School of Computing
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

COMP5530M Group Project Game Egine

Individual Report

Group 4 : Quasarts

Chaoshan Huang

sc20c2h

Submitted in accordance with the requirements for the degree of
MSc High Performance Graphics and Game Engineering

<2021/2022>

Contents

Project Learning

Technical Learning

Premake

For a group project, it is inevitable to face the problem of configuring third-party libraries in different development environments. It is kind of stupid to upload complete compiled files, which has a large size, not to mention that our team needs to develop under Windows and Linux platform at the same time. In this case, we choose to use premake to manage the project.

Premake is a convenient third-party tool to manage third-party libraries and generate project files automatically, so that developers don't need to worry about project dependency issues when they are in different development environments. And premake is more user-friendly compared with cmake. It uses lua as its scripting language, the language grammar is easy to understand, which allows developers to understand configuration information more intuitively.

In the process of project development, I deeply realized the advantages of premake. As long as we make the scripts corresponding to the modules we are responsible for, others team members can also use them to configure the project easily.

Git

Similarly, project version control is also a very important part. This project is my first time participating in group development, I don't know much about git before this. With the help of git, we can develop collaboratively more efficiently.

I use git command line and Visual Studio git tool during developing period. And honestly it really helped me a lot. We can develop our own modules separately and then integrate them. Plus it can trace the history of the project. We did have a project issue once, thanks to git, we could do project version backtracking and avoid bigger problems.

Audio Related Stuff

Audio is a new area for me. At the code level, playing music is not as simple as using the player directly. It requires us to perform a series of operations such as decoding the audio, analysing the audio signal, etc. In order to implement it, I need to learn a lot of new things, including OpenAL and sndfile.

Sndfile

Sndfile is a third-party library which can help to decode audio source and convert it to buffer, and then it can be used by OpenAL.

OpenAL

OpenAL is a cross-platform audio API. it can deal with audio buffers and play them. And OpenAL is similar with OpenGL, so it's easier for me to learn. Besides, it supports 3D audio effect, which could make our engine more alive.

As I mentioned, I'm fresh in Audio area, and with this experience, I learned a lot new stuff.

Script Related Stuff

For script system, it gives me a chance to learn a script language, which is popular in game industry.

Lua

For scripting, we have many language choices, such as python, c#, lua, even C++. Finally, we chose lua as our script language, because it is more commonly used in games, and it is relatively simple and easy to understand, which is more friendly to developers.

As far as my personal experience is concerned, lua is indeed a relatively intuitive scripting language. I believe that learning it will also be very helpful for my future work.

Sol2

Sol2 is a "bridge" between c++ and lua. it wrap the lua virtual machine and provide lots of function for developers to use lua in c++. Without sol2, I would probably suffer a lot from integrating lua.

Latex

To be honest, it's my first time using latex to write a report. At first I was intimidated by its complicated syntax, but I have to admit, it is very helpful for editing text formatting. Maybe I'll try to use it more in the future.

Non-Technical Learning

In this team project, I not only learned a lot from the technical side, but also on the non-technical level.

Firstly, the biggest gain is of course teamwork and communication. This is not only my first more formal team project, but also the first time to cooperate with students from different countries, different culture. In development, we often have different opinions, and we need to discuss and get a final solution at this time. In addition, we also need a lot of communication during project

integration to understand the needs of the other members. At the beginning, I was very uncomfortable with this. On the one hand, it was due to the language barrier, and on the other hand, it was caused by different cultural thinking. As we understood each other, the team communication became more and more efficient.

The second is the planning of the project. At the beginning, our plan was too large and perfect, but as the development progressed, we encountered a lot of obstacles. Under the suggestion of Dr.He, we changed the plan in time and decided to implement a relatively basic engine first, and then expand it step b by step. This change is of great help to the advancement of our project. From this, I also realized that in group development, we first need to implement the most basic content, which is the most important.

Besides, Always have alternative plans ready, to prevent the original plan from being blocked, so that changes can be made in time to avoid affecting the progress of the project.

Assessment and Reflection

I have gained a lot from the project experience that lasted for more than 4 months. In the project, we did face a lot of issues, but fortunately these problems were basically solved.

For me, the easiest part is probably the planning in the early stage (but it caused a lot of difficulties for us in the later stage), and implementing my own independent modules, because the content is all up to myself, and I am familiar with the framework of the modules, so that I can easily extend or modify them.

The hardest part, of course, is integrating the modules, because not only does it require you to make the necessary changes to your own parts, but you also need to understand the other's module design. And we all lack actual practical development experience, so everyone's module structure will be very different, which increases the cost of completing the integration.

However, the entire project development process is very interesting. As the project progresses, we gradually become familiar with each other and work efficiency gradually improves. And it really exciting to see that our engine which is from scratch can actually do something.

If given the chance to restart the project, I might try different modules like rendering and animation. And with this experience, we will have a more complete understanding of the game engine, which can make our engine better.

Of course, we also have a lot of regrets, the biggest regret is not making a real game, maybe we can achieve it in the future.

Overall, this 4-month group project has benefited me a lot. I have learned a lot of things that are not in the textbooks, and I have also put into practice what I have learned. I deeply learned that in practical development, there will be not only technical issues, but also many other non-technical issues. These experiences will allow me to solve problems faster in the future when I have teamwork at work.

Finally, I would like to thank our project supervisor Dr He Wang and project coordinator Dr Raymond Kwan, who give us lots of help during our development. Also, I would like to thank every Quasarts member, Working on the project with you will be the most valuable experience of my MSc master’s period.

Appendix

Timesheet

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Research	3	8	3	2			2	5		1.5					2		
Individual Coding	4	4	4														
Audio System Coding				12	8	8	7	2				3	2	5	5	5	
Script System Coding								12	6	7	7	6	6	2	8	10	5
Integration Coding						3	2		1	4		2	4	3	5	3	
Meeting & Discussion	2	4	5.5	3	5	5	5	5	5	5	4	5	4	4	5	4	7
Writing			2	4	4	2									2	1.5	
Others				2		6									2		5

My working time starts on 28/01/2022(week1), and ends on 22/05/2022(week 17). For more detail, please check my individual logbook.

Individual Coding part includes the coding exercises during preliminary research period and some modules codes that originally needed to be implemented by me but have now changed: such as windowing system and editor.

Audio System Coding part also includes learning and coding exercises about OpenAL-soft and sndfile, which are third party libraries to play audio.

Script System Coding part also includes learning and coding exercises about Lua language and sol2.

Others part includes learning time about premake, github and latex. Besides, the time to fix code bugs and prepare the presentation, etc.