IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

3D Game Development Engines and 3D Modelling

Rajinder Singh

Dept. of Computer Science and Applications PUSSGRC Hoshiarpur (Pb.), India

Corresponding Author: rajinderid@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i4.10471053 | Available online at: www.ijcseonline.org

Accepted: 22/Apr/2019, Published: 30/Apr/2019

***Abstract*—** Game engines are software toolkits which are very helpful in developing games. There are two types of game engines, commercial and free open source. Developers use these for creating and developing games. In this paper a discussion related to 3D game engines is made. Some of these engines are also helpful for creating games for gaming consoles device. Choosing a good game engine is an important decision. While designing a game it is very important to consider the platform for which these are developed. One factor which is also important is scripting language. Various resources which are utilized during the play of the games are also important. Main attention is given to CPU utilization, GPU utilization and RAM used by the games which are developed by these engines.

***Keywords*—** Rendering, Scripting, Physics, 3D Game Engine, 3D Modelling

I. INTRODUCTION

Game engines are software toolkits which are used by the people to build video games. It consists of a number of tools and components which help quick development of the game. Developers use these software development environments for designing and creating the games [1]. These engines are useful to create the games for consoles mobiles, computers and other gaming devices. For a game developer choosing a good game engine is an important decision. Choosing a good game engine affects the result of the product. In this paper some famous 3D game engine are discussed. All these engines have some drawbacks as well as some qualities. For discussion main engines which are chosen are i) Unity3D ii) Panda3D iii) Unreal Engine 4 iv) Ogre3D v) jMonkey Game Engine

Rest of the paper is organized as follows, section II explain main components of a Game Engine. In section III some differences between 2D and 3D games are explained. In this section 3D Modeling process is also explained. In section IV main 3D engines which are taken for this study are explained. A detailed comparison with respect to various resources is also done. Conclusion is given in section V.

II. MAIN COMPONENTS OF A GAME ENGINE

Main components of the gaming engines [2] are i) Rendering ii) Scripting iii) Artificial Intelligence iv) Physics v) Input vi) Collision Detection vii) Animation viii) Network x) sound (Figure 1).

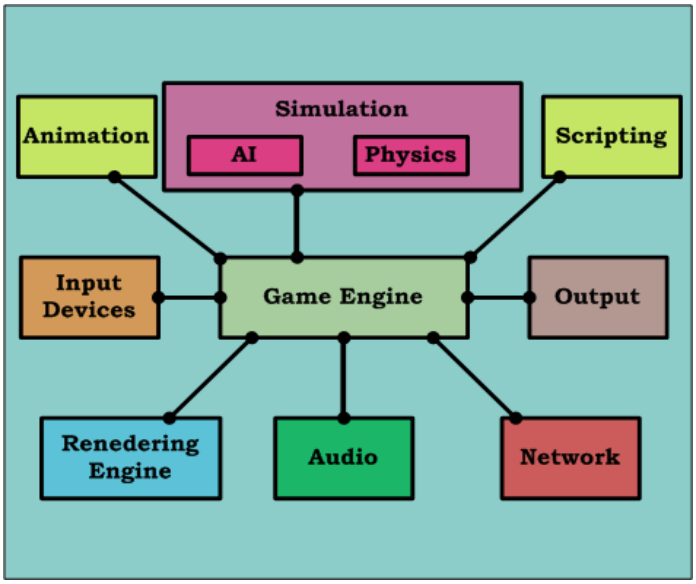

Figure 1 Main Components of a Game Engine

Rendering

It is a process of generating actual 2D image or animation from a 2D/3D model with the help of a program. It consists of the visual parts of the game. For assisting CPU a purpose built device GPU is used for calculating rendering calculations. It is used for handling light, shadows and rays. Rendering is mainly used in simulators, movies and creating visual effects. Main algorithms which are used for renderingare Scale line rendering, Ray casting, Ray tracing and Radiosity [3].

Rendering in case of Unity3D is shown below (Figure 2). It renders canvas in screen space and distance can also be set from the specified camera.

