Space Attackers Using Unity Cross-Platform Game Engine

Aditi Pawar[1], Sukruta Lomte[2], Rutuja Kurkelli[3], Miss. Varsha Vetal [4]
Department of Computer Engineering
[1,2,3,4] JSPM BSP, Pune, India

Abstract- *In our project we will be placed in a space setting. The player can control a spaceship at the start of the game and can be changed during the game play. The player can move in 2 degree of freedom. There are three types of enemies like ships, asteroids and boss. The enemy ships will have basic Artificial Intelligent system as they can change their paths while shooting in the direction of the player. The asteroids will follow a straight line in any random direction. The boss will be a scaled up version of other types of enemies which will stay at the screen until defeated.*

I. INTRODUCTION

The game must allow the player to play the game, save and load the progress at any time, have score system to rate player performance. The game will be divided into stages.

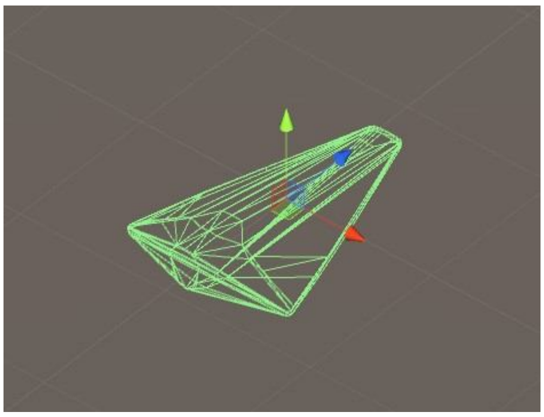
The Player can roam in an Top Down Arcade style when not on a mission. The player controls character movements over obstacles, defeat enemies, reaching end goal to finish one stage. Player character will loss a life or reduces its shields when collided with enemies or lethal obstacles.

1.1 The Player Game Object

Using and Visualizing the Mesh Collider

The Mesh Collider component will not participate properly in physics collisions and will not be visible in the scene view unless we select “Convex” on the Mesh Collider Component.

In Unity 5, the Mesh Collider component needs to be Convex to be able to participate properly in physics collisions. When “Convex” is not selected, the Mesh Collider will not participate properly in physics collisions and will not be visible in the scene view.



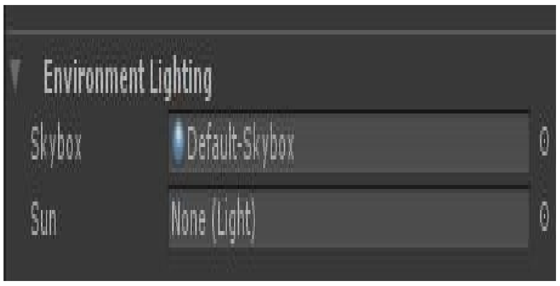
1.2 Setting the Mesh Collider Component options

As well as setting “Is Trigger” to true, we must also make sure (as mentioned in the step above) that the “Convex” value is selected as well.

1.3 Lighting and Camera

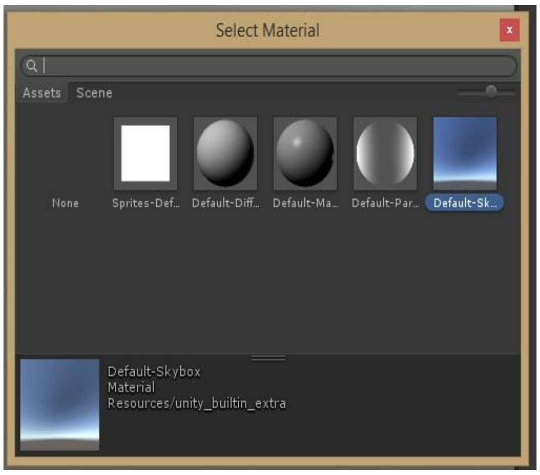
To have no contribution to the lighting from Ambient Light, the Ambient Light source must have no value. This can be done by either leaving the Ambient Source as Skybox and making sure Skybox is None, or by setting the Ambient Source to Color and making sure the color is Black.

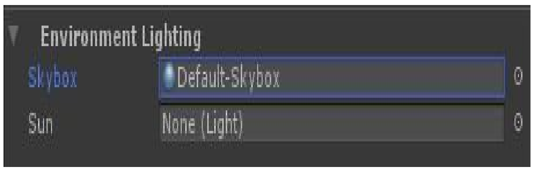
To remove the Skybox, simply delete it. The Skybox field is a asset field like all of the other asset fields in the inspector.



Environment Lighting
Skybox Default-Skybox
Sun None (Light)

You can either select the target button to the right (the circle with the dot in the middle) and it will open up an asset picker window:





1.4 MOVING THE PLAYER

Unity no longer uses “Helper References” to access common components.

In Unity 5 and later we can no longer access components using their “shorthand helper references” and we must access them directly using “GetComponent”.

One example of this is accessing the Rigidbody component attached to the same GameObject as the script. In Unity 4 and earlier, this was simply accessed with “rigidbody.”

Now this must be done with “GetComponent<Rigidbody>().”

It is usually a “best practice” to find this Component when the instance of the script initializes, and “cache” the reference in a local variable.

This is commonly written as: private

```
Rigidbody;
void Start ()
{
                rb = GetComponent<Rigidbody>();
}
```

Now, with the reference to the local Rigidbody component saved in the variable “rb”, we can use this reference anywhere within the script.

