Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(3):785-791

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Strategy research on the performance optimization of 3D mobile game development based on Unity

Peng Hu and Kai Zhu

Oxbridge College, Kunming University of Science and Technology, Yunnan, China

---

ABSTRACT

With the wide application of 3D games and the improvement of smart phone performance, the competition of the R & D and market of 3D mobile game products is becoming increasingly furious. Compared to ordinary computers, the hardware resource of mobile phone is greatly limited, so how to develop a 3D mobile game that can match the performance of PC platform 3D game has become a hot-topic in the field of mobile phone application. This essay analyzed 3D mobile game development framework based on Unity platform and Unity script execution flowchart, and with the rich development experience, this essay proposed optimization strategy and presented real examples ---- optimization experiment of 3D role-play mobile game, and further proves its feasibility and validity.

Key words: Unity; 3D mobile game; performance; optimization strategy

---

INTRODUCTION

With the increasing number of mobile phone users nowadays, the design of mobile phone is becoming more and more complicated, and mobile phone services are much more diversified. However, mobile application is still at the beginning stage with much improvement space. One of the many defects is the majority of games are 2D scrolling picture, which is boring and unappealing to game players. Given the fact that mobile phone functions are more powerful and longer stand-by time, playing 3D game through mobile phone is within the reach of game players. Analysts believe, the process of mobile games changing from 2D to 3D will be rapid, just like the process of PC video card changing from 2D to 3D.

Although the general market demands are switching from 2D to 3D, considering the performance limitation of entry-level and mid-range mobile phones which take the majority of market share, it has greater potential to research and develop 3D game engine.

THE NECESSITY OF 3D MOBILE GAME OPTIMIZATION

Though smart phones are developing rapidly these years, there is still a large gap compared to PC, mainly in the aspects of limitations in hardware resources. CPU and RAM of mobile phones are far behind the mainstream configuration of PC, the key measurement, CPU, only equals to the level of PC 7 years ago. So it is extremely difficult to realize 3D technology whose computation volume and memory consumption are both high. Unity is an integrated 3D game development tool and game engine set, which includes various aspects such as image, audio, physics and network. Unity adopts MONO as virtual machine of script engine, and uses C# or a language resembles JavaScript greatly. It can be published the game on Windows, Mac, Wii, iPhone, Windows phone 8 and Android platform, so it is the most popular choice for 3D game developers.

The purpose of this essay is to analyze the key technologies of 3D mobile game development based on Unity and propose an optimization strategy. At the end of this essay, a real example is presented to illustrate the validity of this strategy, and this strategy will enable 3D game to operate on mobile phone.

DEVELOPMENT FRAMEWORK

This essay adopts Unity as middleware, combined with 3D mobile interactive system development frameworks of PHOTOSHOP, 3DS MAX and ECLIPSE. This development framework is divided into five development modules: Planning, Modeling, Programming, Interaction and Testing. There is other detailed development procedures under corresponding module, as shown in Chart.1.

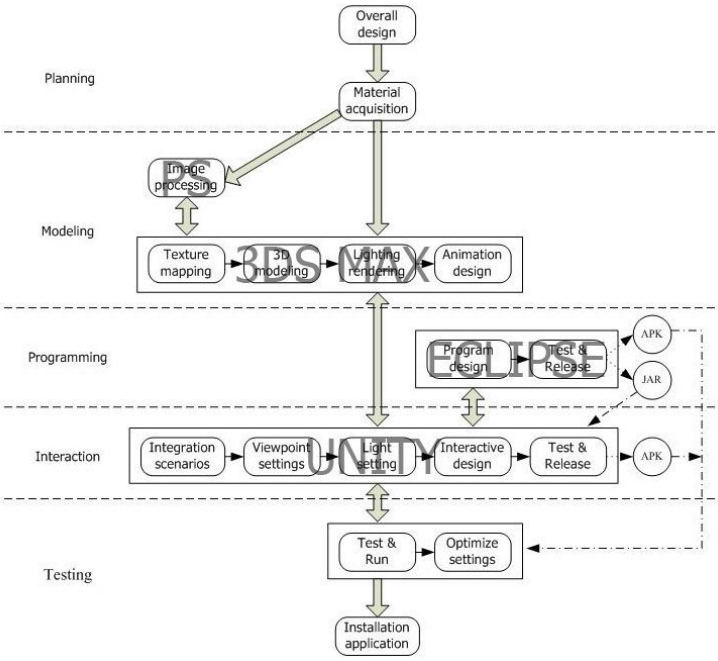


Chart.1. Development Framework

The interaction of Unity all used script, if the purpose is to optimize 3D mobile interactive system of Unity, then the execution procedure of script must be understood. According to personal development experience and relevant knowledge, the following execution flow chart has been drawn, as illustrated in Chart.2.