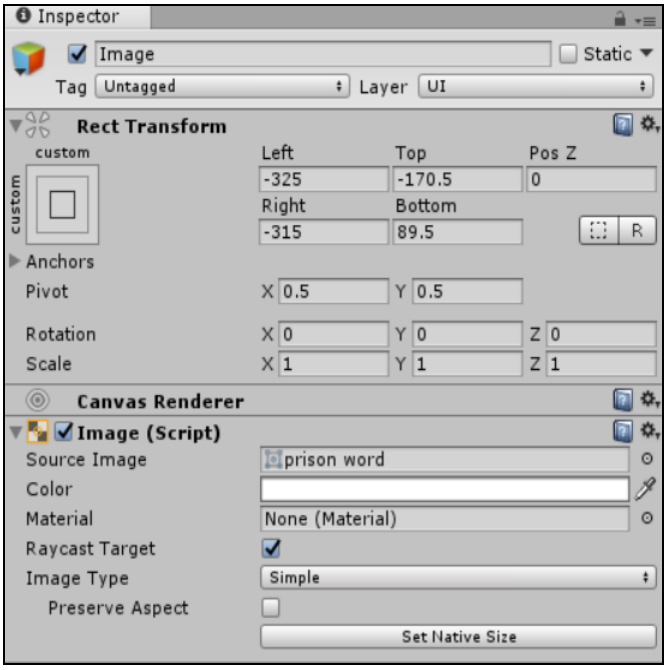

Figure 2 Rendering in case of Unity3D

Scripting

Scripts are essential elements in all the games. Scripts are used for responding user's input and also arranging the events in the game play. Scripts are also used for creating graphical effects and controlling the behaviour of objects. Here developer can use a programming language to control some parts of the game. Scripting languages are used to build consol and window based games. Main languages which are used for scripting are Python, Lua, C++, and Java and C #. For browser games main languages which are used are HTML5, CSS3, JavaScript and SQL. In case of Unity3D scripting is done with C# language [4]. Figure 3 shows an example of script written in C# in case of Unity3D game engine.

```
else if (myState == States.stairs_2) {stairs_2();}
else if (myState == States.courtyard) {courtyard();}
else if (myState == States.floor) {floor();}
else if (myState == States.corridor_1) {corridor_1();}
else if (myState == States.corridor_2) {corridor_2();}
else if (myState == States.corridor_3) {corridor_3();}
else if (myState == States.closet_door) {closet_door();}
else if (myState == States.in_closet) {in_closet();}

in_closet() {
ext.text = "Inside the closet you see a cleaner's uniform that looks about your size! " +
   "Seems like your day is looking-up.\n\n" +
   "Press D to Dress up, or R to Return to the corridor";
f (Input.GetKeyDown(KeyCode.R)) {myState = States.corridor_2;}
else if (Input.GetKeyDown(KeyCode.D)) {myState = States.corridor_3;}

closet_door() {
ext.text = "You are looking at a closet door, unfortunately it's locked. " +
   "Maybe you could find something around to help enourage it open?\n\n" +
   "Press R to Return to the corridor";
f (Input.GetKeyDown(KeyCode.R)) {myState = States.corridor_0;}
```

Figure 3 Scripting Example using C#

Artificial Intelligence

It is interaction of a player with the game environment. It is used for non playable characters. AI is used in 3D games for dealing with the non player characters. Objects which are behind the player such as enemy or animals are called non player character. AI is used to make these NPC objects intelligent and these behave like human. These can change their skill depending upon the game level. AI is used to give intelligent and adaptive behaviour to NPC's similar like human. These NPC's use pathfinder and decision tree AI techniques to guide them in the game [5].

Physics

Physics engine is used to give real world like simulations to the various objects in the 3D game. It is helpful in giving real world like effects to the objects. It is used to deal with basic physics phenomenon such as gravity, rotation and fluid dynamics. It is also used to create real world like environments in the game. It deals how the objects will interact with each other in the game [6].

