

The LOST: An Attempt to Combine Film and Game

Qingpu Lou

University of Utah Salt Lake City, USA

* Corresponding Author Email: u1396612@utah.edu

Abstract. The combination of computer graph model and game modeling has become a research hotspot. The paper focuses on developing a linear role-playing game (RPG) called "LOST" using Maya and Unity software. The study aims to detail the development process and provide a reference for future game development using cinematography techniques. The paper begins with an introduction to the background and purpose of the study. This is followed by an analysis of the design and implementation of the game's main features, such as plot, gameplay, models, and computer graphics (CG). The authors emphasize the importance of carefully crafting and polishing the game's main quests to ensure a complete and enjoyable gameplay experience for the player. They also discuss the trade-off between performance and quality in the design of models and the impact of pre-made animations on the game's visual quality and storage space. Finally, the authors evaluate the strengths and weaknesses of the game and suggest areas for improvement, such as adding more content and variety to the combat system. Overall, the paper provides a comprehensive overview of the development of LOST" and serves as a valuable reference for future game development projects.

Keywords: Computer graphics, LOST, game, film, maya and unity.

1. Introduction

Since the founding of the first-ever video game was launched in the oscilloscope of a physical laboratory in 1958, game development has been evolving with the development of technology. Integrating technology from other fields into the development of games has played a crucial role in the advancement of the medium [1].

Generally speaking, the art design of a game can be divided into original art design, special effects design, level design, the seven Cs design and so on. These are not only related to human-computer interaction, operation logic and other technical links, but also closely related to the artistic design of visual elements [2]. From the scene, terrain, character clothing, login interface, operation interface, game props, skill markers, small objects in the game and so on all affect the player's perception and game experience.

On the other hand, different types of game design have different artistic purposes in the design of visual elements such as interface, icon and costume of characters, and these elements will also change with the transformation of the plot of the game [3]. For game art designers, they should not only have the basic principles of computer programming language and graphic graphics knowledge, but also master the content, form and meaning of game visual elements design [4]. In fact, the layout of the screen, color atmosphere and other visual elements in online competitive games are irreplaceable for highlighting the narrative, competitive and immersive content of the game.

Today, the technology of games has entered a new era, with a complete modern video game often consisting primarily of gameplay, story, art, and graphic design. Almost everything about game development is changed. For example, the running logic ranges from development circuit boards to programming, the graphics range from flat abstract graphics to 3D realistic graphics, and the game platform ranges from arcade game consoles to computers and home consoles. However, these three elements have consistently been considered the core components of a game, while the form and vehicle of the game itself have undergone various changes and iterations. As a new form of media, games are often considered the next-generation medium due to their unique combination of narrative and interaction.

This paper explores how the incorporation of film techniques into games can enhance the narrative and emotional expression of the medium. In the past, games have successfully incorporated artistic approaches such as storytelling and music, which have greatly expanded the medium's potential. It is reasonable to believe that incorporating film techniques can take the possibilities of games even further. The following paragraphs will explore how film techniques were used in game development [5]. In the following paragraphs, this article will discuss the overall development framework, i.e., the functional design and the details of each part. After that, we will discuss some practical aspects of the development approach of each part, the software used, the development steps, and techniques. Finally, the parts will be finalized into a complete game, which will then be tested life, analyzed for its strengths and weaknesses, and finally corrected.

2. Game design and implementation

2.1. Game Frame Description

This game development will focus on performance and narrative as a linear RPG for better storytelling. So, first of all, the story to be told will be straightforward. The background is that the protagonist is summoned to another world, a parallel universe to ours, similar but with different laws. The extinction crisis threatens this world called the "Great Filter." However, due to a malfunction in the summoning ritual, the protagonist is summoned to a world covered with sand and without living creatures ten thousand years after the Great Filter [6]. In the goal set, the player is considered to need to survive in this world, explore this world, learn the truth about the "Great Filter," and ultimately find a way to return to the original world. In the environment setting, due to the destruction and decay for 10,000 years, the world has been transformed into a world full of deserts, ruins, wilderness, and swamps, taking into account the overall atmosphere, so the style of the scene is aimed at post-apocalyptic and barren.

To increase the player's immersion while making the plot more vivid, the development of this game will be combined with filmmaking techniques, the use of montage and staging, as well as some of the realistic camera technology practical in the game camera. This paper aims to find a solution for combining cinematic techniques that can be used later in the development of many types of games. The game is based on a BBS short story from several years ago. The story's overall atmosphere resonates with the designer's experience adapting to an unfamiliar culture and environment overseas. The protagonist leaves his original world to travel to this world and walks around in a deserted environment where no one will react to him. It is hoped that the narrative and plot of the game will also resonate with the experiences of people in foreign countries, and based on the above purpose, the game is named "Lost."