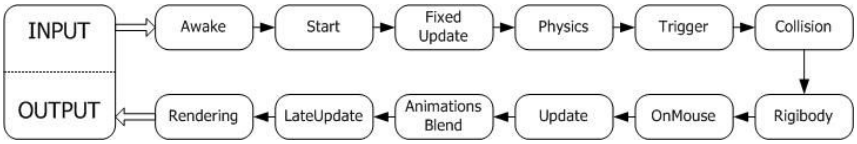


Chart.2. Flow Chart of Unity Script Execution

KEY TECHNOLOGY OPTIMIZATION STRATEGY AND EXPERIMENT ANALYSIS----3D RPG MOBILE GAME AS AN EXAMPLE

It is obvious from the analysis of development framework and Unity script execution flow chart, that the main technologies which influence the operational effectiveness of mobile phone are 3D modeling, collision detection, particle effects, and character animation and interaction code. This essay takes a 3D role play mobile game as an example, which is developed by development framework mentioned above, and then the essay analyzes each key technology, proposes corresponding optimization strategy, and proves the effectiveness of optimization strategy in the end.

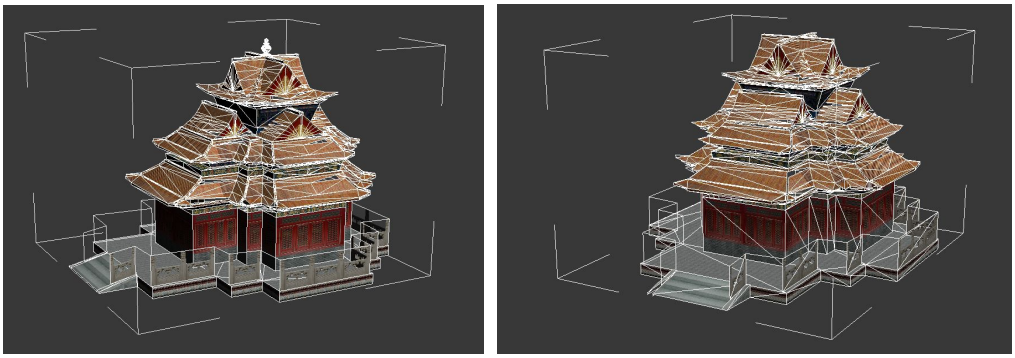
Optimization Strategy of Key Technologies

3D Modeling: The quality of 3D modeling directly affects the performance of 3D game. In order to create a realistic 3D world, the requirement on modeling must be sky high, but it will take up large memory footprint, which will affect the overall operation efficiency of the game. So how to find balance between model precision and 3D performance becomes a crucial issue to be solved. The number of polygons in the scene is a key parameter of measuring model precision.

The models used in 3D mobile game scene should avoid adopting models with high precision. If high precision 3D models have to be used, the solution can be baking the model and integrating lighting, high-lighting, shadow, and

concave and convex into one NormalMap, and adopting in low precision models. In this way, the fidelity of the model is enhanced, the number of polygons is greatly reduced and the efficiency of mobile phone is improved.

The model used should have as few polygons as possible, and the overlapping part of objects as well as invisible part of the scene can be simplified. Using 3dmax built-in Optimize order, MultiRes advanced optimization settings and Polygon Cruncher plug-in can all reduce the number of polygons without compromising appearance look. Take the pagoda model of this game as an example. It has 20,000 polygons. If reducing its polygon number into 10,000 through optimization, the appearance look changes little. The effect is shown in Figure.1.



