Analysis Optimization of Game "X" Performance on The Android Platform

William Tan[1], Riwinoto[2]

ivietwill@gmail.com[1], riwinoto@gmail.com[2]

Informatics Engineering, Batam State Polytechnic, Batam

Abstract. The preliminary research entitled "Game Prototype Performance Analysis on The Android Platform" stated that the game "X" (The name was disguised for development purposes), was not optimized very well. The game needs to be researched further to improve the process of optimization and to find the result based on the conclusion from the preliminary research, which states that the frame from the result of the analysis should reach 60 fps. In this research, there will be many different types of prototype results, with each of them having different types of optimization. There are the scripting process, garbage collection, and rendering. This research focuses on performance which improves the speed of the game and from the result of the research, there will be 4 revisions for having an optimized state of performance that can be used for further development.

Keywords: Performance, Mobile Game, Unity Profiler, Unity 3D, Optimization

1 Introduction

Mobile gaming is known for its portability and having a characteristic that motivates continuous play and spending, and has increased significantly during this decade [1]. Nowadays, the mobile device is supported in many ways that games app can be downloaded for free on Google Play and App Store. Mobile games are in great demand by users at this time. The Mobile games industry hit the highest point in 2020 which generated about 77.2 billion USD in revenue and represents 49% of the global games. [2].

Batam State Polytechnic is having collaboration with Zettamind Studios since 2020 by creating an industrial-scale game for 2 years. The game is still in the development stage called "X". The name was disguised due to the development process and confidentiality of the company. The Author takes part as a programmer in this project and there is one important aspect of this development which is game optimization. This project has reached the prototyping stages. In this stage, the games need to be optimized before they can go to the next stages. Game "X" is a 2.5D platformer that tells a story about an adventure in a micro world with friends that fills the part of the game. There are many interesting adventures with a series of puzzles and obstacles such as collecting points with having a tainted environment because of waste which creates changes to the surrounding. This game is having 3D objects by using a

ICAE 2022, October 05, Batam, Indonesia

DOI 10.4108/eai.5-10-2022.2326588

2D axis screen which is the type of 2.5D platformer [3]. Creating the game can provide knowledge and have positive a impact on the creator and the player with interactive entertainment [4]. The purpose of this game is to collect the objects of the mission from the story. But to collect the objects, there are obstacles and puzzles in the game.

Game "X" has reached the optimization stage of development. The objects were ready in places at the scene. After that, the next stage is to test the result of the prototype on a mobile platform. The process was made using a game engine called Unity. Unity engine is a tool that was developed by Unity Technologies, and launched in 2005 until now, which is becoming one of the most used by the game developer in the world [5]. When discussing optimization, two parts that can be researched when talking about optimization are performance and visual experience [6]. In this research, the scope of research is to focus on the aspect of performance. Performance is an evaluation of all product software and is counted as a set of scores to know how much a device can process. Optimization is the process of strengthening its functionality, to make the game more efficient and effective. Optimization of performance is important because of the target platforms on mobile devices. Mobile devices are having lower specifications than other platforms such as PC [7].

In Unity, there is a powerful tool to show the performance called Unity profiler. Unity profiler works to reduce the finding of bad performance by giving a statistical result. [8] Game "X" was not optimized when checked with the mobile device. The statement came from the preliminary research entitled "Game Prototype Performance Analysis on The Android Platform" [9], which concluded that the game "X" is having a bad performance and needed to be optimized when having a test on the mobile device with 4GB RAM (Random Access Memory). In the market industry gaming, analysis of device specification from the mobile device is needed to create an optimization target of performance in the gaming device. The research used the samples from Swiss countries with specifications of RAM used by the users and found that they didn't use the mobile devices under 2GB of RAM [10]. The focus of this research is to have an optimized gameplay performance by improving the smoothness of the game "X" by its mechanical state. To measure the optimization cycles is by checking the previous versions of the prototype, the value of frames per second, batches, and triangles/polygons in terms of CPU usage, rendering, and memory usage [11]. The results of the prototype from the preliminary research which is developed and taken the data from the results of profiling.

