

DEVELOPING A FPS GAME FOR ANDROID

Extraterrestrial:Influx

Soham Khopde^[1], Gauri Hande^[2], Mrudula Pawaskar^[3], Hasib Shaikh^[4], Amarja Adgaonkar^[5]

¹Engineering Student, ²Engineering Student, ³Engineering Student, ⁴Assistant Professor, ⁵Professor

¹Computer Engineering Department,

¹K. C. College Of Engineering and Management Studies and Research, Mumbai, India

Abstract : The development of First Person Shooter games started in the late 1980s and it developed more with advanced 3D graphics in 1995. Later on, these games became popular online with multiplayer battles and verbal and non-verbal communication in games. The application of these games widened in more areas like the military. For close combat situations, the military needs to be trained in an advanced way. Using FPS virtual 3D games military can help to improve their reflexes and concentration to tackle their enemy. By using these techniques we can develop any structure of architecture in the virtual 3D game. Techniques implemented in the game can be used for simulating and predicting threat in dangerous situations. The First Person Shooter games are based on the 3D environment so the representation of these games are more expressive than 2D games. In First Person Shooter game the user can view the game scenario from the character point of view. There are different software for developing fps games for android like Maya and unreal game engine. This paper describes the techniques used for modeling in 3Ds max and further developing it in unity game engine. Initially, the game artist has to sketch the models involved in the game in 2D. Before the development of game begins, game story and levels in the game are predetermined

IndexTerms - First Person Shooter, 3D Modeling, Unity Game Engine,

I. INTRODUCTION

A. Game Story

This project is based on android 3D gaming. The game is a first person shooter type of game. It includes various missions and difficulty level increases after every checkpoint in the mission. The game will have three planets and each planet will have various missions. The game will start with a story line of alien invading or colonies on other planet. A rescue team from earth will be sent to the captured planet and the user have to destroy aliens and recaptured the map. The game will also include star ship and warship with fabulous 3D graphics. User will control the character with onscreen joystick and buttons. The bots will assists user character and help the capturing the map. The character weapons will be nuclear powered and advance armour will be applied. The game will be based on fiction and imagination. On other planets environment will be different from earth with low gravity. Innovative armour, arms and ammunitions will be applied. Weapons included in the game will be innovative and never existed before. Jetpacks will be applied to Characters in game.

B. Softwares

For creating a 3D android game we require Modelling software and Game Engine. There are different modelling software's available for 3D modelling like blender, Maya, 3Ds Max. There are also different game engines available like Unreal Engine, Unity, Cry Engine. From above Modelling software's we are using 3Ds Max. 3Ds Max is an industry standard for 3D modelling. It is a user friendly software to create 3D models compared to Maya and blender. It is easier and smoother to work in 3Ds Max than Maya and Blender. Modelling in Maya is complex than 3Ds Max. 3Ds Max allows a fast workflow and robust modifiers than Maya. It has all poly modelling tools in an entire UI. 3Ds Max is best for architectural Visualization when compared to other modelling tools. Compared to Unreal Game Engine, Unity has a very sophisticated user interface. Unity has got a great assets store compared to Unreal Engine. Game Logic and game rules and prototype gameplay is much cleaner and consistent compared to Unreal Engine.

C. Prerequisites

The game artist has to sketch the models before they are forwarded to a modeling artist. The game artist has to sketch the 2D model from front view, side view, back view. The game artist also has to design expressions, walk and run cycle and many other movements. These designs are used by a modeling artist as

a reference to convert the 2D model into the 3D model. The game artist uses software like SketchUp for designing purpose.

II. RELATED WORK

FPS is a video game which has a gun in hand from first person perspective. The player experiences the video game through the eyes of a protagonist. There are many FPS games like Counter Strike, Half Life. These mentioned games are based on realistic physical environments and are not adventures and are concentrated on more target shooting purposes. These games does not contained simulation of vehicles and no innovative armors and weapons are used. Enemy's appearance is not unique and used in Hollywood action movies and user may not get curious about his next mission.

III. CONSTRUCTION

1. 3Ds Max

In the initial phase of game development we have used a 3D modeling software namely 3Ds Max. 3Ds max software is used to construct a different 3D Models like Characters, Trees, Animals in a game. 3Ds max software is used for various purposes like modeling, texturing, rigging, skinning, and animation.

2. Modeling

3D Modeling is a process of producing 3D models from basic 3D Geometric shapes like Cube, Sphere, Geosphere. To model these shapes to a game object different techniques like polygonal modeling, NURBS (Non-Uniform Rational B-Splines), Mesh Modeling are available. In this project we are going to implement polygonal modeling technique.

