A study on game development using unity engine

Cite as: AIP Conference Proceedings **2375**, 040001 (2021); https://doi.org/10.1063/5.0066303
Published Online: 05 October 2021

Mukkamala S. N. V. Jitendra, A. Shanmuk Srinivas, T. Surendra, et al.

A Study on Game Development Using Unity Engine

Mukkamala S N V Jitendra[1, a)] A Shanmuk Srinivas[1], T Surendra[2], R Venkta Rao[3]
P. Ramaiah Chowdary[4]

[1]*Dept of CSE, GITAM (Deemed to be University), Vishakapatnam, Andhra Pradesh, India*
[2]*Dept of Mathematics, GITAM (Deemed to be University), Vishakapatnam, Andhra Pradesh, India*
[3]*Department of Computer Science and Engineering, Wollega University, Nekemte, Ethiopia*
[4]*Department of Information Technology, Sir C.R.Reddy College of Engineering, Eluru, AP, India*

[a)] Corresponding author: jitendra.mukkamala@gmail.com.

Abstract. In the current scenario of the fast-developing area of Software Engineering and Development and considerably more rapidly developing division of games, what is yet to come, is difficult to predict. In simpler terms, a project is an attempt concentrating on the making of programming. In a game undertaking, the item is a game. Be that as it may, a game is substantially more than simply its product. It needs to give substance to get charming. The product portion of the project isn't the one and only one and it must be considered in association with every single other part – the story, the characters, interactive experiences, fine art, etc. Twisted – 2.0 is a multiplayer endurance game, developed in a low-poly 3D environment, that is prepared to do all the while supporting 20 online players.

Keywords. *Software Engineering and Development, Game, Multiplayer Endurance Game, Low-poly 3D Environment.*

INTRODUCTION

Making games is a complicated subject. When large companies develop games, they work in big teams. Every member is assigned a specific role. May it be 3D modeling, coding, working with concept art, or sound design. Luckily over the last few years, there's been a big boom in the Indie games industry. In the current scenario, even small teams or individuals also can design and develop popular games. Earlier studios for game designing had to work rigorously for creating a game engine from scratch or buy an already existing one from a different studio which was not cheap. However, many independent engines have been made with the sole purpose of people using them to build their games on top.

This research paper involves a campaign into game development and encompasses each procedure from development to deployment with this repetitive part of programming advancement. It maintains the imperative quality expected in game development. Playing for stakes with the desire for winning goes as one of the endless meaning of gaming. It delineates that individuals would play a game with the desire to win. Be that as it may, there are different reasons why individuals mess around which incorporates the decrease of pressure and eventually for the good times. Likewise, results have indicated that individuals mess around less for the game itself yet for the healthy experience which the game makes: an adrenaline surge, an undertaking predicated on minds, a psychological test, or the structure games give, for example, a snapshot of isolation or the organization of companions. Individuals mess around to make second to-second encounters, regardless of whether they are defeating a troublesome game test, looking for help from ordinary concerns, which indicates along these lines that players in some cases mess around to calm themselves from the day's pressure. Henceforth, it isn't phenomenal to see people playing one sort of game or the other for relaxation when they are less occupied or while they are taking a break from their normal errands.

Essence of Mathematics in Engineering Applications
AIP Conf. Proc. 2375, 040001-1–040001-13; https://doi.org/10.1063/5.0066303
Published by AIP Publishing. 978-0-7354-4127-9/$30.00

The current game, Twisted – 2.0, has been structured as a low-poly 3D condition game. Matchmaking is the procedure of how we interface a few players together for online meetings in multiplayer games. An anteroom, in a multiplayer game, is the place individuals stick around before they go into a particular space to play. At the point when the player comes out of the room, the players are dropped into the lobby once more. The primary capacity of a hall is to help players rapidly discover a room thus that they can join the game. At the point when the player is said to be in a lobby, the player will have the option to see various rooms inside the lobby that can be entered. The player might have the option to see the status of each room. This game gives the adaptability of picking the characters among six accessible choices. The weapons can likewise be browsed from the stock. With the maxim of making due until the end, the player has five lives and should shield himself from different players who are in the battleground.

LITERATURE REVIEW

The ancient backdrop for game creation begins with the improvement of the fundamental computer games, although which game was the first depends on the meaning of computer games. Truth be told, these games required principles to outline computers to design them.