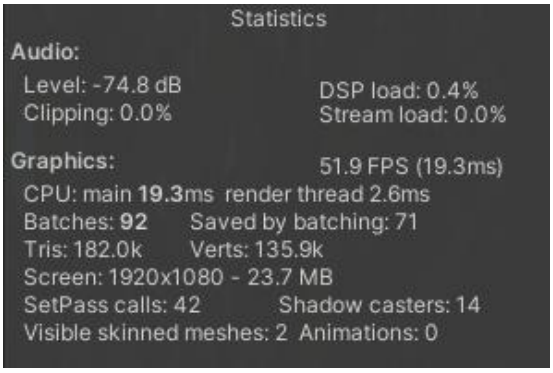


Fig. 1. Statistic Prototype

Statistic for game testing using Unity engine shows the value of tris about 182k and the verts 135.9k as having 92 batches. From those three scores, can be set a parameter of research from the unoptimized version or prototype I. after that, the process will be tested on a mobile device by exporting the game into an APKfile.

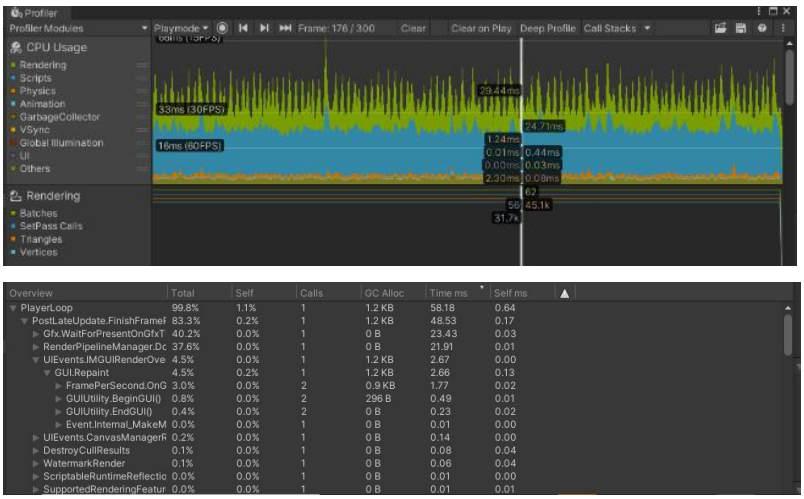


Fig. 2. Profiler Prototype I

From Figure 2 above, it can be shown the results of performance by testing using the Unity profiler on the mobile device. There are a lot of scoring aspects from the rendering, scripting, physics, animation, GC, Vsync, GI, UI, and others. That matter explains a specific performance process of a device. From the data profiler, we can take data as a reference for doing optimization. For that, the result will be shown in Table 1.

This Word document can be used as a template for papers to be published in EAI Core Proceedings. Follow the text for further instructions on text formatting, tables, figures, citations and references.

Table 1. Data Prototype I

CPU Usage	Output Score	Optimized
Rendering	26.20 ms	No
Scripting	23.82 ms	No
Physics	1.19 ms	Yes
Animation	0.45 ms	Yes
Garbage Collection	0.01 ms	No
Vysnc	0.00 ms	Yes
Global Illumination	0.01 ms	Yes
UI	0.03 ms	Yes
Others	2.20 ms	Yes

By having a parameter from the preliminary research using the profiler, it is stated that the rendering and scripting process was not optimized well where the results each are having

29.44 ms, and 24.71 ms. From the total results, it is having 58.18 ms to process all the CPU usage. By having an optimized performance, the total score of CPU usage needs to stay at 16.8 ms. From the scripting usage, it was founded that there is a GC Alloc was created from the script called "FramePerSecond", by having 0.9 kb per frame. So, those three scores will become the primary targets of this research.

Using a parameter from the profiler, the score for rendering and scripting wasn't optimized well with a total of 58.18 ms. For well-optimized game, it needs to be under 16.8 ms for the best performance. For the scripting process, there is a process that creates garbage that comes from the script "FramePerSecond", by 0.9 kb per frame. So, the scope of research will be on those three. The target of this research is to make the optimizing scripting and rendering have a score under 16.8 ms, and for the GC Alloc to have the minimum amount of garbage, and doesn't count for every frame, to be well optimized.