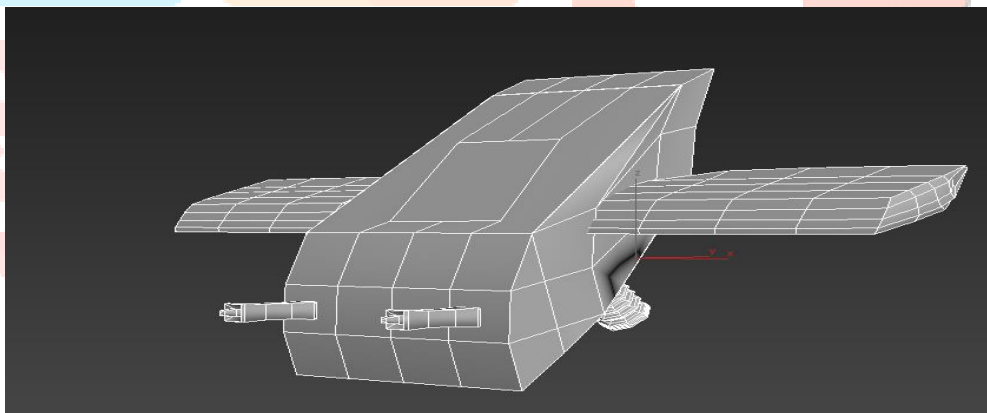


Fig: 3D model of space ship

a. Polygonal Modeling

To develop a model, a model sheet is taken as reference and a geometric shape is model according to reference. To develop a model using polygon modeling, there is a modifier in 3Ds max called as edit poly which simplifies polygonal modeling. The edit poly modifier in 3Ds max has five modes namely vertex, edge, Border, polygon mode and element mode.

3. Texturing

Texturing a 3D model is a process of applying materials like skin, clothes or colour to a model. This techniques includes creating a 2D map of textures and applying it to a 3D model using material editor in 3Ds max. For texturing in 3Ds max there are two modifiers namely unwrap UVW and UVW modifier.

The unwrap UVW modifier helps to texture a 3D model with different materials to its different sides. This modifier helps to create a 2D map of 3D model. The 2D map is imported to Photoshop to apply material on it and then it is applied to 3D model in 3Ds max.

4. Skinning And Rigging

Rigging is a process of creating a suitable bone structure for 3D module. It is used to defined a bone for each part of the mesh. It determines the movement of each part of the mesh. There are different techniques to rig a model in 3Ds max like Bipedal system, Cat system for animals and custom bones. Bipedal system is used for 3D human models and Cat system is used to rig animal model.

Skinning is a process of adding bones to a model in 3Ds max. The skin modifier in 3Ds max is used to deform a mesh using bones. Skin modifier creates an envelope for each bone which affects to its related part of the mesh.

5. Controllers and Animation

Controllers provide powerful tools for controlling an animating models in the scene. Controllers have different constraints like path constraint which can be used to animate objects without using key frames. The movement of character model is animated using controllers. A model can be controlled using helpers in 3Ds max. Constraints are used to control the position, rotation and scale of an object in the scene. Animation can be achieved using different ways in 3Ds max like autokey, constraints. To animate a character, references of its movements are required like walking, running, jumping. In 3Ds max there are inbuilt walk cycle for Bipedal system. A Bipedal system can be animated using existing bip files using motion mixer tool

6. Texturing In Photoshop

The UVW map of a 3D model is exported in png extension. As you can see in below image it is a UVW map of a 3D model. The green lines are the outlines of the 3D model. Using this technique we can texture different parts of 3D model using multiple materials. The material should cover the green outline of the UVW map. After applying the material to the UVW map texture, you can save it in png extension. The png image is ready to apply to 3D model



Fig: 3D model of plasma gun

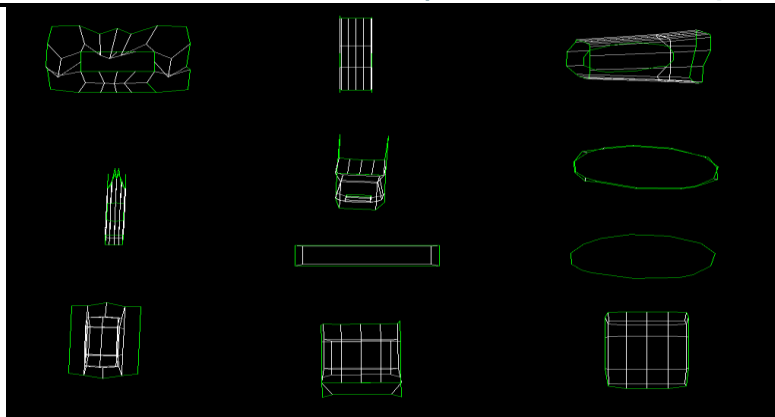


Fig: UVW map of 3D model

IV. IMPLEMENTATION

1. Unity

Unity is a game engine which can be used to develop three-dimensional and two-dimensional games. For computer, play station, X-Box, IOS Android. Unity game engine uses scripting languages like java script, C#, Boo Language. Unity game engine has OpenGL graphics API, Direct 3D. It has built in support for PhysX physics engine which provides realistic effects. It is easy to deploy, optimize a game in it. Unity game engine provide standard assets for different controllers like character controller, vehicle controller, Particles systems, camera controllers.

