Game Engine GUI - version 1
Erick House - Fall 2015

OverView

The purpose of this project was to teach myself the basics of modern day front end frameworks. The audience for this are people with little to no experience with html, css, and front end tools. I'm going to assume the reader has some javascript experience.

The tools that I used were mostly random and based on what I had heard other people were using. After digging into it more I may have picked something more powerful but I what I did pick was still very useful.

I heavily commented the code to the best of my ability so someone can pick up where I left off.

Tools

- HTML: gotta have it for working the in browser

- Twitter Bootstrap: Twitter made a really cool CSS library it's much easier than working with css directly. It gets a little harder if you want to customize your html very specifically. I never got to that point though

- JQuery: A very powerful and simple to use front end javascript framework. Very common in the industry which is why I picked it. There is a large community so it is pretty easy to look things up on stack overflow. Their website is really nice and they have a ton of widgets for making life easier.

- NetBeans: simply because the game engine is already in it. Webstorm is another option that is really cool. If you are a student, you get a free education license.

In the index.html, I would recommend not moving the library code around until you know what you are doing. I created some problems for myself early on with the javascript includes not being in the correct order.

Through my reading I found that some other front end frameworks might be more powerful than JQuery but the syntax is more confusing for a beginner. I would look at angular.js or backbone.js if you are interested.

In order to get some of these things to work I had to edit the gameobject class in the game engine. I added a name and color hex code to the member variables for that class as well as getters and setters.

HTML

I can't say much about this. It really isn't very hard to learn and there are tons of resources online to teach you. Once you get the hang of the syntax it's pretty easy.

Bootstrap/CSS

The website has tools and examples to get you started as well as messing around with some of the code I wrote. Some things are somewhat confusing.

Bootstrap library is a good starting point for making web pages look nice. In some cases it does more than just style it. It can do things like automatically center html elements and so on. Refer to the documentation for more on the behaviors.

When something is referring to an html element's 'class' is means its CSS class (how the element is styled). This was a little confusing to me since I'm used to a class meaning object type. If the class is omitted then html provides a default style that looks pretty lame, which is why we are using bootstrap. An html object can have more than one css class.

Important syntax!
You will see very often something like: col-md-3. Bootstrap simplifies columns by dividing them up into 12 units. In the example col-md-3, **md** means a medium screen size. I used md as a default because it seemed to work well. You could also say sm (small) for something like a smart phone. The more important part is the **3** which means it's going to take up 3 units or 25% of the screen width.

Look at the documentation for more examples.

JQuery

JQuery is a front end framework that simplifies the interface between javascript and html. Trying to communicate to html without JQuery is very painful. The real power of JQuery comes from the event handling and the widgets. Many things you would want to do on a web page are already built. You can go to their website and download most anything you would need.

Most of what I did was simple JQuery. I downloaded and tried implementing one widget which was actually pretty easy.

Syntax Basics:

JQuery queries the html for a specific html element (DOM element) and returns you back a javascript object that represents the html.

The syntax for a JQuery object is `$(the html object I want)`

you can do things like:

```
var myObj = $(html object)
console.log(myObj.text( ) );
```

this will output the text inside the html tag

There are multiple ways to find an html element. These are the ones I know of but I’m sure there are more.

```
$(#’myHTMLID’)
```

- this finds a specific html object with a specific ID, very useful. notice the # syntax for finding by ID.

```
$(‘.myCssClass’)
```

- search for elements with a specific css class. Notice the (dot) syntax for searching by class.

```
$(‘div’)
```

- searches for html elements with the <div> html type

There are some really fancy things you can do but these were the ones I used the most. Once you find the html object, save the reference somewhere so you don’t have to keep querying it. Sometimes it’s hard to find an object again if it doesn’t have an ID.

From what I have found dynamically adding html elements to the web page is pretty easy once you get the hang of it. I have plenty of examples of it in the code.

Querying dynamically created html for events is a little more complex than querying static html. Using the standard syntax will not work for dynamic html.

you will want to use the event handlers **.on**

$(dynamic html).on(event type, css class type, function to be called);

There are several examples in the code to show you how to do it. There is more than one way.

Much of the power of JQuery comes from the event handling. Just starting looking at the code for examples in view.js. It should make sense from the comments I added.

Best Practices

There are all sort of thing that you can do with JQuery I would recommend looking at some design patterns using JQuery. I ran out of time to really look into it. The majority of the code is stuffed into view.js with the intention of moving it elsewhere basic on a design pattern such as MVC.

What I ended up doing was abstracting/hiding the JQuery and html into a javascript object. Please looks at the EditorPanel.js for examples. Everything is heavily commented. You should get a sense of how to abstract the html so the web page isn't one massive, static mess. I'm not sure if this is best way to go about it but I felt like it was a good place to start.

Resources

Most everything I did was through google search. These are the sites that I found most useful. If I were to do it over again, I would try and pick up a book for reference.

https://jquery.com
http://www.w3schools.com
http://getbootstrap.com