2 Research Method

The method of Analysis optimization performance in this research was measured with the observation method and execution method using the revision optimization cycle. The analysis of this research is to find the optimized results from the preliminary research which stated that the result of prototype I, was not optimized. To optimize the product, dividing into more specifics parts is the better option to optimize the results. There are flows that needed to be done in this research which can be seen in the figure below.

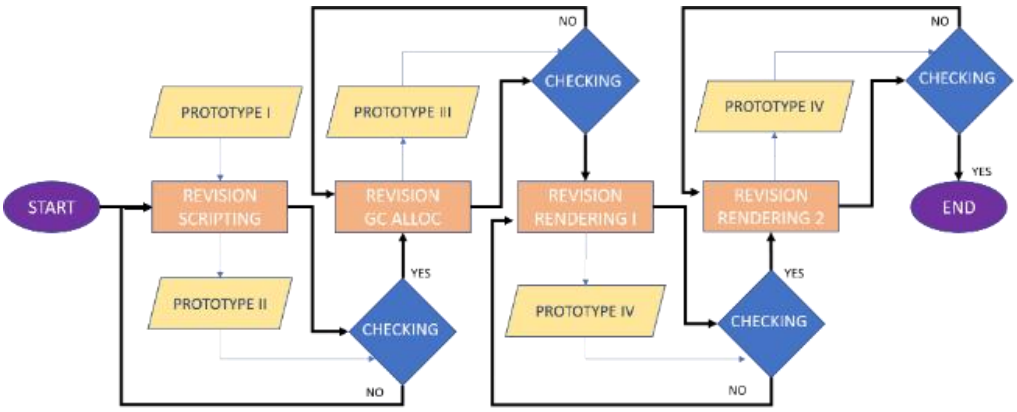


Fig. 3. Revision Optimization Cycle

From the figure above, there are are five steps that needed to be done starts from optimizing scripts, check *gc alloc*, focus on rendering I, and rendering II. It is better to start with sequentially to make sure that the problem was not from back-end which is difficult in the later parts of optimization. By using this flow, it taken using the methods from the preliminary research which called Unity Optimization Cycle[12] which deepened the process and creates Revision cycle by having 5 stages of prototyping, starting from prototype II, optimized on the scripting process. Prototype III will be focused on garbage allocation.

Prototype IV will be focused on the first rendering, and the last, prototype V will be the second rendering. This method will collect performance data and go to the next stage. Every prototyping that is created, there will be a checking phase with the observation method to compare with the previous research, based on the conclusion from the previous research. Using Qualitative methods emphasize observing phenomena and researching more on the substance meaning of the phenomenon [13]. If the score has reached the optimized target, then the next step of optimization can be continued towards the next stage, and if it's not, then the process will repeat at the revision step until it goes into the optimized score.

2.1 Revision Scripting & Rendering

Two aspects that cause the poor performance of the device and that is Rendering and Scripting. The result from the prototype has a total score of 53.91 ms or equal to 15-20 FPS. Humans need 24 FPS or about 41.75 ms minimum to see a moving object to see it smooth [14]. The result of prototyping on a mobile device will be truly seen when testing with playing on a mobile device, that the game doesn't go well from the performance side. There is a need to standardize the usage of the research, and that was supported by Julio Ardyanto from Dreams Studio, which refers that FPS from a mobile device needs to have 60 fps for performance [15]. So, the target of this research is to create a game that can play with 60 FPS or 16.7 ms.

From the data that is shown in the table above, there is a need for revision and checking of the inefficient part to increase performance. The first thing to do is to optimize scripting. Unity in default makes the device work on 30 fps, and to increase to 60 fps, there is a need to add a function code to increase the score from 30 fps to 60 fps. But adding the script doesn't mean that the score automatically increases to 60 fps. The usage of object pooling and reducing the garbage can increase the quality of performance from scripting. Allocation of memory is also one of the research that can increase the performance of scripting. The example of the problems are:

```
1. Debug.Log( "Test" + "Hello")
```

Use "`System.String.Empty`" when we want to delete the result of the object, and do not use " " that may create garbage because of the string is still kept in the memory and no deleted.