In 1952 Alexander S. Douglas', OXO was the initial PC game to use a computerized show. Later, a game that showed profits on the oscilloscope was created by Higginbotham Willy in 1958. The game was called 'Tennis for Two'. The engineer was a physicist who worked at the Brookhaven National Laboratory [1]. In 1961, a fundamental casing game is known as 'Spacewar!' was created by a gathering of Massachusetts undergraduates guided by Steve Russell. Nonetheless, the genuine business improvement of games started during the 1970s, when arcade computer games were promoted. Engineers worked inside large organizations to create such games [2]. Be that as it may, the industry didn't see tremendous change or transformation in-game structuring and improvement because countless consoles had comparative games.

Makers of Console before long began to assemble comforts that had the option to play self-ruling created games and they ran on a chip, which at last denoted the start of second-age reassures. Home PCs came into seeing in the market, which in the long run permitted singular software engineers to create games. This offered to ascend to the different working of equipment makers and programming producers.

Numerous games could be created by solitary people, as games were anything but difficult to design since, pictorial and memory restriction didn't take into consideration much substance. When of around1987, an ordinary computer game required a year to create and an additional half-year to design promoting. Ventures were for the most part solo endeavors, with single designers conveying finished games to their distributors. With the ever-expanding handling and graphical abilities of PC items, arcade, and reassure, alongside a huge increment in client desires, game improvement moved past the force and extent of a solitary designer to create a merchantable game in a sensible time. This started the start of the group-based turn of events. [3] In more extensive terms, during the 1980s, pre-creation included outlines and test schedules of the main designer. During the 1990s, pre-creation comprised generally of game workmanship sneak peeks. In the mid-2000s, pre-creation, for the most part, delivered a playable demo.

The main PC games showed up in the center of the twentieth century. Similarly, as the improvement of littler however progressively incredible innovation has helped drive the ages of PCs, the game business and the games created have been driven by a similar development. [4] This is because innovation has taken into consideration the development of incredible consoles on which the games truly dwell and are played. Similarly, as a similar innovation has considered PCs to be versatile as prove in workstations, individual advanced colleagues, and cell phones, it has taken into consideration the rise of portable game consoles, for example, the PSP, the Gameboy, and the Nintendo Wii. All these advancements have given a more extravagant and progressively vivid involvement with gaming with every improvement in the innovation eventually bringing about an expanding customer base that is continually being engaged by the item. [5] A little miracle we find that a lot of individuals mess around to incidentally disengage themselves from the real world, regardless of whether it be work, school, or at play.

Gaming reassures have developed from what we need to ground-breaking independent machines that play high-end games with very good quality illustrations and usefulness. As of now at the fourth era with the presentation of PlayStation 4 and the Xbox one, these consoles have preparing power that is as amazing as the PC framework itself;

it is in this way of nothing unexpected that one would see titles, for example, FIFA that precisely emulate the genuine situation to give the player a characteristic encounter while playing. Additionally, [6] these consoles have been utilized for cutting-edge recreations, for example, flight reenactment which considers the minor subtleties that are associated with guiding a plane.

OBJECTIVES

Within the scope of this paper, the purpose or objectives can be stated as follows:

- Maintaining the game-state consistent
- Providing realistic rendering of 3D components
- Developing the game free of glitches and errors

METHODOLOGY ADOPTED

In the development of a game, there are a few approaches that are utilized by different studios during the development process. The approach or methodology that was adopted for the improvement was Winston Royce's Waterfall model. This model was endorsed by game designers for game advancement which is known as the Game Waterfall Model. The cascade or waterfall model is where development is segregated into various stages. When a phase is finished, the user goes on to the following stage. The waterfall model permits returning to a prior stage, which is known as splashing back. This strategy is most appropriate for the proposed framework since it includes planning which is significant for individual and group ventures the same. Consequently, the game waterfall model is the favored technique for this project. There was a high threat for the project regarding plan however the waterfall model is an organized methodology that took into consideration time. There are five distinct stages to the waterfall model which all fill an alternate need towards the advancement of the ancient rarity. The stages were requirements gathering and specification, designing, coding, testing, and implementation which shown in Fig 1.

