the backend properly adds, tracks, saves texture renderables to the scene :

**ClientScene.js** for support of loading the textures to the engine, unloading textures from the engine, adding an instance of a texture renderable to a scene. These are the following methods that will need to be altered:

1. ClientScene.prototype.loadScene
2. ClientScene.prototype.unloadScene

**GuiBase\_ObjectSupport.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. We will alter the following methods:

1. CreateDefaultObject
2. CloneGameObject
3. GetDefaultCodeGO

We will also need to create a method for:

1. Change a renderable into a texture renderable
2. Also change all instances of a renderable into a texture renderable
3. Modify the Objects code to be a texture renderable
4. Add this code to the window

**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. We will alter the following methods:

1. fileSave
2. loadObjects
3. loadScenes

**GuiBase\_TextureSupport.js** will need to be 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 will have 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** will need to be modified to add new textures to the texture panel

**ColorTextureContent.js** may need to be modified so that it is hidden when a texture is added to an object

**TextureSelectionContent.js** will likely need to be created so that there is a way for the user to select which texture they would like to be attached to the gameObject

## TimeFrame

This table is an estimation of the times needed to complete each objective

|  |  |
| --- | --- |
| TimeFrame | File |
| July 18th | ClientScene.js  TextureContent  ColorTextureContent  TextureSelectionContent |
| July 19th-20th | GuiBase\_ObjectSupport.js  Add ability to show textures in GUI |
| July 21st | SaveAndLoadSupport.js  Create content for textures, add all needed functionality to remove/ add textures |
| July 22nd-24th | Fix any bugs, tidy up code |

All of the GUI widgets for minimum functionality been created, so it will just be a matter of plugging them into contents and attaching the appropriate functions. However we would like to enable users to select textures by image. This widget will have to be researched and created.

# 

# 

# **Week of July 25th**

## Camera

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 18th | Add a create Default Camera method  Add a default Camera Code getter  Modify CamerasContent |
| July 19th-20th | Integrate Cameras into editor  Add ability to instantiate Cameras from View  Add ability to edit code from View |
| July 21st-22nd | Integrate Cameras into LoadSaveSupport  Testing |
| July 23nd-24th | Fix any bugs, tidy up code |

## 

This is a large to the code base so these estimations are somewhat optimistic.

# 

# 

# **Week of August 1st**

## 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.

## Backend Modifications

**GuiBase\_LightSupport.js:** This file will need to be created in order to store, 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 |
| July 18th | Modify ClientScene.js to accept adding/removing LightSources  Create LightSourceContent.js, LightControlWidget.js |
| July 19th-20th | Create GuiBase\_LightSupport  Attach GuiBase\_LightSupport to the Gui |
| July 21st-22nd | Integrate Lights into LoadSaveSupport |
| July 23nd-24th | Fix any bugs, tidy up code |

# **Week of August 8th:**

# Options

1. Add multi-script support for Game-Objects
2. Refactor code
3. Flesh out draggable interactions
4. Add save to server
5. Re-organize Tabs
6. Make UI auto-resize to different screen sizes
7. Work on Poster