One example would be to add force to the rigidbody with:

```
rb.AddForce (someVector3Value);
```

II. GAME ENGINE

In our project we selected Unity3D version 5.6 for development of the project. Unity3D is a powerful crossplatform 3D engine and it isuser friendly development environment. Unity3D is a easy to understand so anybody who want to easily create 3D games and application for mobile, laptop, computer, web etc. create 3D games and applications for mobile, desktop, the web and consoles.

2.1 MOVEMENT OF GAME

Movements in mission is designed in such a way that confine the player to face forward towards the stage progression and control the space required to design the mission towards sure achievement if the player goes through all the enemies.

III. USER INTERFACE

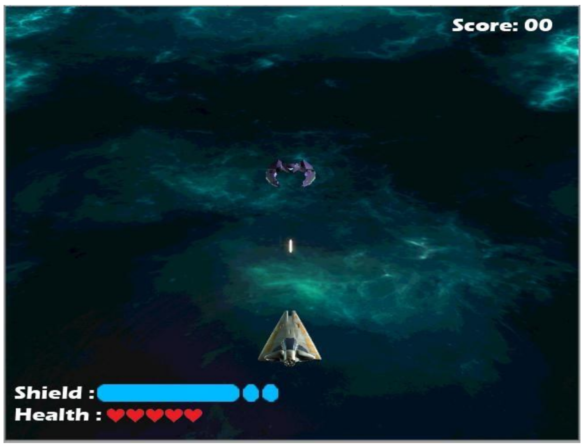


Fig 11: Screenshot during a gameplay

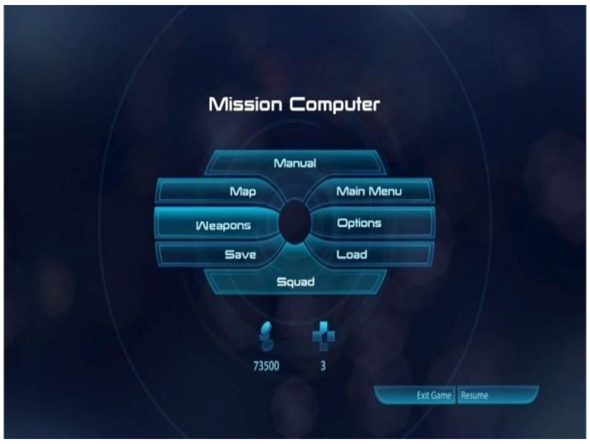


Fig 12: Pause Menu

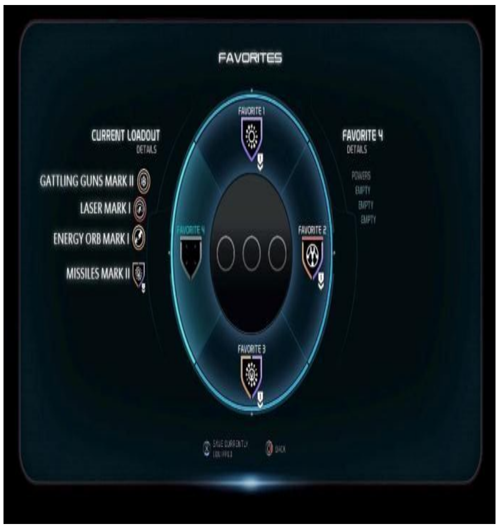


Fig 13: Weapons selection and favorites menu

IV. TECHNOLOGIES USED

4.1 C# Language

(For writing object's dynamic behaviours)

C# (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common

Language Infrastructure.

4.2 Unity3D

Unity is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop videogames and simulations for computers, consoles and mobile devices.

First announced only for OS X, at Apple's Worldwide Developers Conferencein 2005, it has since been extended to target 27 platforms.

V. FUTURE SCOPE

- Round Extension
- Improve Graphical Representation
- Introduce new game features
- Introduce new environment and scenes
- Take user response through websites and produce web rank list.

VI. CONCLUSION

A Software project means a lot of experience. In this section we summarise the experience gained by project team during development of “Space Attackers”.

Working with game engine completely a new experi ence for us. Normally we are working with different OO languages, DBMS, mark up languages etc. We adopt these things by video tutorials, text tutorials, internet and learning materials given by the tools themselves. It’s a matter of time, patience and hard work.

It is very sensible work and it demands much time because the game engines try to connect game environment with the real world. Creating a 3D model is very difficult because you need to work with each and every point of the model. The Exists game engines demands vast knowledge about its properties, sections and subsections. After all the thing is that a game project is not a project of 6 or 8 months for three people!

REFERENCES

[1] R. Galantay, et al., "living-room: Interactive, space orientedaugmented reality," 2004, p. 71.

[2] K. Kim, et al., "ARPushPush: Augmented reality gamein indoor environment," 2005

[3] M. WEILGUNY and D. MEDIEN, "Design Aspects in Augmented Reality Games," 2006.

[4] AKENINE-MÖLLER, T., HAINES, E., and HOFFMAN N. 2008. Real-Time Rendering. Third Edition, Boca Raton, CRC Press.

[5] BLINN, J. 1978. Simulation of Wrinkled Surfaces. ACM SIGGRAPH Computer Graphics, vol. 12, no. 3, pp. 286–292.

[6] NYSTROM, R. 2014. Game Programming Patterns. Genever Benning, ch 2.