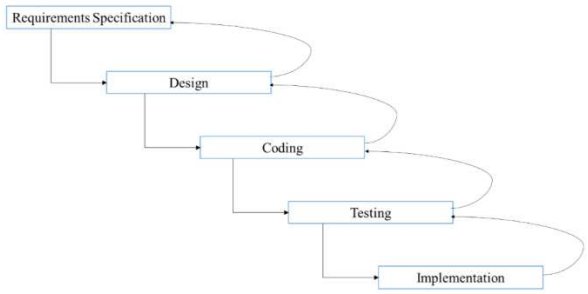


FIGURE 1. A common Life cycle of System Development

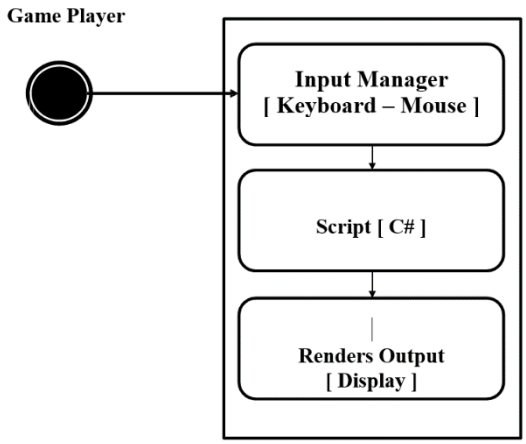


FIGURE 2. A System Environment

The gamer can communicate with the system by giving input to the system (By clicking the 'Play' button). The system processes the given inputs to the script, if any changes occur (if the value is changed), this object is sent to render to display the output (The player or character moves) shown in Fig 2.

REQUIREMENTS GATHERING

A. Normal and Expected Requirements

Software development is a prototype change from the support and maintenance of standard applications in the current software industry. Designing and developing an application or a game from scratch is a huge task. In terms of cost and commitment, it needs investment.

The following have been identified as the list of requirements for the development of the game:

- User-friendly system
- Scope for minimum expected requirements in PC.
- The operation should be easy.
- Game with measured coding.
- Maximum high definition.
- Efficiently design the whole system.
- Easy to update.
- Game developed with professional thinking.

B. External Interface Requirements

UI or Menu Driven Interface

- A game should have a menu bar component to be used to fulfill the needs of players and provides flexibility to quit or change options.

Software Interfaces

- "Twisted – 2.0" has been designed and developed with the following set of tools.

1. *Unity3D:* Any video game runs on a framework or architecture called a game engine. Unity, however, allows us to build games in both 2D and 3D that can be exported to a variety of platforms. Unity deploys to both the web, desktop, mobile, and console. For programming in Unity, one can choose between C# and JavaScript – the widely adopted programming languages [7].

2. *Blender3D:* Blender software is an open-source 3D creation software that supports the entire 3D channel and all the essential functions that this program is capable of and how it is being used. All 3D modeling has been done using Blender [8].

3. *Visual Studio 2019:* It is an integrated development environment that's very popular for .Net and C++ workloads as well as a ton of different other languages and frameworks. Visual Studio is what can be called, a live-in editing environment. It's also one of the most popular editing environments. VS is really powerful and one can make Windows applications in C#, F#, VB. Visual Studio is free for open source and small teams. It can also be used for mobile development on Android, iOS, and Windows [9].

4. *C# for Scripting:* C-sharp is known as an object-oriented programming language. Just like visual basic.net and visual c-sharp and most modern-day programming languages, what that means is that we build and model our application by looking at it as a blueprint. Another interesting fact about the c-sharp language is its popularity because it has become well documented. In the case of C#, we have the DynaMed framework and think of the dotnet framework as it contains things like the runtime of a virtual machine like CLR (common language runtime) which, anytime an application is executed it is executed through the .net framework [10].

DESIGN AND IMPLEMENTATION

The following chapter consists of the project design, system features, and also implementation features.

A. System Features

- Behavior Designer.
- Multiplayer.
- Lobby.
- Inventory.
- Health Counter.
- Ammunition.

a) Behavior Designer

Behavior Designer is a behavior tree implementation and offers an effective API permitting us to, without difficulty, create new tasks. It gives an intuitive visible editor making it possible to create complicated AIs without having to jot down the identical piece of code multiple times and may make modifications by way of altering a single line of code [11].

A fashionable overview of all aspects of Behavior Designer is explained. With Behavior Designer one doesn't want to recognize the underlying implementation but is a superb concept to understand, a number of the important thing concepts along with the varieties of tasks (motion, composite, conditional, and decorator).