The development of "LOST" is based on the production process of conventional 3D RPG video games. This game type mainly consists of the following modules: model, combat, animation, sound, and UI (Fig.1). As the most considerable workload, this part determines the overall visual perception of the game. Combat mainly includes the game's basic attacks, weapons, equipment, skills, health, and props effects, this part of the work is mainly code. Gameplay largely depends on the combat system's quality. Sound includes all the game's sound effects, such as environmental sound effects, interactive sound effects, background music, etc. Good sound effects can help create a sense of atmosphere and immersion. The UI contains the system settings, item details, and other panels in the game, and this part mainly plays a role in helping players better understand the game properties. Among them, for the need for narrative, the part of the animation will be put more effort into showing a more vivid character performance and CG animation. Especially in the CG part, the mainstream method in the market today is to use real-time algorithms to produce in-game CG animation [7]. The benefit of this animation is to save memory to a great extent. However, to better combine film production skills, the CG of this work will use pre-production video, which has the advantage of saving performance in the case of maximum picture effects and using some camera techniques in filming.

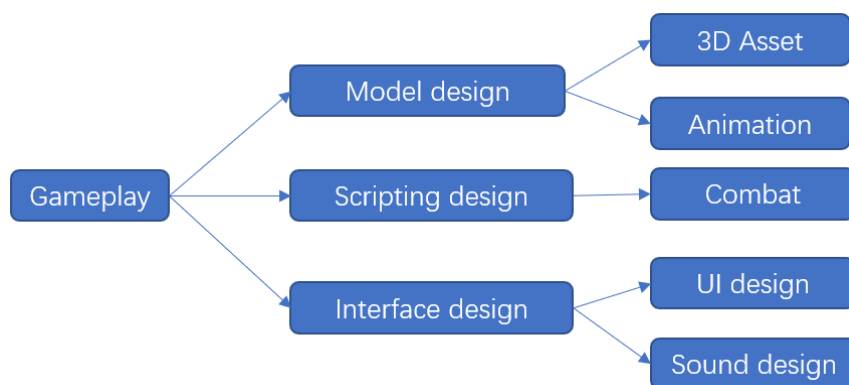


Fig. 1 System structure

LOST runs on Unity's 3D modules, one of the leading game engines. Unity is heavily used in small to medium-scale game development due to its high degree of modularity and compatibility. The developers should write modules for Unity with C#, the game development-specific code language, to customize everything from shaders, pipelines, and all physics particle effects. Even before Unity officially supported ray tracing, custom URP pipelines were used to add support for ray tracing. This article describes the development process for LOST on Unity, where several 3D software packages were used to reduce the production cycle and ensure the final result. It is generally accepted that complete game development involves modeling, animation, environment creation, and scripting, and the subject of this article, LOST, is no exception. These steps will be introduced in the following paragraphs.

2.2. Modelling Design

As the visual body of the game, the creation of the model determines the first impression the overall game gives to the player, ensuring the overall model's versatility and compatibility and reducing the pitfalls of possibilities in different scenarios [8-9]. We will use the current mainstream professional 3D modeling software for production. Autodesk Maya often referred to as Maya, is a 3D computer graphics application that runs on several major systems and is often used to generate 3D assets for film, television, games, and commercials (Fig.2). Maya's relatively weak rendering and the material system is dedicated modeling software. However, she has a robust hard surface modeling capability, and as such, Maya will primarily produce low polygon models in production.

In the second stage, we import the 3D assets created in Maya into Zbrush, a professional model sculpting software from Pixologic (Fig.3). Due to its high performance in high polygon modeling, it is widely used in many fields, such as film, television, and games. Because it is used differently from traditional 3D software, it lowers the threshold of use, making it easier and more efficient to create digital sculptures. It is more suitable for artists who have a technical background [10]. The sculpted model is exported separately as a model and a texture, the latter being utilized in a lower polygon version of the same model, which reduces the need for performance and increases the game's compatibility on different hardware. In the last stage, we imported the previous model into Adobe Substance Painter for surface mapping.