Input

One aspect of the game is how it is played by the user. Game engines support various inputs such as keyboard, mouse and touch. Other inputs which are supported by the game engine are joystick, multi-touch, steering wheel [7].

Network

This component of the game engine helps building multiplayer online games. User can play game with their friends online. This component of the game engine handles server client communication and all the TCP/UDP traffic. Other things which are handled by the network components are: different speeds of the player's device, different platforms of the players and different communicating speed [16].

Sound

Sound is also a major component in a 3D game. All the sound related activities are handled by audio engine. Main function of the audio engine is to play the sound without any delay when needed. Since getting the sound data from disk and then playing it will cause delay, so these audio files are placed in memory. Some of the sound files are played for whole time period of the level. Some of the audio files such as dialog, gunshot or footsteps etc. are played when the character trigger that sound. All these functions are handled by the audio engine. Figure 4 given below shows the audio setting in case of Unity3D engine. Audio Source is used to set up audio file in the game. Audio can also spread among different speakers.

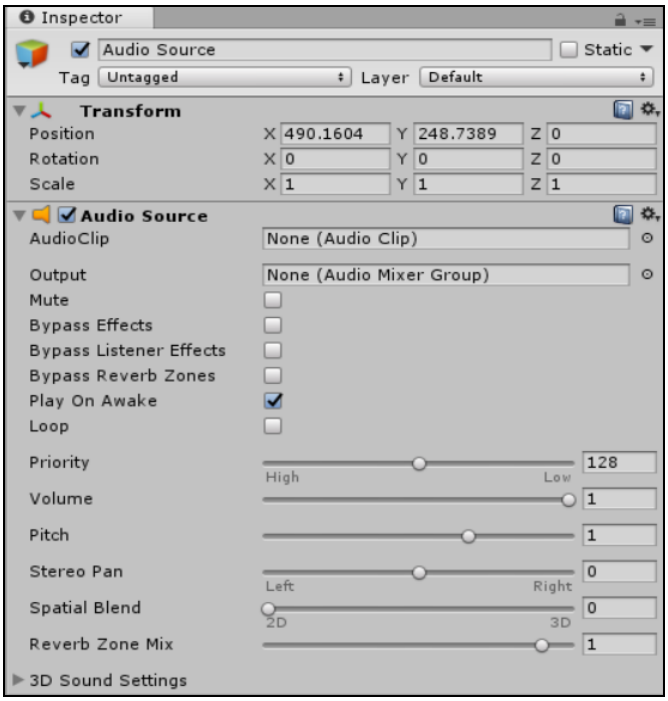

Figure 4 Audio setting in case of Unity3D

III. 2D/3D GAME AND 3D MODELLING

2D games are also called platform games. Here the character is held on a platform. Character can walk, run, jump, punch, shoot and can collect objects. These objects can be power or treasure (Figure 5).

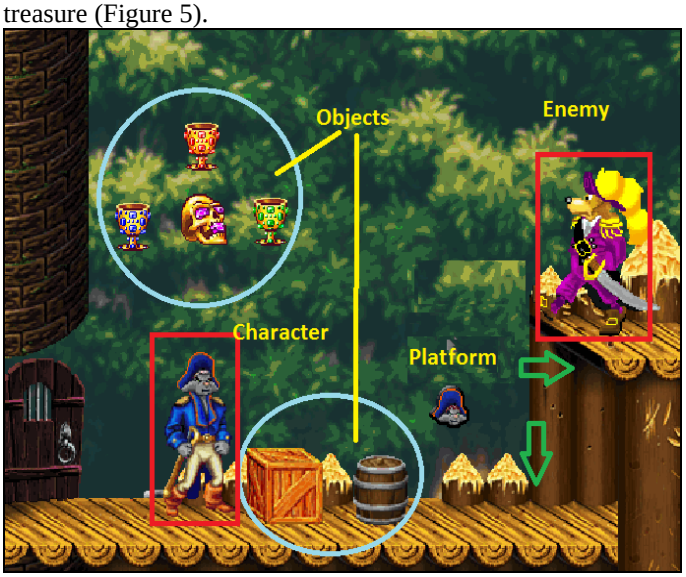

Figure 5 Platform, Objects, Character and Enemy in a 2D Game

Main motive to play the 2D games is that developer get innovative idea. In 2D games character is cartoon like. In 3D games game developers can give depth to the objects, therefore objects are more realistic as compared to 2D character. Sometimes facial expression is also given to the character to look it more realistic (Figure 6).