Behavior trees are a combination of many distinctive techniques: hierarchical node modules, scheduling, planning, and motion execution. One of their main blessings is that they're smooth to recognize and may be created

with the use of a visual editor. At the only level conduct trees are a set of tasks. There are 4 specific varieties of tasks: motion, conditional, composite, and decorator [12].

The situations where conduct clothier is used over conventional programming languages are: At the best level, conduct trees are used for AI whilst finite nation machines (FSMs) are used for more trendy visual programming. While you may use behavior trees for standard visible programming and finite kingdom machines for AI, this isn't what each device became designed to do. Behavior designer has a few advantages: they provide lots of flexibility, are very powerful, and they are clean to make modifications to.

b) Photon Networking

Unity provides a package for multiplayer games, known as Photon Unity Networking (PUN). Adaptable and responsive matchmaking allows the players into similar rooms where their gadgets may be synchronized over the network. The Photon activities are just some of the functions [13]. The swift and dependable exchange is accomplished through dedicated Photon servers, so players/clients necessarily don’t have to connect individually.

PUN's Structure

The structure of this networking framework for Unity is not complex. An overview is given below. The Photon Networking package for Unity comprises three layers of APIs: The topmost stage is the PUN code, which enforces the game engine Unity-specific functions like RPCs, networked objects, and so on [14]. The next level encompasses the required logic to work with servers, to perform callbacks, for lobby creation, matchmaking, so on and so forth. The lowest or last stage is built of DLL files, which constitute the protocols, de/serialization, and so on.

PUN Features

Photon networking provides authentication, reliable, and rapid in-game communication through the Photon backend. Photon networking is a benefit due to Unity integration to easily expand and deliver or issue the multiplayer games globally,

- *Works like a cloud in real-time*

PUN games are facilitated in the all-inclusive circulated cloud for Photon to ensure reduced inertness, most limited full circle times for all the players around the world [15].

- *Multiplayer*

Associate, Correlate and Participate: PUN is the rock-strong basis for any sort of lobby or a room based multiplayer game. We listen to constructing an exceptional title at the same time as we cope with the re-creation's backend.

- *Multi-Platform*

It can be exported to desktops, consoles, or the internet and mobiles. PUN is the usual cross-platform provider for multiplayer games and ranks first for most Unity games.

- *Maximum Extensibility and Expandability*

Games built with the PUN scale smoothly and continuously and spontaneously in the PUN Cloud: from only some to tens of thousands of concurrent users. Equal and obvious cost estimates are included. This was proven successfully [16].

- *Matchmaking API*

PUN provides the feature for gamers to enter into games either randomly or by using specific searches. The initiator can create a list of game rooms and let the players choose one as per their choice. It works reliably on PCs, Consoles and the internet, mobiles, and within the web.

- *Personalization*

What type of multiplayer would the developer want to build? FPS or arcade shooter? RPG or RTS? Racing or Athletic? Another form of action re-creation? PUN aids and assists in this by allocating any type of room mostly established on video games on any famous gaming platform.

- *Client to Server*

No problems whether with Reliable UDP, TCP, HTTP, or Web sockets. Photon's highspeed client-to-server-structure is the business' most solid foundation for video games.

- *Unparalleled Adjustability*

Photon Unity Networking is incredibly flexible: one can connect with an iOS Game Center person with a person using Google Play Services, authenticate a person through Facebook or add a custom authentication, make use of a gaming provider.

- *Worry-Free Pricing*

Reasonable and Transparent: 100% utilization dependent on a simultaneous client premise. Adaptable: Start with a FREE arrangement, up-or minimize whenever. Effortless: no prompt top when surpassing CCU with any month to month plan. Ideal for the game's dispatch.

B. Aesthetics of Game

The game's visuals are quite simple. The design selections chosen for the game is in such a way to establish a low-poly atmosphere in the visuals. The aesthetics should be in such a way that the natural and easy-going and accessible nature of the game sets it apart from the other existent games. Also, the aesthetics are made chosen in this particular way so that they may appeal to the younger viewers. In addition to this, the low polygon count, count of the 3D assets, and the absence of photograph textures minimizes the nominal hardware requirements which can benefit likely players with older systems greatly [17].