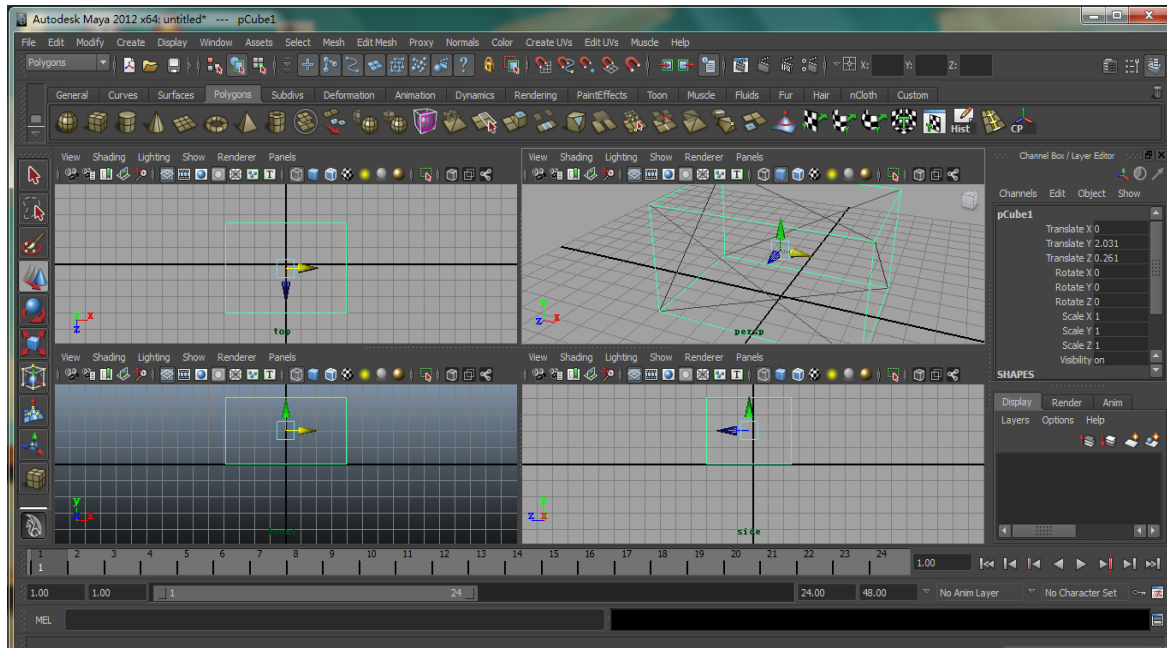


Fig. 2 Maya interface schematic

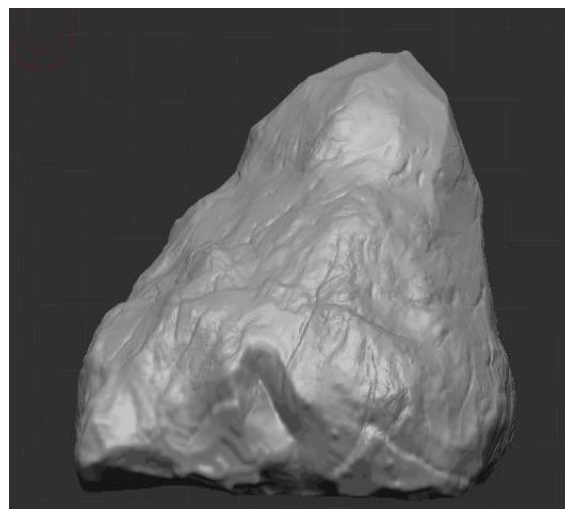


Fig. 3 Stone Made by Zbrush

As a drawing and mapping-based painting software, Substance Painter helps texture artists to add fine, high-precision textures to their models without actually sculpting polygonal details and is therefore widely used in the production of 3D assets in directions such as games and product displays [11]. As this is a game-oriented production, we will only export Diffuse, Normal, and Height maps once the textures have been painted.

2.3. Animation

There are currently two solutions to the problem of in-game animation, each with its advantages and disadvantages, the first of which is an utterly hand-made animation by animators, which has the advantage of being relatively surreal and exaggerated while allowing for many physically impossible movements. An excellent example is the Call of Duty series, where hand-made animations are still used in many modern games because of their ability to engage the player's passion. However, even with such great results, manual animation is slowly falling out of the mainstream production workflow due to two significant drawbacks: manual animation has a long workflow [12]. A minute of animation requires around 360 keyframes; without considering interpolation time, a single keyframe takes a skilled animator around 6-7 minutes. This means that one minute of animation requires at least 36-42 hours of pure work by the animator, which is almost a whole week. The second

point is that high-quality hand braking is relatively expensive and requires experienced artistic creators to ensure realism, which raises costs. The prevailing solution is motion capture combined with an animator's correction.

