Developing a Videogame using Unreal Engine based on a Four Stages Methodology

Carlos Mauriccio Torres-Ferreyros
Universidad Peruana de Ciencias Aplicadas
Lima, Peru
U201110968@upc.edu.pe

Matthew Alexander Festini-Wendorff
Universidad Peruana de Ciencias Aplicadas
Lima, Peru
U201011204@upc.edu.pe

Pedro Nelson Shiguihara-Juárez
Universidad Peruana de Ciencias Aplicadas
Lima, Peru
pedro.shiguihara@upc.edu.pe

***Abstract*— The goal of this project is to prove the development of a videogame using Unreal Engine, based on an agile methodology that is viable in an economic, quick and sustainable way. This methodology has four stages that are: preproduction, production, testing and postproduction that were advantageous to finish the project on time. To achieve this, we have designed and developed an action platform game following the previously mentioned stages. In conclusion, we achieve to prove the applicability of the four stages methodology since we made a high quality game in a short period of time, using limited resources.**

Keywords—CryEngine; Videogame Development; Unity; Unreal Engine

I. Introduction

Since 2006, some videogame companies such as Epic Games, Unity and Crytek have decided to sell their game engines products for a low subscription fee or completely free [1]. A game engine allows videogame companies to make games of high quality. In the case of Unreal Engine, for example, it charges 5% of sales generated for the game on the market [2]. This has led to developers and designers to create quickly games of high quality with modest sums of money. In contrast, ten years ago the cost related to create a videogame was about a million of dollars. Because of this positive change, for a new videogame company, the opportunities to enter to the videogame industry have considerably increased.

Because of this background, in 2015 the students Evelin Espinal and Flor Huisa decided to conduct a research project based on the system proposed by IGDA, the system itself consists that the game development must have 4 marked stages ("Preproduction", "Production", "Testing" and "Postproduction") in which 9 areas are developed ("Critical Game Studies", "Games and Society", "Game Design, Game Programming", "Visual Design", "Audio Design", "Interactive Storytelling", "Game Production" and "Business of Gaming") simultaneously in interactive way during the development stage. They concluded that it is viable to use this methodology and it was left it as a background for future research works that could validate it. It is because of this that our work focused on development work based on them as a direct source of information and focused it on using a game engine[12].

The main benefits of using a game engine like Unreal Engine are: reusable code with libraries, the concept of object oriented programming and process of computer generated graphics. The engine has specialized libraries for game development, it's not about thinking how to develop a game, it's about using the offered code to focus on the idea and not the technical details. In addition, the use of object oriented programming, makes it possible to create class the traditional way, but in a very intuitive way such is the case with the Character class, this class contains all the necessary code to create a character on the game. Finally, we use complex algorithms to calculate the graphics the game needs, Unreal Engines uses the Disney's Physical Based Rendering [11] to calculate the lights and shadows on real time, so it isn't necessary to program on a low-level programming language, you can focus on decorating the game.

Our proposal's objective is design and create a game for computer on Unreal Engine 4, proving you can create any type of game, it has been decided to create platform game in a phantasy universe. In a platform game, the player has to evade obstacles and defeat enemies until reaching a point in the map. Likewise, game development methodologies will be used to

978-1-5090-2532-9/16/$31.00 ©2016 IEEE

find the best possible improvements and considerations to improve the Entertainment Software Development Process thesis. In such way, it's planned to details all process of designing and game development of the several areas involved in the Project cycle life, the name of the game is Vermillion.

The article is organized in five sections. In section two, articles that used the Unreal Engine technology and its alternatives will be analyzed, to emphasize the positive and negative aspects of each project, likewise, a comparative analysis with the Vermillion Project will be made to rescue the negative and positive aspects of the Project, as well as the similarities with the rest of the designs. On the third section, Project Vermillion is proposed, an action platform game designed and developed using Unreal Engine game engine. In the fourth section, Vermillion will be validated if the methodology used in game development can reach a high quality visual standard, gameplay and user experience. Finally, the last section will present the conclusions.