C. Dependencies

The final platform for running the build of the game can either be on an Android/iOS/Windows/MAC. However, all the operations such as the development of the game and integrating it with the respective platform will be taken care of by the Unity game engine.

Building the Game

There are various in-built components provided by Unity to quicken the process of game development immensely.

The following are few such components:

- Physics Game Engine.
- Recognizing Collisions and managing them.
- Recognition of User Input.
- Position and rotation of game objects.
- Integration of multiple scenes.
- Node Wrangler, UV Editor for mapping of 3D models and materials.

C. Character Design

Five characters have been included in the game. Instead of using a different character each time, which can lower the optimization of the game, a new approach was adopted. An approach where a base model is created and all the necessary keyframes have been added. Utilizing this base model, the characters are designed on top of it. Similarly, only a single animation is applied to the base model which is reused for the other characters. The characters developed are modular. For instance, the design elements like skin color, clothing, accessories, etc is show in Fig 3. can be swapped with the other available options [18].

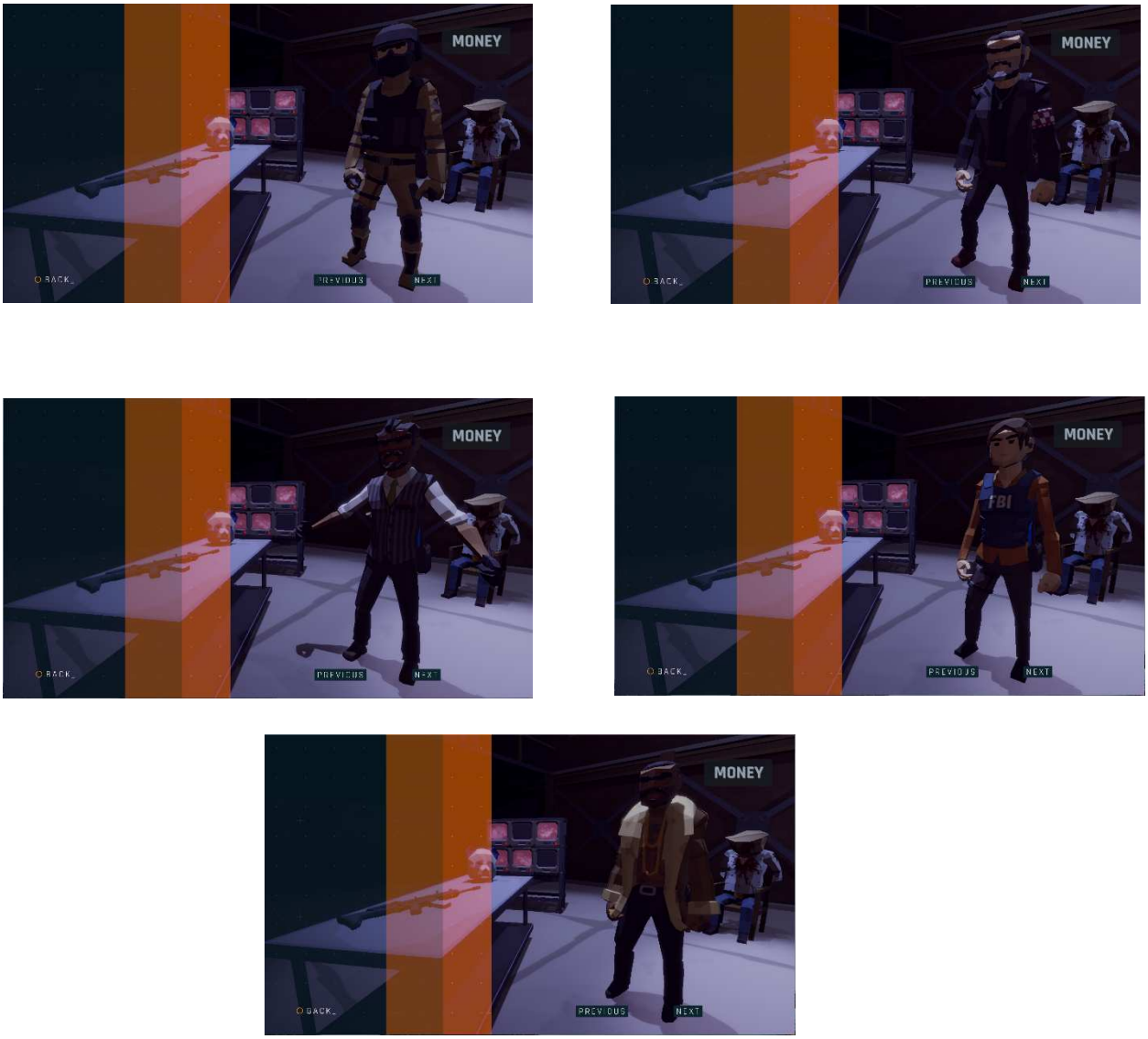


FIGURE 3. The different characters with the design elements like skin colour, clothing, accessories, etc.