20000Mesh **10000Mesh**

Figure.1.3D Model Look Before & After Optimization

Character Animation: Character and animation are the most basic elements of games, which directly interact with game players. Although the technology of creating character and animation in games is advancing in a daily basis, the ultimate goal stays the same ---- the pursuit of realistic character and animation. The character animation system of Unity is supported by character animation file and skeletal animation of fbx format imported by 3DS MAX or MAYA. The number of character, skeleton and the complexity of character animation will affect the performance of the game.

The performance of the game will be improved if the number of character model faces and skeleton system is reduced. Furthermore, character animation can be presented by sequence frame. 3DS MAX can be used to export 2D character animation, and then control the character animation of the game with frame animation system of Unity. The result is time of animation production time, but the resources of the game can be saved, the utilization rate of memory storage will be greatly increased and the performance of the game will be improved. How to control attack animation script using frame animation, as shown in Figure.2.

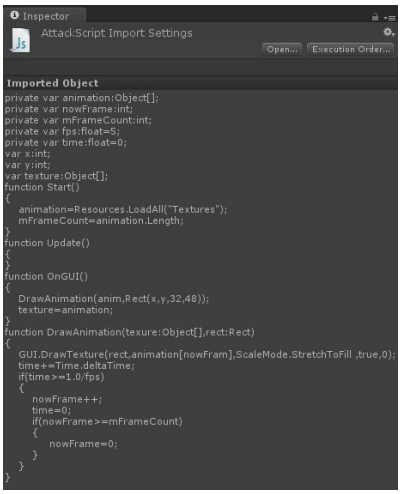


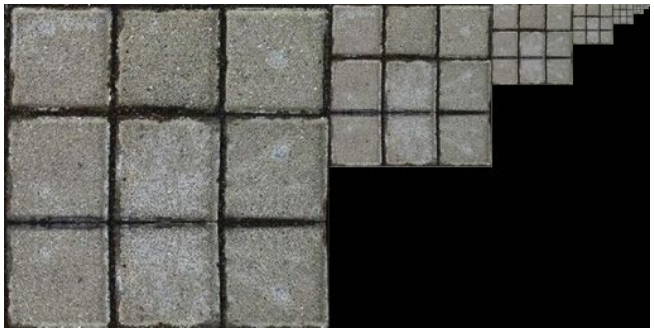
Figure.2. Frame Animation Controlling Script (Attack Animation)

Texture mapping: The stand or fall of a 3D scene picture, 40% depends on the model, 60% depends on material and texture. Thus, in the game pictures material and texture play an important role. But, in a 3D game, due to the amount of data of material and texture is very huge, scene and role of the texture data volume is very large, which

will have great influence on performance of the game.

Texture mapping used should be as far as possible in the texture merged without affecting the image quality, thus reduce the number of Texture mapping. This game not only used traditional mapping types such as PSD, PNG, JPG and TGA, but also other mapping types like Lightmapping(as shown in Figure.3.(a)), which can save the lighting information of the objects onto texture and render in real-time instead of lighting calculation; Mipmapping (as shown in Figure.3.(b)), by setting different Mipmapping level, different presenting detail of objects at different distances can be controlled. This optimization strategy can reduce the rendering cost, and at the same time present the fidelity of the game.

(a)Lightmapping



(b)Mipmapping

Figure.3.Texture mapping

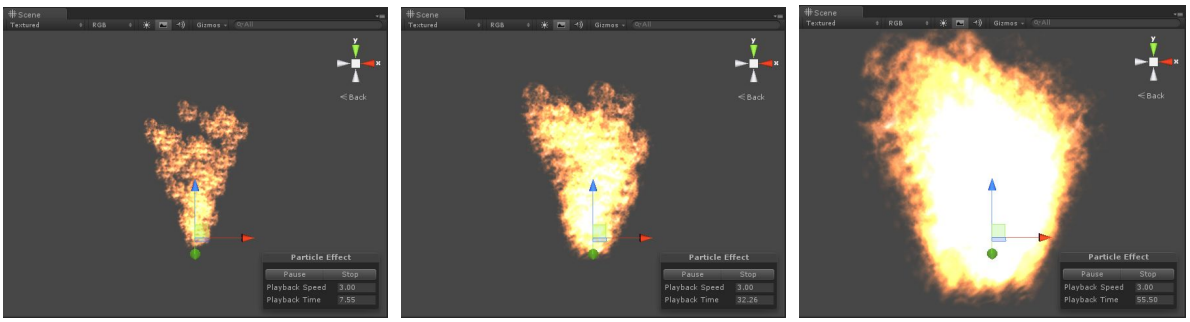
Collision detection: Collision detection is an important part of the interaction function of 3D games. According to the result of the collision detection, by triggering different interaction effect, game system gives timely feedback, which can make the players get more vivid and real game experience. While in the game physics engine is used to calculate collisions between objects, if the calculation is relatively complex, which leads to increased responding time and the overall system may slow down.