Figure 6 2D and 3D characters

Main difference between 2D game and 3D Game development are:

Coordinates

In 2D game two coordinates x and y are used to represent an object in game. In 3D game three coordinates x, y and z are used to represent an object. In 3D there is an extra dimension used for depth (Figure 7).

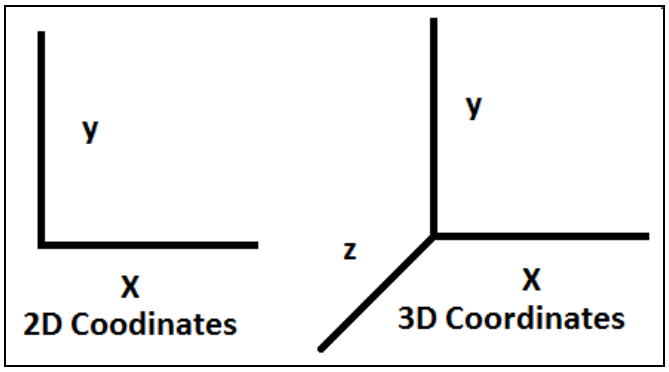

Figure 7 Coordinates in 2D and 3D

Graphics

In 2D rendering process requires less data. In 3D rendering process requires large amount of data.

Movement of the Character

In 2D game player moves left to right or right to left until goal is reached. In 3D game player can move deeper and closer to screen.

Physics

In 2D game rotation of an object is around only one direction that is around z. But in 3D game rotation of an object are around three axes X, Y and Z (Figure 8).

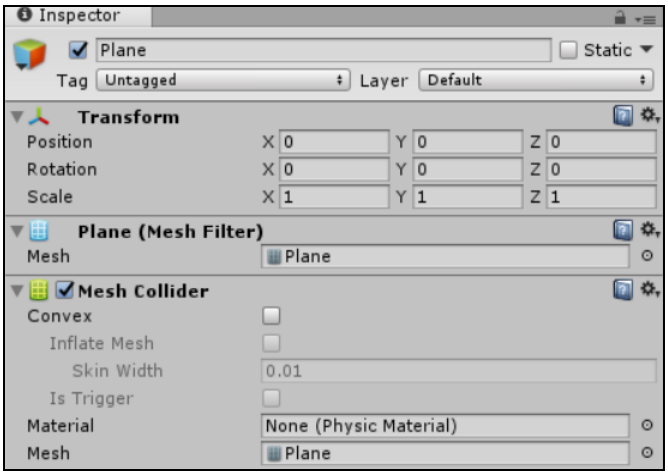

Figure 8 Rotations of Objects in 3D

Production

In case of 3D game rendering, arts, physics and animation process are very complex. Therefore 3D games are time consuming and costly as compared to 2D games.

3D Modeling

3Dmodelling means mathematical representation of an object in three dimensions with the help of a computer. It is called 3D model. It is displayed on the monitor with the help of 3D rendering. Popular methods which are used to represent a 3D model are Polygonal modeling [14], Curve modeling and Digital sculpting [8].