Although this solution is not suitable for some surreal movements, it reduces the production cycle and costs to a great extent. So, for two reasons, LOST finally adopted the motion capture solution provided by Rokoko Studio. Firstly, all of LOST's character models were built using only procedural automatic skeletons, without rigging, which made manual animation relatively complex. Secondly, the production cycle was considered limited, and there needed to be more time to ensure the quality of the manual animation. Binary file format, which is a popular 3D file format that Kaydara initially developed for MotionBuilder, was acquired by Autodesk Inc in 2006. It is now one of the leading 3D exchange formats used by many 3D tools. It is generally used to provide easy-to-manage pre-made animations for game engines and renderers due to its inclusion of mapping and recording of animations based on point coordinates.

2.4. Environment Creation

Regarding terrain creation, LOST is a linear semi-open world game where continuous world terrain is not required. Several minor natural terrains are sufficient. Automatic procedural terrain generation was therefore considered non-essential. With this in mind, the designers felt the environment should be modeled in a fine-level design manner [13]. In the end, we chose World Creator to create our custom terrain. World Creator is a professional stand-alone terrain creation tool to create the most realistic terrain data for 3D assets. It empowers the user with the most advanced procedural techniques available nowadays. Furthermore, desolate, the designers used World Creator to create many deserts and mountainous terrain. Some of the better-known scenes were used as references in the production, such as the valleys of Utah, the deserts of northern Africa, and the wilderness of western China (Fig.4).

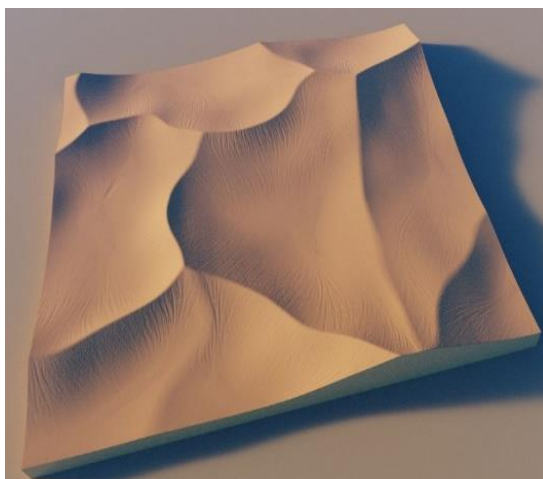


Fig. 4 Desert Made with World Creator

2.5. Scripting

As a game developed on Unity, C# is used as the primary language in LOST for best results [14]. The first is the battle system, where animations are connected to nodes in a state machine, with scripts made in C# to help determine the battle state, calculate data in battle and build the battle system, such as collision boxes. The second is scripting to build the game's UI, including information and plot systems, and to interact with player and game data. Script development can also be used to craft enemy AI systems in the form of state machines to improve the playability and difficulty of the game and help players have more.

2.6. CGI

In the CGI section, as mentioned earlier, we will be using pre-made animations for the CG due to the need for a combination of effects and cinematography [15]. As there is no high demand for real-time performance, the rendering will use the aforementioned high-quality texture mapping and high polygon models to get the best visual effect (Fig.5). Cinema 4D will be used as the rendering software, a 3D graphics software developed by the German company Maxon Computer, known for its high computing speed and powerful rendering plug-ins.

Cinema 4D is used in various applications in advertising, film, industrial design, Etc. Regarding the choice of the renderer, the previous solution, Octane Render, a GPU renderer, only maximized rendering efficiency and reduced rendering cycles [14-15]. It, therefore, did not use realistic optical physics calculations. An Arnold Render, developed by Autodesk and heavily used in the film and television industry, was introduced into the development process to avoid lighting errors and apparent mistakes. This is a physical CPU renderer that uses multiple renderings to minimize the effects of noise. In the production process, the high-precision models and motion capture data from the previous sections are imported into Cinema 4D. Once the scene and character bodies, light sources, and skyboxes are ready, camera movements are created and rendered based on the camera view.



Fig. 5 Colored Stone CG Preview (Not actual game scene).

2.7. Exportation

After completing all the steps, including model import, animation, asset import, and script binding. It is time to start exporting the release files. First, we will integrate the system and export the exe file as a build, which is a generic self-running file for Windows, then select a folder to export to, after exporting the project is complete and can be run on Windows, Linus, and other platforms by opening the exe file.