II. State of the art

In the University of Massey, an Unreal Engine 4 simulator was made using Virtual Reality tools. The objective was to submerge the player into a virtual world where we could see a movie on real time. The immersion in a virtual world turned out to be a richer experience compared to a common movie, the interaction with the scene made a world of a difference [5]. In the same University, the students designed and developed a war simulator on Unreal Engine 3. The purpose of this project was to reduce the cost on a real-life scenario, where the soldiers can attack the medics and the loss of one of these translates into a high cost due to the lack of specialist on this field. To solve the problem, the war simulator offers battlefield sceneries where the player can take the role of the medic and do the tasks given [3]. Unreal Engine allowed the developers to create realistic environments and simulate weathering effects that made the interaction feel realistic. On the other hand, Christian Rubino, from the RMIT University made an investigation about real time optimization in games using Unreal Engine 2. The first and biggest differences between the cinema and videogames were the limitations with the hardware and frames per second (FPS). It is necessary to optimize the environments so the calculations made on real time can be greater than 30 FPS. To solve the problem Unreal Engine offers a complex architecture where is possible to reach high frames with a good aesthetic quality. Using Level of Detail (LOD) simplifies the 3D models depending on the viewing distance of the camera, reducing the number of polygons rendered on the screen [4]. The collisions, which is a system that allows the objects collide with others inside the level are pre-calculated to create a simplified version of the 3D model silhouette [4]. Using Unreal Engine's tools made possible to increase the visual quality of the videogame's environments without having any performance issues.

Besides Unreal Engine, there are other videogame developer tools focused on other technologies like HTML5. For instance, Michael Weeks, from the Georgia's University developed a videogame using HTML5 and Javascript programming languages. The purpose of this project was to show that is possible to create a videogame only using the two technologies mentioned before. The videogame was an action platformer game using 2D concepts. It was in a 2D space where the Tiling textures were used to create the environment. A Tiling textures are used to duplicated them inside the level instead of create a much bigger texture that can take more memory space. Using this made possible to create a 2D type videogame [6]. In addition, there are other technologies than HTML5. For instance, students of Fortaleza's University experimented on developing a videogame called Funcopter in the shortest time possible using the Unity 3D Game Engine. It was designed to run on mobile devices, so the simplicity of the game had an important role in the development and planning stages of the project. With that said, the team only focused in only two stages: Preproduction and Production. During the Preproduction stage, the documentation, storyboards and core concepts of the game were avoided to save production time [7]. They also analyzed games with similar mechanics to take the best out of them and incorporated them into FunCopter. For this case, the simplicity on the development of the game took an important role in the scope of the game. Unity 3D turned out to be a good option to finish the game in 2 days due to it was a game made for mobile devices [7]. There are even shorter games made, Valhalla is one of them and was made under 48 hours. The main core concept to develop these short games is to iterate inside the engine as quick as possible to test the gameplay on real time. Without any documentation or any other idea, the game was evolving using only the Unity Editor that made possible to make changes on the fly and iterate with different solutions to see how the final product would look on real time [8]. This made possible to design the levels inside the engine without any documentation [8]. Therefore, it was possible to create a game in two days using the right tools. In the other side of the spectrum, students from Lisbon University, developed a videogame that can help other students to prevent the bullying in the schools with kids between 10 and 12 years old [9]. The project was made in collaboration with a group of phycologists to get the information of the students and group them according to their psychological profile. The use of this interactive technologies helped to promote specific behaviors in the users who are using it. The game was designed so the users and connect with other phycologists, before entering the level the player will tell his emotional status and other relevant variables to the study case. Finally, the levels simulate a real-time scenario where the player is attacked by other kids and depending on his profile is assigned to do a task inside the classroom [9]. Students from Lisbon University, made a research of how lightning in games can give a richer user experience [10]. During the research, the found techniques that allow the player to submerge into another universe if they are used in the right way. The lights must be placed in a smart way to call the attention of the player, due to technical limitations lights with much higher quality should only be used on places where the player will interact and lights with the lowest quality in shadows should be placed on inaccessible zones. The contrast of colors and illumination placed in important places turned out to be an important aspect to guide the player to the main objective of the game. These techniques resulted to be essential to create a high quality product and a better user experience [10].