IV. GAME ENGINES FOR THIS STUDY

a) Unity3D

Nowadays Unity3D engine is very popular among the game developers. It has many tools and features. Unity3d editor is well structured. Scripts can be easily set. This engine allows using high level language for game development and these languages are very easy to learn. It uses C# or JavaScript programming language [15]. Its documentation is very rich, a full description with a lot of examples is given. For better understanding video tutorials are also given. Developers can also learn about this engine with the help of interactive exercises. Moreover third party tutorials are also provided for learning. Main companies which use this engine are Microsoft, Sega and Paradox. Its drag and drop property allows quick development of the games. It also contains large number of assets for those developers which are not good artist. They can use these assets. These assets can be easily added to the game. Mainly these assets consist of music, arts, code even module. Its Mechanim feature allows animation of one model can be applied to other model. Unity GUI allows developers to manipulate, pause, check the game development and progress by frame by frame. It also allows the developers to use 3rd party IDEs. It also allows free development of the game for mobile devices. Its cross platform integration features helps game developers to switch from one platform to other easily. It supports around 25 platforms [9].

b) Panda3D

Panda3D is a free open source game engine created by Disney VR studio. It is very popular among game developers. It is open source and game development can be done without any royalties. It contains many functions which are helpful for crating 3D games. It allows C++ language which is easy to learn. This engine itself is written in C++. It also allows python up to some extent. Its physics engine is very simple. At the same time it supports NVIDIA, PhysX and ODE and bullet. Developer can check performance with a number of tools. It also has a web plug-in. Therefore development of the game can be done for many famous web browsers. Game scenes can be edited and designed using Maya and Blender and can be imported into Panda3d [10].

c) jMonkey

jonkey is freely available for the game developer. Games can be developed as well as released with no royalties. Due to the open source nature of this engine it can be used for creating different plug-ins. Beside its own IDE it also support Net-beans IDE and Eclipse IDE. It is developed in Java language which is also used for the game development. This language is very fast as compared to RUBY and python. JME allows better game development. Gaming code can be used in iOS as well it supports many android based devices. It can be used for handling inputs, physics terrain and networking [11].

d) Unreal Engine 4

This engine is provided with free development license. It allows quick prototyping of the game. It allows C++ programming language. Game engine is also developed in this language. Editing and game development is very easy. Updating the game engine is also very simple. It allows creating amazing effects. It supports windows, MacOS and Linux Operating Systems. It support many features such as animation, AI, lighting, terrain, and FX. Its compilation is very fast. Its Blueprint Visual Script feature also allows non programmers to create games. One drawback of this engine is that it does not allow creating games for previous consoles [12].

D) Ogre3D

Ogre3D is a popular open source graphics engine. It is written in C++ language. It is simple and very easy to render 3D scenes with this engine. Some of the features related to rendering, culling and transparency are automatically done in this engine. This engine also supports Direct3D, OpenGL and WebGL. Scene management is highly customizable and flexible. Predefined scene management classes can also

plug- in into this engine. It also supports multiple shadow rendering of the objects in the game [13].

Comparison:

Table 1 show various platforms which are supported by these 3D game engines. All these gaming engines support Windows. Some of them also support game console device such as Playstation2 and Playstation3.

Table 1. Platform and Programming Language

Game Engines	Platform Supported	Programming Language Supported	Asset Stores
Unity3D	Windows, Mac OS, Android, PS4,Linux,	Java Script, C#	Yes
Panda3D	Windows, Linux, iOS	Python, C++	No
Unreal Engine 4	Windows, PS3,	C++, Unreal Script, Blueprints	Yes
jMonkey	Windows, Linux	Java	No
Ogre3D	Windows, Linux, Xbox, PS3	C++	No

Table 2 show additional features which are supported by these game engines. Networking means support for multiplayer online game play, Physics means real life behavior of the objects in the game, and Rendering means support for creating real life scenes and visuals.