AI enemies in this game, for example, the enemy will automatically detect the location of the protagonist and approach. When collision detection occurs, the enemy will attack. At the beginning of the original game, it will appear automatically every 5 seconds one enemy. But when the enemy number is up to 20, the game can not run almost. It really shows that complex physics calculation has a greater influence on the performance of the game. If set the maximum number of enemy for six, when the enemy dies automatically appear new enemies, the performance of ascension is inevitable. Running effect is shown in Figure.4.

Figure.4. Optimized battle effect

Particle effects: In the game, in order to truly simulate snow and rain, mist and dust, and other natural phenomena, particle system must be used. But as a result of particle system is a real-time change, which needs to consume a lot of system resources. Especially when increasing the particle emitter quantity or particle number, it will not be able to enable the game to run smoothly, appear even frame drop.

In the case of the flame in the game, produced by Unity's own Particle System, when the number of particles is set to 25, the game will be stuck at runtime. If using LOD as its optimization, when the particle system is far away from the current perspective, the number of particle generator can be appropriately modified. The effect also accords with the law of the different distance to produce different effect in the real world. LOD particle effect is shown in Figure.5.



Vision(Particle number:5) **Middle(Particle number:10)** **Close(Particle number:20)**

Figure.5. LOD particle system of the flame

Interaction code: Unity can be put in the C# and JAVA as script, it provides a possible code optimization. For example, there is a number of logic in Unity script that is placed in the Update(), and the frequency of execution of the Update() is carried out in accordance with the different frame rate. If too much logic in the Update(), can cause a lot of calculation, so as to reduce the frame rate and the execution efficiency.

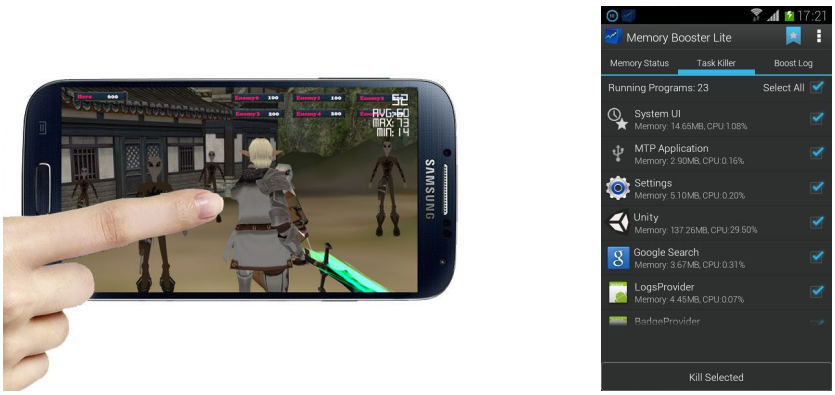
In this mobile game code, for example, using StartCoroutine() instead of the Update(), can significantly improve the efficiency of the game and simplify the code. In addition, in the script, use the following method can optimize itself. Such as using var to define more GameObject, using less as far as possible to Find command, and so on.

Comprehensive experimental analysis---3D role playing mobile game as an example

Operating environment: Samsung GALAXY S4, Operating system: Android OS 4.2, The core count: Dual quad-core, CPU type: Samsung Exynos 5410, CPU Frequency: 1638MHz,GPU type: Imagination PowerVR SGX544 MP3, RAM Capacity: 2GB, ROM Capacity: 16GB

Testing tool: FPS Meter, Memory Booster

Experimental method: Run the original mobile phone games running at the same time relevant testing software; Record the consumption of overhead, namely loading time, run to the specific scene of frame frequency (as shown in Figure.6.(a)), the amount of memory footprint and CPU share (as shown in Figure.6.(b)) and the APK size; Similarly, to run mobile phone games of different optimization scheme, record the corresponding overhead. The results are shown in Table.1.