D. Gameplay Mechanics

- Locomotion
  The player can shift and change his digital camera's position and his character chosen, around the game world.

- Battle
  The player in-game fights with the other opponents to win. If the player's character dies, he resurfaces at his base after five seconds.

- Uncertainty
  The game shall incorporate a feature called the fog of battle that shall disguise interactive gadgets from the player unless the game character is inside a certain range of the object.

- Winning Condition Mechanism
  If either player kills the other opponents or players 5 times without getting killed by any of his opposing players, he has won the sport. If any player quits the game using the 'Return to lobby' button from the in-game menu he's going to lose the game, the opponent player will win and the game's end cutscene will be played [19], [20].

E. Controls in Game

The controls for this game are not limited to a single hardware device and it can range from mouse and keyboard to a joystick depending on the platform on which the game is being played. The primary input devices for the player using the desktop version are the mouse and keyboard only.

Camera Movement

The camera movement for the developed desktop version is entirely dependent on the mouse movement. Changing the direction of the mouse in any direction will cause the camera to move in that corresponding direction thereby allowing the user to manipulate in which direction he wishes to see and the same is shown on his or her display.

Basic Actions (Movement and Attack)

The player has two essential movements that can be performed: motion and assault. Basic movements are achieved by using the mouse click button. If the player performs a right-click whilst the mouse hovers above a particular point in the game scene, he will enter into the aiming position with the cursor pointing towards the target. By holding the right-click, if the player performs a left-click, he shall attack or fire at the locked target. If the target isn't within range the player will walk towards the target till it's within the range. An additional crouch animation is also provided in case of stealth attacks. Clicking 'C' on the keyboard activates the crouch position and aiming plus attacking follows the same procedure.

TESTING

Few test cases have been included, to verify the integrity of the game and if various functions are working properly without any bugs and glitches.

Test Case 1

- Test Case: Testing if the game leads to memory overload in case the game is left turn on for several hours.
- Test Procedure: Run the game and leave it idly for a few hours.
- Expected Result: The game should work at the same speed and doesn't freeze or crash.
- Actual Result: Working is the same.
- Comment: No need to make any more changes but need to follow the same procedure of memory allocating as done previously.
- Accuracy: It depends on the hardware configuration.

Test Case 2

- Test Case: Testing if the multiplayer module is verifying the connection to a network is being created or not.
- Test Procedure: Connect to a network with good internet speed.
- Expected Result: Status changes from 'Offline' to 'Online'
- Actual Result: The status toggle is working.
- Comment: No further changes are required.

Test Case 3

- Test Case: Testing and examining if dynamic lobby creation is being created without any lag and bugs.
- Test Procedure: Build the game and run the built version on two separate computer systems. Create a lobby in any one system (host).
- Expected Result: Created lobby must be seen on the other system.
- Actual Result: Lobby is being created but is not visible on the other system.
- Comment: Have to inspect if we are retrieving data from the Multiplayer Listener from the parent class.
- Conditional test: Run the game again in both systems.
- Expected Result: Dynamic lobby creation should work.
- Actual Result: Yes, it is working.
- Accuracy: Depends on network speed.

RESULTS

On running the game application, the following is displayed to the player/user. 'Campaign' mode is the story mode of the game which has two levels. Maximum attention has been paid to the 'Multiplayer' mode in the current

project. Multiplayer mode can be played only when one is connected to the internet. This can be verified from the status option which shows ‘online’ if the player is connected shown in Fig 4(a) ,and (b).

FIGURE 4. (a) The player 1 at online mode.

FIGURE 4. (b) The player 2 at online mode.

The ‘Inventory’ option consists of options like Characters, Weapons, and Masks to be chosen by the players. In short, customizations have been made available in Inventory.

The following are the sample weapons and masks that have been included shon n Fig 5 (a), (b), and (c).