Table 2. Networking, Physics Engine, Rendering

Game Engines	Networking	Physics Engine	Rendering
Unity3D	Yes	Yes (Separate one for 2D and one for 3D)	Yes
Panda3D	Yes	Yes	Yes
Unreal Engine 4	Yes	Yes (PhysX 3.3)	Yes (DirectX11)
JMonkey	Yes (SpiderMonkey)	Yes (jBullet)	Yes
Ogre3D	Yes	Yes (Bullet)	Yes

Comparison of these game engines with respect to various resources: Main resources which are used during playing a game are given below.

a) CPU
b) GPU
c) RAM
d) CPU TEMPRATURE
e) FRAMES PER SECOND (FPS).

a) CPU

CPU is the most important resource that is used in gaming. Some games are the CPU intensive while some games are GPU intensive. CPU is mainly used by the games to produce the frames that are to be rendered by the GPU. It is also noticed that on the heavy load of the gaming the CPU temperature, electricity draw and usage are also increased. On the behalf of usage the 2D games are less CPU and GPU intensive than the 3D games.

b) GPU

GPU is known as the graphics processing unit. That may be an integrated or a discrete. A GPU unit is mainly used to render frames that are provided by the CPU. GPU utilization when no game is played is shown below (Figure 14).

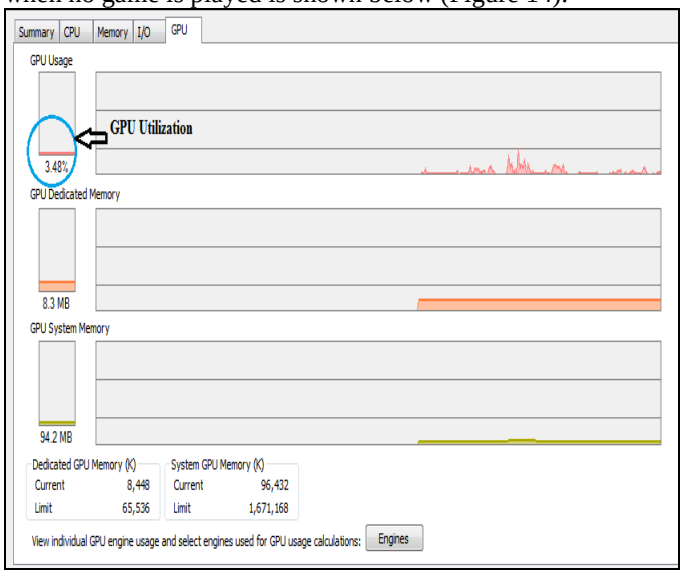

Figure 14 GPU utilization in normal case

A GPU only knows how to render the graphics nothing else. The discrete or dedicated GPUs are more powerful than the integrated GPUs. GPU utilization when a game is played is shown below (Figure 9).

Figure 9 GPU utilization during play of a game

c) RAM

In terms of gaming the Ram is the place where all the files of a game are loaded that are to be used while running the games. It also noticed that the more Ram helps the games to run the games smoothly without lag and stuttering. And the frequency of the Ram that is usually measured in MHz is also matter when running the games. Higher frequency Ram gives the more playable frame rates, as the more Ram does not create any type of bottleneck.

d) CPU TEMPERATURE

If the CPU is running at the stock frequency then the stock cooler is enough to cool it. But if you are using an over-clocked CPU then the Liquid cooler (AIO) or the third party air cooler are required. CPU intensive games consumes more CPU power so the temperature may goes up so it also necessary to keep the CPU at the Safe temperature. Although CPU and GPU can work up-to 95 degree Celsius, but the GPU and CPU will suffer from the thermal throttling that will drop the performance.

5) FPS

Basically Fps generally is the calculation of frames that are produced or rendered by a GPU in unit time. 60 FPS are the decent frame rate that is required to enjoy the smooth experience of games. Fps is calculated to benchmark a GPU to know the capability of the graphics card.

Method Used: First different game-engine games were chosen and played for the fixed amount of time. Then the resources such as CPU temperature, GPU utilization, RAM utilization, Frames per second were noted. Averages of all above were taken.

Games Used: Spoxel (jMonkey), Cuphead (Unity), Deadpool(Unreal Engine 3), Zombi driver (Ogre3D), Quest for Fire(Panda3D)

Device Configuration: I-5 7th Gen 8GB RAM, Graphics card AMD Radeon 650 2GB DDR3

Graph given below shows the average CPU temperature during the play of games. Although threshold range of the temperature highly depends upon the manufacturer and design of CPU and GPU, but nowadays lower range of CPU temperature is considered optimal. Nowadays it should not exceed 80 degree Celsius.

