Custom Game Engine Reflection

Shawnee State University
ETGG 3802 Realtime Two
Thomas Gilman

Over the course of the semester, learning to develop and modify our own custom game engine, it has given me a lot to think about and contemplate. Between the creation and usage of an in house custom engine, compared to an external industry engine; there are plenty of pros and cons to be considered between the two. Straight away, the pros to using an industry, off the shelf game engine is definitely time. A great deal of time has already been put into designing the engine that you would be using. This means that a solid foundation for the engine has been formed, which includes plenty of tools that have been tested and modified through feedback over the course of its development. Another note to consider is the amount of public resources and references surrounding that engine; as an already built public engine has had time for a community to form around it providing references, people to talk to about problems, and get clarification on documentation and ways to use the tools provided. All these traits make this sound like the go to solution for game development, this may not always be the case though. There are other traits to consider about picking a public industry engine compared to building and using an in house engine. One trait to consider is that some tools are restricted by a pay-wall, for a pro version of the engine, or a license stating that after so many sales of the products you create, you must start paying royalties; royalties are not a big deal to consider if the products you make are not going to sell passed the specified amount. This can be a bit more of a consideration if the developers using the tool are part of a company where the game is more likely to sell passed that specified threshold to start being charged for royalties. Using a custom engine means that you do not have to worry about being charged for royalties, or signing a EULA if you had needed to modify any of the tools.

While there is a significant amount of upfront time, effort, and money that must go into designing a custom engine. That engine however will be very malleable, and will not cost any more than the time and money put into it, as well as be customized or built off of for any task needed by the user or team. By using a custom engine, this also means the team knows the majority, if not all of the code base, so if an error or bug is present, the team can fix it right away, rather than having to wait for an external company to fix the issue. All of this does not come easily if while building the engine, it hasn't been properly documented and organized during that process. If the engine has not been properly documented, it raises the difficulty of building a game around the usage of unknown functions or tools. Those are reasons why industry public engines are used and supported so much, as it is due to the documentation that has been built around it. If the engine being built is properly documented, which could be done as its being built; it will allow its users to easily develop and build a game from it. So in the case of a company that uses a custom game engine; anytime an artist requires a tool that could make their work much easier, or improve the process of their work, or even make improvements, they could suggest or make mention of these changes and they could be made right away by the programming side rather than wait for an external company that develops the engine to make changes or having to modify someone else's engine.

When picking a game engine to use for development, whether it is an industry engine, external custom engine, or internal game engine; documentation is very important. Without good documentation, or even any documentation at all, it makes it very difficult to use the tools provided. Without documentation, all the users would be doing is for the most part guessing or spending a majority of the time, trying to figure out what certain functions or tools do, which would essentially be about the same if not worse than building your own custom engine from scratch. Therefore, properly documented tools, references, or even a community surrounding a

game engine is very ideal and is important to take into account when choosing an engine to use for development.

Through the process of developing our own game engine over the course of a semester, I have learned more about the process going into building a game engine. While we have used tools such as Ogre and SDL for handling the visualizations of our game engine. The process has allowed me to learn more about some of the functions and tools necessary to create an engine that can be easily used to make a game. Some of the tools I can take away from this experience would include being able to integrate python in with our c++ code, which seems extremely useful. Other experiences I can take away from the course are improvements to being able to integrate with pre existing libraries such as tinyxml and improving my c++ skills using templates, classes, and handling data. Altogether I feel as though I have learned a great deal about making a game engine and have learned a great deal from the course in all. Some improvements, or sections I wish I could have further covered would be improvements in performance and optimizations of our engine; learning more about optimization techniques for game engines would have been useful. While I know a lot of the optimization techniques are taught in the Optimizations course, being able to see a glimpse of them being applied to creating a game engine would be interesting and convenient to explore

If the tracks approach is implemented to the major, I think it could be very useful and nice to see done. Using the engine that is built in Realtime for Senior project and building off the engine would really showcase the content taught at Shawnee and throughout the major, as well as give students a chance to showcase their skills including learning to integrate further with other people's code. The only downside that can be seen with this, would be if the put together engine is underdeveloped or a lack of process has been made, then it could possibly stagnate the project. Overall, I have really enjoyed the course as a whole, and am excited to take away new skills and experiences from it.