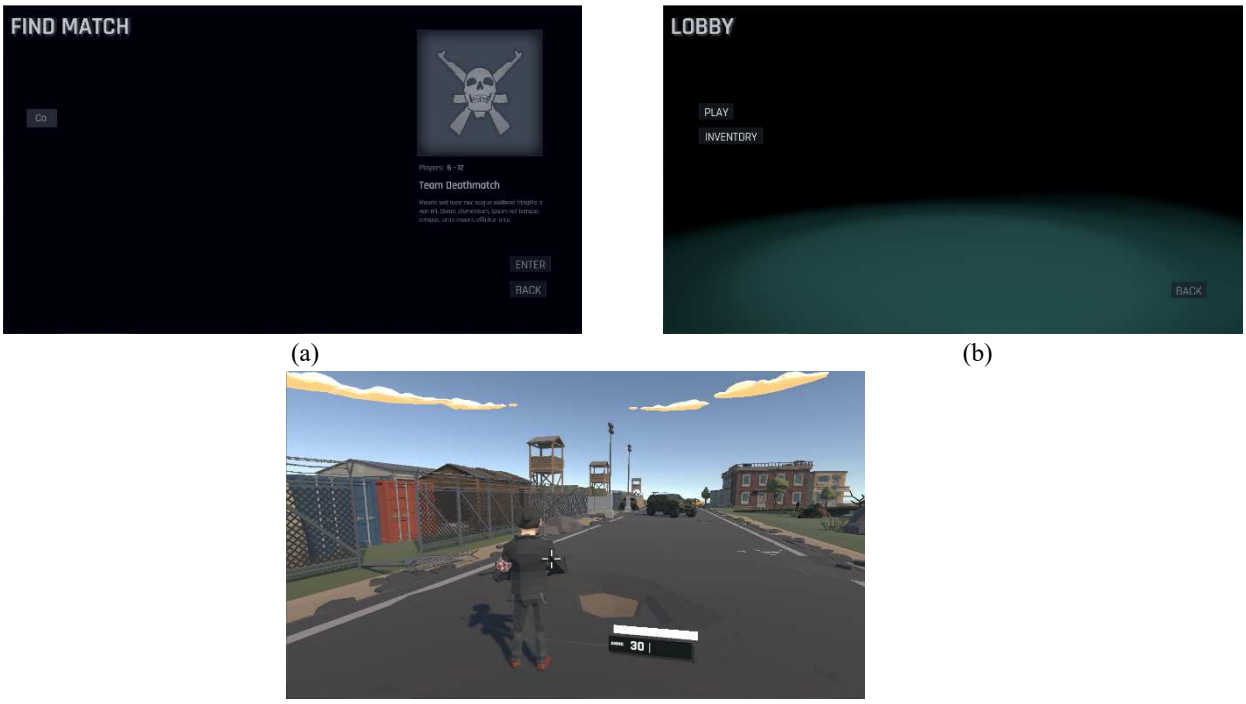
(a)

(b)

(c)

FIGURE 5. (a), (b), and (c) sample images of weapons and mask.

To initiate the multiplayer game mode, the player has to either create a room or join an already created room. On selecting Multiplayer mode, the following is displayed. If the player creates a room, then he/she selects ‘Co’ and clicks enter. The player is then redirected to the lobby where he waits till the other players have joined sample pictures are shown in below Fig 6 (a), (b), and (c).



(a) (b)

(c)

FIGURE 6. (a), (b), and (c) sample images of finding match, lobby and player.

Only the creator of the room can start the game. Once the play button is clicked, the game begins. Every player can see the status of his/her health bar which is represented with a white bar. Also, the count of bullets or his existing ammunition is shown below the health bar. However, the ammunition and health bar of one player is kept hidden from the others.

CONCLUSION

Game development does not include just the creation of graphics, animations, sound effects, etc. The procedure can be executed successfully only if there is enough knowledge about the game engine being used like how it works, what properties it has, and what kind of objects does it use. The construction of a model and the ways to animate the model also can be understood. Various scenarios such as the concept of a game engine, its abilities, its uses, animations and graphics, interactive experiences, fine art, and software-hardware interaction have been explored and understood. Also, this project has been tested destructively presumably to the best of our technical knowledge.

Future Enhancements

The following have been included for the future improvements of the developed game.