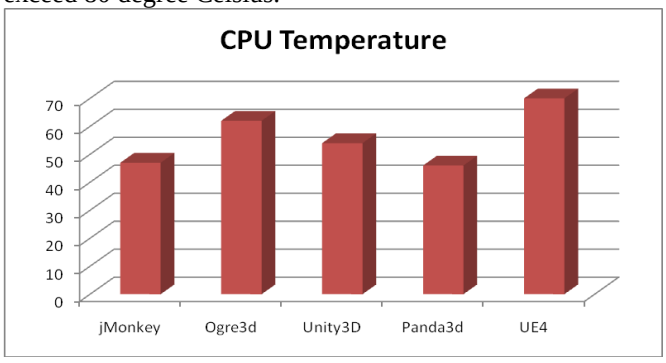

Figure 10 CPU Temperature

From the Figure 10 it is clear that CPU temperature in case of UE4 engine game is high as compared to other engine games. GPU utilization means how much GPU is utilized. It is measured in %age. 99% GPU utilization means the GPU is maximum utilized. From the Figure 11 it is clear that there is maximum GPU utilization in case of Ogre3D and UE4. But in case jMonkey and Unity3D it is less utilized.

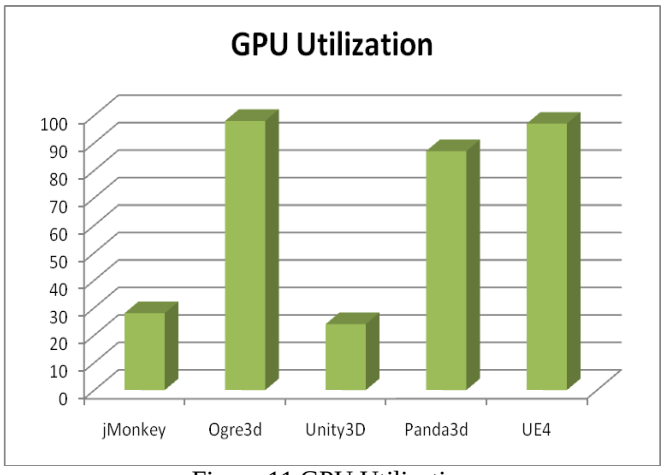

Figure 11 GPU Utilization

Here in the given below average frames per seconds (FPS) are taken. All the games were played for a fixed amount of time. Then number of frames per second noted from starting time. Then the average is taken from the start of the game, during when menu comes in the game, then during the play, when player looses a life and middle of the games. FPS means how much frames are rendered by GPU per second and how many displayed on the screen per second. 30 to 60 FPS rate is considered ideal for the games.

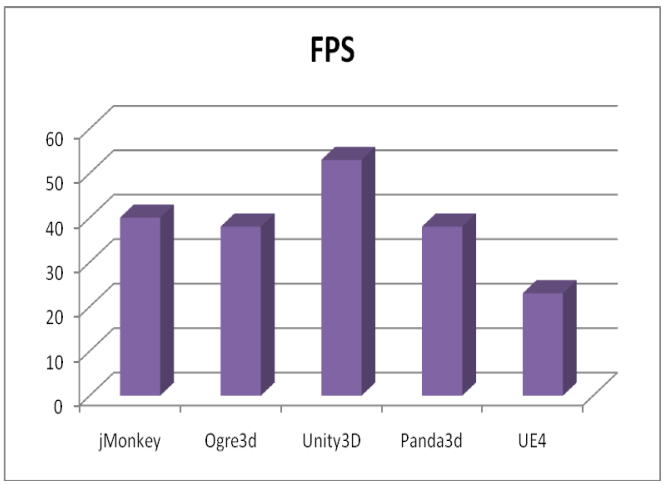

Figure 12 Frames per Second

From the graph (Figure 12) it is clear that in case of Unity3D FPS is 50. It is around 20 for UE4 engine.