(a)FPS Meter **(b) Memory Booster**

Figure.6. Testing tool

Table.1. Test data table of using different optimization scheme

Items	3D Modeling	Character animation	Texture mapping	Collision detection	particle effects	Interaction code	loading time (S)	Memory footprint (M)	CPU share (%)	Frame frequency (FPS)	APK size (M)
Original game							20	285	32	35	98
Optimization scheme 1	√						9	163	17	60	53
Optimization scheme 2		√					11	187	21	54	71
Optimization scheme 3			√				13	204	23	49	64
Optimization scheme 4				√			16	226	26	45	60
Optimization scheme 5					√		18	239	27	41	57
Optimization scheme 6						√	19	261	30	39	79
Optimization scheme 7	√	√	√	√	√	√	7	136	13	72	50

By the experimental data can be seen in Table 1, the optimized performance than the original game had different degrees of ascension. The optimized scheme 1, which is 3DModeling, won the best performance improvement effect. Compared with before optimization, the loading time reduced by 55%; memory footprint reduces about 31.7%;CPU share fell by 15%;Frame rate increase of 71.4%; 45.9% reduction in the size of the mobile game APK file. In addition, compared to the related data as you can see, in addition to the APK size, the optimizing scheme from 1 to 6, the optimization performance decline gradually, shows that its impact factor also showed a trend of decline. Therefore, in the process of 3D mobile game development should give top priority to the optimization of Modeling.

If using comprehensive optimization scheme, which is optimization scheme7, the optimized loading time compared to the original one reduced by 13 seconds (35%);the memory footprint reduced by 149 MB (52.3%);the CPU share plumped to 29%, decreased 17.56%;frame frequency almost doubled to 72 FPS; the APK size reduced to 48 MB, which is about half of original cost.

CONCLUSION

3D mobile game performance optimization process is actually the process of balancing game picture quality and hardware performance, not simply reducing the game scene, but to adopt effective optimization strategy. Especially the technology being used, such as proper textures for reducing model complexity, merge, setting up reasonable game complexity, these strategies will not significantly reduce graphical quality of the game under the premise of greatly increased game performance.

Thus it can be seen that in the process of 3D mobile game optimization development, the traditional software engineering theory cannot satisfy the continuing changing market demands. It needs to learn from successful experience in PC platform 3D game development, and more importantly, it needs to be aware that the smartphone resource is extremely limited, so the top priority is to find a more effective way to optimize 3D mobile game development.

Acknowledgments
The authors wish to thank the foundation of Scientific research fund project of the department of education in Yunnan Province of China for contract 2011C009 and the research foundation of Oxbridge College, Kunming University of Science and Technology for contract 2011C009, under which the present work was possible.

REFERENCES

[1] Ryan Henson Creighton. *Unity 3D Game Development by Example Beginner's Guide,* p.42-45, **2010**.
[2] James Sugrue. *IOS 5 game development*, August **2012**.
[3] Cheng Ming-zhi. *Unity game development technology*, p.75-79, June **2012**.
[4] Xuan Yu-Song. *Unity3D game development*, p.101-103, June **2012**.
[5] Shi Xiao-ming,Michelle Menard. *Unity game development practice*, p.10-13, April **2012**.
[6] Zhao Ke-xin. *Several optimization Suggestions of using Unity to rapidly develop high quality games*, p.5-8, **2011**.
[7] SU Zhi-tong,SHI Shao-kun, LI Jin-hong. *Computer Engineering and Design*, v31, n 7, p.1631-1634, **2010**.

[8] Peng G,He Y, Sun Y, etal. *Three-Dimensional Game Modeling and Design Research Based on 3Dmax Software ET Intelligent Transport Systems*, Advances in Computer Science, Environment, Ecoinformatics, and Education, **2011**.