

TECHNICAL REPORT

S3342133 Phan Thanh San

The Design Pattern that our group has implemented inside the assignment program- which we call the cyber game, is consist of two design pattern that can work together as one to support our program. The two design pattern are: Strategy Pattern and Factory Pattern. The reason why we choose these two pattern- in my opinion, is because of their capability in handle things with the technology we use. Since Strategy pattern allow us to develop the program in a theoretical state in which we can plan ahead everything with the available source, then the factory pattern allow the optimize and generate code like a factory so we can easily maintain the program.

As mentioned above, we used the technology call Unreal Engine to support our program. Unreal is a great technology which allow its user to freely modify the code as well as its large library of graphics and sprites to their own needs. If i have to compare the benefit of this technology to our classic Java it would show a significant benefit toward Unreal. In terms of graphics, Unreal support a graphical view which is easy to understand and modify. To sum up, the graphical view support of Unreal allow a design of 3D dimension which is a good things in developing game like first person view- first person shooter. Unreal also allow user to freely rotate its camera to create more perspective and thus, we have managed to create a third view person game thanks to the support of Unreal Engine. Following the support of graphics is a massive library of source code that allow user to freely implement and create new things from available library. This is one of the reason why one line of code from Unreal can do much more things compare to 40 line of code from Java. User can freely integrate, create, and implement existed library and generate new ideas with the easy modification of graphic interface. Another feature of Unreal Engine is the compactable of Steam Engine, this allow our game to run through steam and let user to communicate with their friend on steam client.

However, even great things have its own downside, Unreal Engine is also not an exception. First is installation: to actually run unreal you first have to have third party to read and execute its code such as Context. Unreal Engine can only run on Window, so any attempt

to run Unreal on others OS is not an option. Its massive library require large resource and take up lots of space to runs, thus weak computer like laptop with a simple core dual might have problem in running the graphic emulator. The installation of Unreal Engine also cause user problems. The Unreal Engine has an auto check when compile but this does not make Unreal a great tools but instead the opposite. Unreal Engine check for changes made in the created project folder locate in the src file, but changes that include the replace of an entire file will make Unreal unable to compile- this means you cannot simply override the exist file with a new one like in Java. this is the error we got in the attempt of demoing the latest of what we got in the first demo day. Last but not least, is the emulator itself. Unreal only have the graphic emulator which allow user to create a beautiful graphic interface, but the emulator to actually run the code is none. Compare to Java in which runs on an emulator, all code from Unreal is written in form of script. This means there are no suggestion to write code, no on-time alert error like Java. All checking will only happen in the execution time. thus, this will take lots of time for a developer to learn, for us it takes each of us at the cost of 2 weeks to actually learn the simple basic of this engine. Even so, this technology allow us to have lots of opportunity in creating what we needs, thus, choosing it for any project that need a short time to develop is actually a good option. But good technology will be nothing without a good teamwork.

In our group's case, our team function in good scrum sprints as each of our team member was able to do what they do best. we held a scrum meeting every weekend to check the status of our project's progress as well as assign new task to each team member. This allow us to keep track of what we have done so far, what we have not done and what we currently do. However, a perfect cycle can only function probably if there are no impact can affect its element, in our case is our team member. Some of our team member suffer from sickness, sometime work from other course, and some time part time job. thus, this cause some of our meeting to be cancel. However, we manage to find a walk around as we try to use any available time we can to come and talk about our problem. this makes up all the time we miss and help us to continue in making the project. In my case, i have done all assigned task i can to help with the team. But i have to admit that my contribution is still far less than the contribution of the three others members. Overall, our teamwork is in good condition.

s3275049 Bui Trong Nhan

Design Pattern

Simplified Factory (creational) and Strategy (behavioral) are design patterns to be applied into the game. The game requires a groups of enemies that can move around and shoot as well. A variety of items are also an important factor of the game. Therefore, Simplified Factory should be applied into the game design. At the end of Scrum 1, we recognized that enemies should be spawned randomly so the factory class was slightly modified by adding a function called `SpawnEnemyRandomly`. Enemies also can behave differently in the game such as moving around, shooting and focusing the target. Strategy design pattern allow developers to define a family of algorithms and encapsulate behaviors. As a result, Strategy design pattern is the most suitable design pattern that can be applied to make the game design more beautiful and more logic when adding new behavior into the design.

Technology

At the beginning of the semester, Java and Spring framework were our only solution to make a client server game like Software Architecture course. However, after some days researching the internet and game-related websites, Unreal Script and HTML5 were added into our technology shortlists. Both programming languages can support game programming and also easily apply design patterns. Days passed, a team member in our group has featured the Unreal Script so our final decision was to make a client server game in UnrealScript technology. Throughout the game making process, Flash is also used to make screens and game graphical user interface components. There are a variety of tools to work with such as ConTEXT Editor, UDK Editor, and Git Hub.