RAM is the area where game related variables and game related content is stored. RAM is used because it is much faster than the hard disk. RAM utilization during playing these games is shown in the Figure 13.

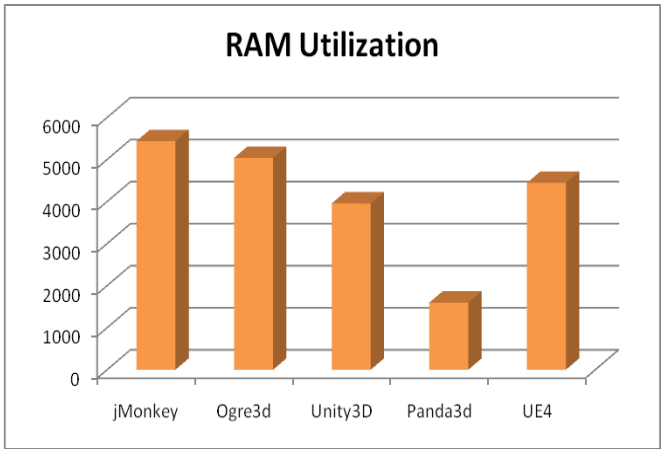

Figure 13 Usage of RAM

V. CONCLUSION

Creating a 3D game is a complex process and it requires good knowledge of design and technical knowledge of that engine. Therefore choosing a good engine is very important. With the help of a game engine game developers need not to develop game from the start. Game engine reduce the time in game development. It is easy to make a realistic game with the introduction of physics engine, rendering and sound features in the engine. Game developer can check advantages as well as disadvantages of these engines. Then they can choose the one which fits most to their requirements. In this paper role of a game engine with respect to game development is discussed. Various components of a 3D game are also highlighted. Some of the common engines which are available nowadays for 3D game development are discussed. Then a comparison is done by considering various resource consumption. Main resources which are used during the play of a 3Dgame are CPU, GPU and RAM. Consumption of these resources is recorded by playing different game-engines game. Nowadays games with good special effects are in the market. If a game developer want to create special effect in his game then he can chose Unreal Engine 4 because this engine can create realistic visual effects. This engine contains a rich support for creating particle effects and environment effects. It also support realistic rendering. One disadvantage of Unity3D is that graphics quality is not that good.

REFERENCES

[1] Navarro, Andres, Juan Vicente Pradilla, and Octavio Rios. "Open source 3D game engines for serious games modeling." Modeling and Simulation in Engineering. IntechOpen, 2012.

[2] Finney, Kenneth C. 3D game programming all in one. Cengage Learning, 2013.

[3] https://en.wikipedia.org/wiki/Rendering_(computer_graphics)

[4] https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html

[5] https://hub.packtpub.com/best-game-engines-for-ai-game-development/

[6] https://www.g2.com/categories/physics-engine

[7] https://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/

[8] https://en.wikipedia.org/wiki/3D_modeling

[9] https://unity.com/

[10] https://www.panda3d.org/

[11] http://jmonkeyengine.org/

[12] https://www.unrealengine.com/en-US/

[13] https://www.ogre3d.org/

[14] Remondino, Fabio, and Sabry El‐Hakim. "Image‐based 3D modelling: a review." The photogrammetric record 21.115 (2006): 269-291.

[15] Xie, Jingming. "Research on key technologies base Unity3D game engine." 2012 7th international conference on computer science & education (ICCSE). IEEE, 2012.

[16] Lewis, Michael, and Jeffrey Jacobson. "Game engines." Communications of the ACM 45.1 (2002): 27.

Authors Profile

Rajinder Singh is an Assist. Professor in DCSA, PUSSGRC, Hoshiarpur, Punjab, India. He has more than fifteen years of experience of teaching post-graduate classes. His areas of interest are Wireless Network Security, Cyber Security, Artificial Intelligence and Android Security. He is currently pursuing His Ph.D. Degree from P.U. Chandigarh, India.