3. Conclusion

Overall, as a linear RPG, LOST has a relatively short and straightforward plot line. All of the main quests have been carefully crafted, and details have been added to ensure a complete gameplay experience for the player through these episodes. In terms of gameplay, the producers have forgone adding a lot of mechanics and game design and instead focused on polishing the quality of the existing mechanics. These changes have improved the feedback and ease of the combat system for the player, making it easier for the player to get up and running quickly. In contrast, more combat skills and mechanics will be unlocked as the player progresses through the game, and chapters are unlocked, making it easier for the player to experience new things as they play and not get bored with the combat system. In terms of the models, the creators have gone to the limits of performance-to-quality ratio,

providing a good level of detail in the models' textures while ensuring that most devices can run them, allowing everyone to enjoy the content smoothly. As for the CG, although the pre-made animations make the game a considerable package, they give the player a stable and high visual quality animation. However, LOST has many drawbacks, including minimal game content and multiple playthroughs. The game needs a larger total map to explore and collect, so once the player has finished all the main story and a small number of side stories, they have little else to play. The second has a large package that takes up considerable storage space. Another is the limited combat experience. Being a small-volume game, LOST has only one set of combat moves and skill sets, and even though more mechanics will be slowly unlocked, the single combat style tends to be exhausting.

Overall, this paper develops LOST, a linear RPG based on Maya and Unity software, detailing the development process, completing the Maya model animation design and C# scripting process and CG rendering in C4D, achieving the game development goals and providing a reference for future game development using cinematography techniques.

References

- [1] Smith, R. "The disruptive potential of game technologies". Research-Technology Management, 2007, 50(2), pp. 57-64.
- [2] Lin, Ting-Chieh, Zen-Chung Shih, and Yu-Ting Tsai. "Cinematic Camera Control in 3D Computer Games." International Conference in Central Europe on Computer Graphics and Visualization, 2004, pp.322-331.
- [3] Mathews, Jana. "Cinematic thanatourism and the purloined past: The "game of Thrones effect" and the effect of game of Thrones on history." Virtual Dark Tourism: Ghost Roads 2018, pp. 89-112.
- [4] Gurevitch, Leon. "Cinema, Video, Game: Astonishing Aesthetics and the Cinematic 'Future' of Computer Graphics' Past." Cinematicity in Media History 2013, pp. 173-195.
- [5] Mitgutsch, Konstantin, and Narda Alvarado. "Purposeful by design? A serious game design assessment framework." Proceedings of the International Conference on the foundations of digital games. 2012.
- [6] Balson, F., Aitken, S., Hillenbrand, P., Vu, T., Ward, M., Jablonski, J., & Rabb, A. "The past, present and future of the video game cinematic: a look at the evolution of the video game cinematic and its impact on the industry". In ACM Special Interest Group on Computer Graphics 2018, pp. 1-2.
- [7] Girina I. "Video game Mise-En-Scene remediation of cinematic codes in video games", Interactive Storytelling: 6th International Conference, 2013, pp.45-54.
- [8] Girina I. "Cinematic games: the aesthetic influence of cinema on video games". University of Warwick, 2015.
- [9] Dickey, Michele D. "Game design narrative for learning: Appropriating adventure game design narrative devices and techniques for the design of interactive learning environments." Educational technology research and development 2006, pp. 245-263.
- [10] Jiang, J. H., Feng, F., Rosen, P. E., Fahy, K. A., Das, P., Obacz, P., ... & Zhu, Z. H. "Avoiding the Great Filter: Predicting the Timeline for Humanity to Reach Kardashev Type I Civilization. Galaxies", 2022, 10(3), 68.
- [11] Akenine-Mo, T., Haines, E., & Hoffman, N. "Real-time rendering". AK Peters/crc Press, 2019.
- [12] Faria, A. J., Hutchinson, D., Wellington, W. J., & Gold, S. "Developments in business gaming: A review of the past 40 years". Simulation & gaming, 2009, 40(4), pp. 464-487.
- [13] Ilett, D. Introduction to Shaders in Unity. In Building Quality Shaders for Unity: Using Shader Graphs and HLSL Shaders 2022, pp. 1-8.
- [14] Гаранін О М, Моїсєєнко Н В. "Adaptive artificial intelligence in RPG-game on the Unity game engine" Computer Science & Software Engineering: Proceedings of the 1st Student 2018 (2292), pp.143-150.
- [15] Kim S L, Suk H J, Kang J H, et al. "Using Unity 3D to facilitate mobile augmented reality game development", 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 21-26.