Course Experience

Edward Seim

February 10, 2014

This was my first experience ever with game development. Up until this point I had only done console level programming and some HTML + javascript stuff. This course thrust me into the depths of a full game development environment and new languages to learn. I would say the three grand things I had to deal with in this class were the environment, blueprints, and C++.

My first experience with the course was obviously the environment. The start of the course was getting ourselves situated with how Unreal Engine 4's environment worked. This included things like navigating the 3D playground, the menus and options, and understanding the workflow they intended to facilitate. Stepping into it, I was immediately recalling my previous experience with Adobe's Creative Suite. In highschool I had taken classes in Graphic Design and Unreal Engine 4 reminded me of that sort of navigation. There are menus for all the things in the playground, and windows for editing their options, as well as options for the whole world or game. One of the confusing things for me is where certain options are supposed to be. Because the underside of Unreal Engine is C++ the interface is built on top of how they laid out all their code. This means options you would think are in the main editor when selecting an item are actually in the item's window and vice versa. I believe this stems from the fact that they are trying to make a game editor available to graphic designers instead of solely programmers. For all I know, graphic designers would have no problem navigating the interface. If I weren't tainted by my programming experience, maybe I would have understood it better.

Moving onto Blueprints, I never once thought I would doing visual programming. Blueprints in essence function like regular lined code, just laid out on a 2D surface with lines hooking up all the various bit so they do what you want. Once I understood how Unreal Engine hooks up events and classes to items and things that happen in the game, everything else about Blueprints works the same as any object oriented programming. They did an excellent job, in my opinion, of color coding the links and boxes to signify the type of data that traverses them. For example booleans are red, so if I see red links or words, I know those are booleans. Arrays of each type have a symbol to signify their arrayiness with the color of their type.

Once we got working on EpicMan, I really started to get an understanding of how all the pieces of Unreal Engine and its editor come together to help you in your game creation quest. You have your main window for all the main aspects of your game and level design, with a 3d rendered view of your layout. You then have your classes which are each objects that you can place into the world. Their editor is based on what type of thing you are trying to create, weather it be a basic block, or a tool, or an AI that needs to react to the player. Once we got EpicMan to its basic featureset and with an eye on future development, I could see how Unreal Engine made it possible, easy, and quick enough for us to work on it in such a short span of time.