III. VERMILLION

For the game development of the Vermillion game stages and areas proposed by the IGDA (International Game Developer Association) were used. Which are: Critical Game Studies, Games and Society, Game Design, Game Programming, Visual Design, Audio Design, Interactive Storytelling, Game Production and Business of Gaming. For purposes of our project, we didn't use the last three development areas that are focused on the business area.

A. Preproduction

Vermillion was developed under the four stages proposed by the IGDA: Preproduction, Production, Testing and Postproduction. To begin with the project, during the stage of Preproduction the OneSheet was designed, this document contains information about the game vision and the features it provides unlike others. Likewise, it serves to find investors that desire to invest on your Project before it begins, as well as seeing if the product has the potential worth the development, being developed based on the production costs. So, we worked on finding references to see previous games and art styles that could fit with our vision of Vermillion.

Based on what was obtained in the first stage, we can create a 3D character to have the vision of what the artistic direction of the game, this helps so the rest of the content can be created in the same style.

B. Production

During Production stage, we create all the content and assembly it inside Unreal Engine. This means we need to put the 3D models in the environment to make it look good and start to improve the C++ classes previously created in the last stage. For this Unreal Engine gives us predefined classes to create the core logic of the game: GameMode, Pawn, Player Controller and HUD.

For the development of map we use 3D models, materials, lights, volumes and Navigation Meshes collision. There are 2 types of 3D models within Unreal Engine: Static Mesh and Skeletal Mesh.
The first is a 3D model containing only visual information and can't move. The Skeletal Meshes contain information of bones and influences they have with the vertices of the 3D model, so they the animations. For Vermillion we Static Meshes to delay the map: rocks, statues, stairs and architectural structures. 3D models need information from the materials also called Shader, containing instructions on how the surface of the 3D model should react when light touch, these materials have color information, roughness, if metal or not and normal 3D model.

C. Testing

During the Testing stage, all the content created in the Production stage was used to test the game and balance the enemy mechanics, the character and their scores. Likewise, it is necessary to continually test the videogame to adjust in an interactive way.

D. Postproduction

In the last stage, Post Production, we put the last details in the games like lights, particle and cinematic effects and sounds. To make the game look good we place three kinds of lights inside the Editor: Static, Stationary and Movable. The Stationary lights are the ones with the highest quality possible but are not able to move, due to the fact that they require a lot of memory, we only place them in areas where the player will interact with the environment. Static lights are used to put lights without increasing the costs on the performance, because of their quality isn't the best they are placed in areas where the player won't interact with the environment.

E. Software management

Thus, the Project had a lifetime of 1000 hours. Then, it is shown how much time we have dedicated of each stage of the project.

As we can appreciate, the Production face was the longest one, taking near 500 man hours which turns to be 50 per cent of the lifetime of the Project cycle. This is because the game needs art and programming. The Preproduction and Testing stages took around 20 percent and the Post Production stage only the last 10 percent.

As a result, we have a computer videogame with high quality visuals and free of errors. Vermillion counts with three types of enemies, four unique abilities and a combat system with levels and player experience.

IV. RESULTS

A. Development

The phases of developing an entertainment software are different from other industries in some particular aspects. In the first place, the development of an entertainment software not only requires engineer's experts in their fields, but are made of different multidisciplinary teams who can combine the art and engineer to create a unique product. The IGDA (International Game Developer Association) proposes different areas and phases to create a videogame.

During the development of our videogame, we checked that the methodology can be effective to make the process faster and increase the quality of the product. For the case of Critical Game Studies and Games and Society, it was very helpful to look for references such as images and different gameplay mechanics from other games to incorporate to Vermillion and create a starting point.