2. Calling mode GUI (Unity GUI)

Cause the increase in performance. It's better not to use this calling often or it may increase the work performance.

3. Class and Struct

```
Foo foo = new Foo();

MyFunction(foo);
```

Foo is a class when the object is always together for a long time, as Struct for the object is when the object is having an ephemeral characteristic. One of the cases for the object that may use struct better is for the usage of Vector3. If it was created into a class, then the work progress will not become optimal.

4. Application for the mobile platform as default usually makes the device work on 30 FPS. To create 60 FPS, there is a need to add some function to increase the frame up to 60 FPS.

The next step is to do checking to see if the scripting has been optimized well, if it is optimized then the process can continue to the next step and that is garbage collection. Garbage collection for the game “X” comes from the scripting process. Garbage collection needed to be optimized by ensuring that there are no repeating calling systems or looping that may cause the garbage to become to pile up and be kept in the memory.

The process continued to optimize the rendering. The rendering process was the highest of them all. So, to optimize the rendering process, one of the optimization processes that are worth trying is to mix real-time lighting and baked lighting, by creating a static object to bake. Real-time lighting may cost performance whereas the baked will saved some performance. The next thing that is worth trying is using a batching system to group the same object into one object so that the system may save some space for rendering. There is also occlusion culling to erase the object that doesn’t include in the camera. And the last is to check the tris and verts of the objects. The more triangles and vertex in one layer may cause a bad performance.

2.2 Testing Method

Results from the testing can be seen at the checking stages. Checking uses profiling by observing the output of data and checking the results are optimized. The results of the optimization should be under 60 FPS/ 16.8 ms for the process to be continued. If the result hasn’t reached the standardization target, the cycle will continue until the score reaches the specified value.

3 Result and Analysis

From the discussion of the research method, there is the process of developing the game “X” by increasing for better performance to continue for the next development stages.

3.1 Revision Rendering & Scripting

Optimization is carried out by creating the fps value into 60 fps or 16.8 ms by taking a median sample from the profiler from some parts of the output frame. The process starts from the optimizing script. Two parts needed to be done. The first part is to change the default Unity state which is only to make the device running at 30 FPS into 60 FPS, by adding a script named “Application Set” to increase the maximum value to 60 fps. After that check the usage of calling mode GUI in the “FPS Controller” by reducing the calling time from one frame into one second. After that, the project was exported into becoming prototype 2 which is the game can be launched at 60 fps and reducing the amount of scripting process as we can see in the figure below.

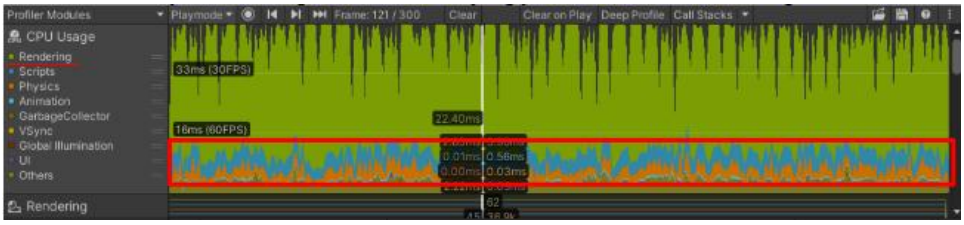
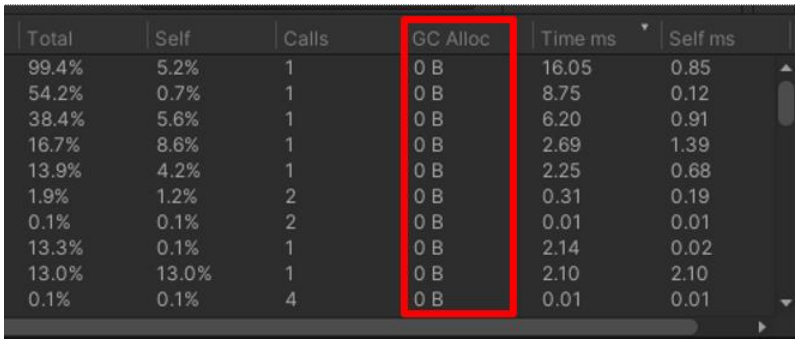


