**Game Programming Essay**

**Rocket League Bot**

The purpose of this essay is to compare and contrast general purpose programming and video game programming. In order to do this I will take an in-depth look at both topics. First I will talk about ways in which they are similar, followed by how they are different. After this I will hope to provide a clearer picture to the reader as to the relationship between these two types of programming “methodologies” and how myself, as a software engineering student can take this information and apply it not only in this course this semester but also in my future software engineering endeavors.

First of will be the comparison between video game programming. On a basic level there isn’t much different about general purpose programming and video game programming. Both can be written with all the same languages. Java, one of the most popular programming languages, was solely used to create one of the most popular video games ever, Minecraft. Almost any programming language can be used for video game programming, all the way from assembly to scripting languages to fully developed object oriented languages. And because of this much of the same syntax and structures are used in both. Providing a player a decision that can have multiple outcomes could be structured in an if-else statement just like a variety of other things in general purpose programs. So in that sense a lot of the methodology on a basic level is very similar. And logic is another thing. While on a literal level the same programming structures are used, on a more abstract level some of the same logic can be used by developers. The underlying logic that runs video games and general purpose programs are identical and this is the logic based code. That is what you would see if you strippped a video game of the graphics and audio and all the extra bells and whistles that video games have to make them appealing and fun. That’ll be discussed more in the contrasts as well. A lot of the developer tools are similar as well. While there are definitely tools specific to video game programming that I’ll touch on in the next section, things such as IDE’s are used for both.

Secondly will be to contrast them. The primary difference has to do with the general structure of a game vs a generic piece of software. A game is structured with something called a game loop. This loop is the basis for the behavior of the game and is constant regardless of if a user is providing input or not. The game loop traditionally does things such as evaluating input, updating the game state and then rendering the outputs. From the time a game is started up this loop is running continuously. This is obviously a simplification but nonetheless, most games follow this programming structure at least on a very low level. Now take a generic piece of software, say a calculator program or a word processor that would be created through general programming. These programs are structured in a way that they are generally doing nothing while waiting for user input. Open up the calculator on your desktop and don’t provide it with any input and nothing will happen. Open up a video game and do nothing, things such as NPC’s, the environment and such can still be active regardless of whether any input is present. Going back to the similarity I mentioned of them being programmed similarly in terms of logic, syntax, and programming languages, the big thing that allows for this is a gaming engine and that is another big area in which the two differ. Gaming engines often handle the gaming specific programming on their own such as the API’s and libraries necessary for the graphics, sound effects, networking etc and leave only the logic based coding for the developers. The presence of these engines is not really something that is present in general purpose programming. And while many games are programmed in languages that are general purpose, there are also many custom languages specifically for programming video games. These are often more or less integrated with some of today’s gaming engines.

In conclusion I would say that general purpose programming and video game programming have more in common than not. It may be pretty obvious, but I would say video game programming is mainly just a narrower niche in which developers can put their programming skills to use. My biggest take away from this was my knowledge of the game loop, what it is and how it distinguishes a video game program from a generic software program.