- New advanced levels.
- Improvement of the Graphical Representation.
- Introducing new game assets and creating new game levels.
- Introduce new features.
- Create a cloud-based ranking repository to store details for logins.

REFERENCES

1. Blackman S. (2011) Navigation and Functionality. In: Beginning 3D Game Development with Unity. Apress, Berkeley, CA.
2. Blackman S. (2011) Cursor Control. In: Beginning 3D Game Development with Unity. Apress, Berkeley, CA.
3. Paula Escudeiro, Nuno Escudeiro, Marcelo Norberto, Jorge Lopes, Serious Games, Interaction, and Simulation, vol. 161, pp. 42, 2016.
4. H. Basori, R. Sarno, and S. Widyanto, "The development of 3D multiplayer mobile racing games based on 3D photo satellite map," 2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08), Surabaya, 2008, pp. 1-5.
5. Ramiah Chowdary, P., Challa, Y., Jitendra, M.S.N.V. (2019).“Identification of MITM Attack by Utilizing Artificial Intelligence Mechanism in Cloud Environments” Journal of Physics: Conference Series, 1228 (1),012044.
6. Thota, J.R., Kothuru, M., Shanmuk Srinivas, A., Jitendra M, S.N.V. (2020) “ Monitoring diabetes occurrence probability using classification technique with a UI” International Journal of Scientific and Technology Research, 9 (4), pp. 38-41.
7. Jitendra, M.S.N.V., Radhika, Y. (2020). “A review: Music feature extraction from an audio signal” International Journal of Advanced Trends in Computer Science and Engineering, 9 (2), pp. 973-980.
8. Mukkamala S N V Jitendra, Shanmuk Srinivas Amiripalli, Vishnu Vardhan Reddy Kollu, P Ramaiah Chowdary, R Venkata Rao. (2020). Analysis of Airline Connectivity System using Graph Theory. International Journal of Control and Automation, 13(4), 77 - 84.
9. Mukkamala S N V Jitendra, P Ramaiah Chowdary, Shanmuk Srinivas A, R Venkata Rao, Ananthasetty Vedavyas Sai. (2020). Music Listener Mood Prediction from the Lyrics Using Machine Learning. International Journal of Advanced Science and Technology, 29(05), 7760-7768.
10. Amiripalli, S.S., Venkatarao, R., Jitendra, M.S.N.V., Mycherla, N.M.J. (2020) “Detecting emotions of student and assessing the performance by using deep learning” International Journal of Advanced Trends in Computer Science and Engineering, 9 (2), pp. 1641-1645.
11. J. Jie, K. Yang, and S. Haihui, "Research on the 3D Game Scene Optimization of Mobile Phone-Based on the Unity 3D Engine," 2011 International Conference on Computational and Information Sciences, Chengdu, China, 2011, pp. 875-877.
12. Huijuan Zhu. Virtual Roaming System based on Unity3D. Computer system application, 2012, 10 (21), 36 ~ 39.
13. Lei Cao, Hongtao Dai. Mobile Sales System based on Unity3D Technology. The design and implementation of software, 2014 01 (35), 40 ~ 42.
14. Jinmin Liu, Tieming Ma, Na Wang. Motor Show Virtual Simulation Platform Design based on Unity3D. Journal of heilongjiang August first land reclamation university, 2014 01 (26), 66-68.
15. Xiaoqing Chen, Shaowei Wang. The Virtual Reality Technology Application in Education. Software Tribune (education technology), 2011,12,76 ~ 78.
16. Amory, Alan., et al., “The Use of Computer Games as An Educational Tool:Identification of Appropriate Game Types and Game Elements,” British Journal of Educational Technology . Volume 30, No 4, 1999, 311-321
17. Amory, A., & Seagram, R. “Educational game models: Conceptualization and evaluation,” South African Journal of Higher Education, 17(2), 2003, 206–217.
18. Huang, L., & Gui, B. (2015, January). Research on the Application of Products based on Unity3D. In 2015 International Symposium on Computers & Informatics. Atlantis Press.
19. Handriyantini, E. (2014). Developing Computer-based Educational Game to support Cooperative Learning Strategy. IC-ITECHS, 1, 129-133.
20. Xingjie Wang, Chunhua Li. 3D Virtual City Research and Application Based on Unity3D Platform. Computer technology and development, 2013, 2013 (23), 241-244.