About Unreal Script, it is a powerful game engine that helps creating powerful game. It is designed for simple and high level game programming. It also supports object-oriented programming without multiple inheritance (similar to java). However, it also has some disadvantages. It requires a -strong-enough computer to be able to develop and compile code. It uses professional design tool to build 3D actors so it is hard for beginners or non-designer people.

Participating in a good team is a great advantage. Scrum meetings were well-designed each week to assign tasks and discuss together about features to be implemented. The team had meeting outside classes each week at school and we also had group meeting after lecture. About the report, the team shared a Google document to accomplish sections of the report and help others to check the report (format, structure, images, references...).

Section in the Demo presentation was also be assigned before the demo.

s3275145 - Hoang Ngoc Thanh

Technology

For the project, our team use Unreal Script which is C++ base, and use UDK Engine to simulate our game play.

Advantage

- UDK Engine provides many super classes that we can extend. For example, game character can be extended from the Actor class supported by UDK Engine; we just have to override necessary methods.
- UDK Engine provides UDK Editor, which help building maps, characters, weapon, and other objects very fast.
- UDK Engine is server-client base, so the connection between characters can be automatically done by the server by stimulating replication method.
- Support DLLBind to integrate with other programming language.

Disadvantage

- Although UDK Engine supports Scaleform, a technique that use flash file to make displayable parts for example main screen, information bar of character, and inventory; the connection between these parts with the game play is really hard.
- Action script 2.0 and 3.0 is also different and invoking methods written by ActionScript 2.0, and 3.0 causes many problems.

Team work

Our team has for members, and all members tried the best to participate the project. The most advantage is the connection between members, we can easily contact with each other and control the project progress, not only follow the progress on github, but also by email, and mobile phone.

After scrum meeting, each member has a specific task depending on the member's ability. If some task cannot be done before the dead line, all members had very good attitude on finding the reason, and solving the problem.

Environment working is very friendly and comfortable. We almost do the project at school, and sometime received feedback from other students.

Design pattern

For the first time of doing project, we control the project mainly by logical thinking. Our project try to brainstorm how the game look like?, what is result? And what is the next thing have to do? Then, we discover that applying design pattern into the project have many

benefits. It helps our project code cleaner, clearer, and focuser. That's why we applied factory pattern and strategy pattern into our project. With factory pattern, we can create separate factory to create many types of different objects, and strategy pattern control object's behavior, it also help us in debugging wrong behavior if occur.

s3260624- Pham Dinh Son

At first, we chose Java + Spring for our project. Because we had a choice, we decided to ditch it and chose Unrealscript. An entire API dedicated to making games, Unrealscript was ideal.

I mostly involved with AI programming and Network programming in this project. AI behaviors can be managed using states (as in, state of mind), so it is generally easy to program simple AI behavior sequences. Given more time, we can add more behaviors to AI. Switching and swapping states around can produce interesting AI behavior sequences (such as run way and find cover then wait). We implemented some Navmesh pathfinding for our bots. Unfortunately, it is not perfect- bots get stuck sometime in manners similar to Daikatana's bots.

Network programming is perhaps the most confusing part of Unreal Engine. Basically, client and server use the same code; we do not need to write server code, which is a good thing. The bad thing is, they use the same code; and trying to sync data between clients can be problematic. In one class, such as Player class, you have to make sure all variables that vary are replicated among clients; otherwise, they cannot know them. Functions that are invoked on client machines are not replicated unless you tell the server that you have invoked that function. Variables that have changed on the server are unknown to clients unless they tell clients to update certain data. In other words, what you see on the client might not be visible to server and what server sees might be not visible to clients.

Furthermore, there are some functions (provided by the engine) that only work if you start the game in single player mode. How do you know which one works and which one do not? Trial and error.

Most of the time I find myself in situations that forced me to read through documentation and modify certain functions of the engine. Unreal Tournament 3 is a competitive arena shooter; the amount of efforts it took to convert it into a top-down Alien-Swarm Bullet-Hell shooter game is too much for the team to add additional features. It is quite unbelievable that the team somehow managed to implement SQL Database, despite the time constraint.

Another obstacle in our project is that none of the team members are 3D designers. 3D Models have to be reused over and over again. Working with flash animation also proved to be difficult, since all members did not know how to animate flash objects.

+Teamwork:

_SCRUM proved to be effective in this project. All members attend meetings 100%. Each

week the team contributed to the project tons of code and ideas for the game.

+Design Pattern:

_We mostly used factory pattern since the requirements asked the team to generate random enemies, items, world objects in real time. With the help of UDK, factory pattern really shines in this project. All you need to do is to drag a factory and drop it to the UDK and it will produce appropriate objects for the game. Each factory also acts as owner of objects so we can trace their identity