Fig. 4. Optimization Scripting, Prototyping II

It can be seen that the script has been under 16 ms and the value is 3.96 ms if compared to the previous value which is 24.71 ms. Then, the process can be continued to the next step. After doing some profiling, there are garbage fill for every frame that may cause bad performance. Garbage that fills out the memory can cause lagging and crashing while in-game. The problem came from the same script called “FPS Controller”. The script is creating a “String” for every frame. the solution was to resolve the string by adding a deletion for the string process. But after the “FPS Controller” script itself didn’t have many functions just to know how much performance was created in-game. The conclusion is to delete the “FPS Controller” script. The result of exporting and checking implies that the GC Allocation from the usage was 0 KB and now did not reduce the performance of the game and become the final results of prototype III.



Total	Self	Calls	GC Alloc	Time ms	Self ms
99.4%	5.2%	1	0 B	16.05	0.85
54.2%	0.7%	1	0 B	8.75	0.12
38.4%	5.6%	1	0 B	6.20	0.91
16.7%	8.6%	1	0 B	2.69	1.39
13.9%	4.2%	1	0 B	2.25	0.68
1.9%	1.2%	2	0 B	0.31	0.19
0.1%	0.1%	2	0 B	0.01	0.01
13.3%	0.1%	1	0 B	2.14	0.02
13.0%	13.0%	1	0 B	2.10	2.10
0.1%	0.1%	4	0 B	0.01	0.01

Fig. 5. Optimization GC, Prototype III

After finishing the garbage problem, the next process is to optimize the rendering process. In this case, the process was divided into two parts where the process of rendering optimization needs a lot of approaches. The first thing to do is to optimize the lighting setting. The result of using real-time lighting may cause a drop in performance. In this process, the next step is to mix real-time lighting for the dynamic object, and baked lighting for the static object. After that occlusion culling was included in the process where the rendering process will be focused on the screen camera. To maximize the quality of performance. the texture was reduced, for UI and the object of rendering including the skybox. After that, the result was shown in the figure below, which created prototype IV.

Fig. 6. Optimization Rendering, Prototype IV

Prototype IV resulted that the rendering process is still heavy for performance and has not been optimized well. For that problem, we need to configure the process further. The problem may be varied, but in this case, the problem that we faced came from the mesh. Mesh may

include the object characters, terrain, and the other supporting object. The mesh may create a mesh that works heavily on the processing unit. To optimize the mesh, change the terrain with the object made from the *blender,* and create another mesh character with a more optimized tris and verts than before. The result may improve the quality and change the tris with the new result which can be seen in the figure below.

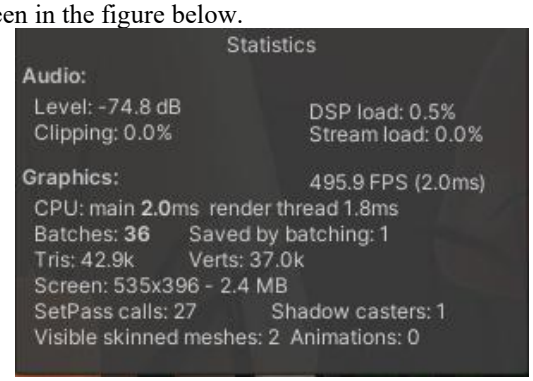


Fig. 7. Revision Statistics

The results show that the tris was optimized into 42.9k with 37.0k verts and reduce the number of batching to 36, from the statistic, shows that the new results are lighter than before and way more optimized. Then the game was exported and checked. The result can be seen in section 3.2 Result of Profiling.

3.2 Result of Profiling

Fig. 8. Optimization Rendering, Prototype V

From the Table 2, shows that the optimization of rendering reduces from 17.74 ms to 9.05, which is reduce about half of the performance. From here, we can see that rendering has been optimized very well, where the score for optimization should be under 16.8 ms. And from the figure, the total usage of the CPU Usage is 16.11, and it is under 16.8 ms. The game has reached 60 fps and the CPU usage is quite stable with not much spike. The data from the profiler is shown in the table below.