B. User experience

Likewise, to measure if the game has the expected Outcome. We have made a small poll of ten persons to check the quality of the product. Therefore, we decided to measure 3 points with simple questions: the first one, not being more important that the others, is the graphic quality with the question "Is the game visually attractive?", this was asked because when using unreal the idea was to achieve a visually attractive game with the low amount of resources of time we had, afterwards we had gameplay which is an important pillar because when developing a game you must ask yourself how well will the player interact with the game, so to find out we asked the question "Is it playable?"; last but not least is if the game was entertaining, because the usual point of the game is to make the player enjoy themselves while playing a videogame, so we asked the simple question "Is it fun?". This is why we focused in a measurement range from 1 to 5, to be able to quantify them since they were qualitative variables.

We hoped to achieve a combined score of 3.5 across the 3 measurements we propose. In a score from 1 to 5, where 1 was the lowest quality and 5 the highest, it turned out the game was visually attractive but the fun factor needed to improve.

As a result, we got that the majority of the people found the game visually attractive but lacked game content, this is because the scope of the project and the limitations we had as a result of a short life cycle.

In conclusion, we have proven it is possible to achieve our measurement goals getting an average score of 3.53 across the 3 of them, although our game lacks gameplay we try and compensate this by making it visually attractive and use it as the game strongest point.

V. CONCLUSION

Our videogame called Vermillion has served as a study case and validation of a methodology specialized on the four stages to construct a videogame. The stages and areas segmented in this methodology had turned to be beneficial to develop a high quality game in a short period of time. Nevertheless, the use of this methodology can change depending on the type of project that is required as the team size. In our case, an action game with two person size team, needed to adapt to not work in some areas like: Interactive Storytelling, Game Production and Business of Gaming, which they didn't help in the production process. Likewise, the area documents provided by the area of Game Design weren't needed because of the team size. In other words, we can say the use of this methodology is viable for game development in Unreal Engine with limited resources and in a short period.

Finally, the use of Unreal Engine to create a videogame turned out to be very beneficial to get an aesthetic pleasing product in a short period of time. Likewise, the Blueprint system allowed us to prototype the game since the first day and create complex behaviors that could have took longer in other technologies. The engine and the techniques used made possible to create high quality rich graphics and user experiences.

ACKNOWLEDGMENT

We would like to acknowledge and thank Evelin Espinal and Flor Huisa for their support, establishing the four stages methodology and helping us to developing our video game using the methodology.

REFERENCES

[1] F. Zhu (2016, 1) Industry Outlook [Online]. Available: http://fzdschool.com/entertainment -design/industry-out look.(URL)
[2] EPIC (2016, 1) Unreal Engine Features [Online] Available: https://www.unrealengine.com/unreal-engine-4. (URL)
[3] R. Kuskuntla, E. Imsand y J.A. "Drew" Hamilton, "Enhanced Expert Field Medical Training Simulations and their effect on the Modern Combat Life Saver Training Procedures", Auburn University, AL – 36849, 2006, pp 4-5.
[4] C. Rubino "Level design optimization guidelines for game artists using the Epic Games: Unreal Engine 2", RMIT University, 2006, pp 27-30.
[5] G. Hodkinson "Taking animation project learning into the virtual environment" Massey University, 2015, pp 1-2.
[6] M. Weeks "Creating a web-based, 2-D action game in Javascritpt with HTML 5 " Georgia University, 2012, pp 1-6.
[7] D. Valiente de Macedo, M Formico "Experiences with Rapid Mobile Game Development Using Unity Engine", Universidad de Fortaleza, 2011, pp 1-12.
[8] E. Fiske, T. Steinke, M. Lisenbard "Valhalla", Cal Poly State University, 2016, pp1 -2.
[9] C. Raminhos, A. Cláudio, M. Beatriz, S. Carvalhosa, M. Candelas, A. Gaspar "A serious game-based solution to prevent bullying", University of Lisbon, 2013, pp 1-8.
[10] M. El-Nasr, J. Zupko, K. Miron "Intelligent Lightning for a Better Gaming Experience" Penn State University, 2007, pp1-2.
[11] J. Wilson (2015, 1) Tutorial: Physical Based Rendering, And You Can Too! [Online]. Available: http://www.marmoset.co/toolbag/learn/pbr-practice (URL)
[12] E. Espinal and F. Huisa, "Entertainment Software development's process", Universidad Peruana de Ciencias Aplicadas, Lima, Peru, 2015