Table 2. Data Prototype 5

CPU Usage	Output Score	Optimized
Rendering	9.05 ms	Yes
Scripting	2.94 ms	Yes
Physics	0.75 ms	Yes
Animation	0.42 ms	Yes
Garbage Collection	0.00 ms	Yes

Vysnc	0.00 ms	Yes
Global Illumination	0.01 ms	Yes
UI	0.03 ms	Yes
Others	2.74 ms	Yes

3.3 Analysis Optimization

The analysis is to test by observing the data profiler and compare with the conclusion from the preliminary research which is game should be around 60 fps when running with mobile devices. The data reach 60 fps and is quite stable so the game has a smooth performance compared with the prototype. The game didn't have issues like lagging or even crashing. The game was even tested with another similar mobile device that shows the gameplay performance running smoothly. Even though there is a spike, it is not bothering the gameplay because the result doesn't even hit 30 fps [16].

4 Conclusions

From the data of preliminary research, stated that the performance of the game "X" does not optimize. Optimization is needed to ensure the development process of creating the game, and in this research there 5 results from the process of optimization which called revision optimization. The results from revision optimization cycle is divided into scripting, GC Alloc, Rendering I Optimization, and Rendering II, which are optimized well in this research by having a total score of 16.11 ms in the CPU usage which the under the minimum score of 16.8 ms.

The game was tested into another similar device and shows that the gameplay running smoohtly, which shows the results was acceptable in this research and the project will be continued for further development stage.

Footnotes. In this research, the process of optimization is on the bigger picture. More research is needed to improve further research regarding the process of optimization which includes the more specifics result of optimization.

Acknowledgments. Thank you for the Zetta Nusantara Company that helps and facilitates the development of this research.

References

[1] André Syvertsen, Angelica B. Ortiz de Gortari, Daniel L. King, Ståle Pallesen, “Research Report SAGE,” Problem mobile gaming: The role of mobile gaming habits, context, and platform, pp. 1-17, 2022.

[2] Socialpeta, Mobile Gaming industry statistics and trends for 2021, Business of Apps, 2021.

[3] S. Knight, “What Are 2.5D Games? How They Differ From 2D and 3D Games,” MUO, 30 June 2021. [Online]. Available: https://www.makeuseof.com/what-are-2-5d-games-2d-3d/. [Accessed 28 04 2022].

[4] N. Kustriawan, “Journal of Applied Multimedia and Neworking,” Pembuatan Game Bamboo Menggunakan Aplikasi Unity 3D, vol. 2, pp. 1-5, 2019.

[5] Sandro Villinger & Ivan Belcic, “FPS and Optimize Your PC for Game,” 2021.

[6] T. Amsterdam, “Granulate,” Mastering Performance Optimization - The Basic Metrics and What's Wrong, 2021.

[7] T. Unity, “Unity,” Optimize Your Mobile Game Performance, 2020.

[8] C. Dinkinson, “Optimization Unity Second Edition,” Second Edition ed., Packt Publishing Ltd, 2017.

[9] Riwinoto, “Journal of Applied Multimedia and Networking (JAMN),” Analisis Performa Prototype Game Pada Platform Android, 2021.

[10] T. Long, “Newzoo,” *Utilizing Mobile Device Data to Forecast Addressable Market & Set Minimum Specs for Your Mobile Game or App,* 4 Augustus 2021.

[11] Georgios Koulaxidis, Stelios Xinogalos, “Performance with Basic Optimization Techniques in Unity,” Improving Mobile Game Performance with Basic Optimization, vol. 3, pp. 201-223, 2022.

[12] A. Sirota, “Makaka Games,” *Unity Optimization : Mobile & Desktop,* 2021.

[13] Y. Ardianto, “Memahami Metode Kualitatif,” *Memahami Metode Kualitatif,* 2019.

[14] MasterClass Staff, “Guide to Frame Rates,” How Frame Rates Affect Film and Video, 2021.

[15] J. Ardyanto, Interviewee, Optimasi game "World of Tofu". [Interview]. December 2021.

[16] M. Hergaarden, “Optimizing Performance of Unity Games,” Optimizing Performance of Unity Games: A Battle Tested Plan, 2021.