2. Game User Interface

A game user interface has different buttons like fire, jump, joystick, inventory, pause, resume, play, exist. An on-screen joystick is use to control the movement of the character in XYZ axis in the game. A touchpad with a convenient sensitivity is registered on the screen. A touchpad controls the rotation of first person character in XYZ axis. The jump button is use to make the player jump and fire button is use to shoot the non-player characters. In the above image there is an inventory button which is use to pick up or change objects from the hands of player. The pause button will display a canvas menu on screen which will display main menu button, restart button and resume button.

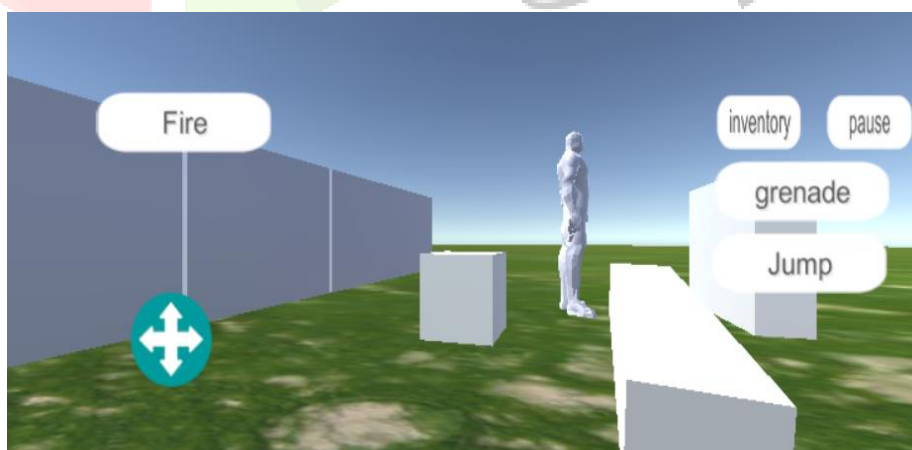


Fig: game user interface

3. Animating objects in Unity

To animate objects in unity, you need to import FBX files from 3Ds max. This FBX files contains different animation clips of the objects. This FBX files are save in assets folder of unity. A animator component should be attached to game object in unity. The animator component should be attached to an animator controller. The animator controller has references to one or more animation clip of the game object.

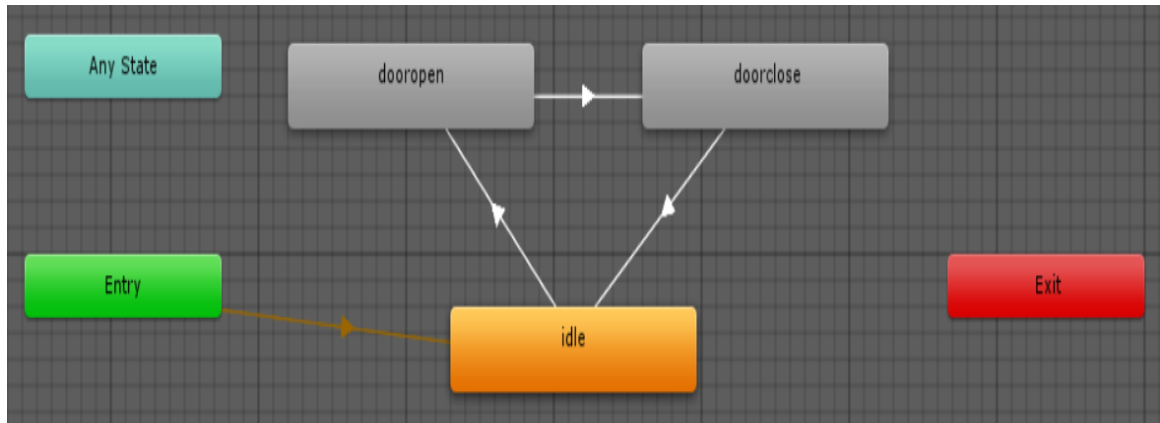


Fig: animation clip of door model

The above image is an animation clip of a door open state in the animator file. The above animator file has 3 states idle, open, close. The animated file has triggers. The triggers are used to decide when an state of animation should be executed. In this case there are two triggers open and close

States and Trigger in Animator :

- When the open trigger is selected then door open animation will be executed.
- When the close trigger is selected then door close animation will be executed.
- When no trigger is performed then the door will be idle state of animation.

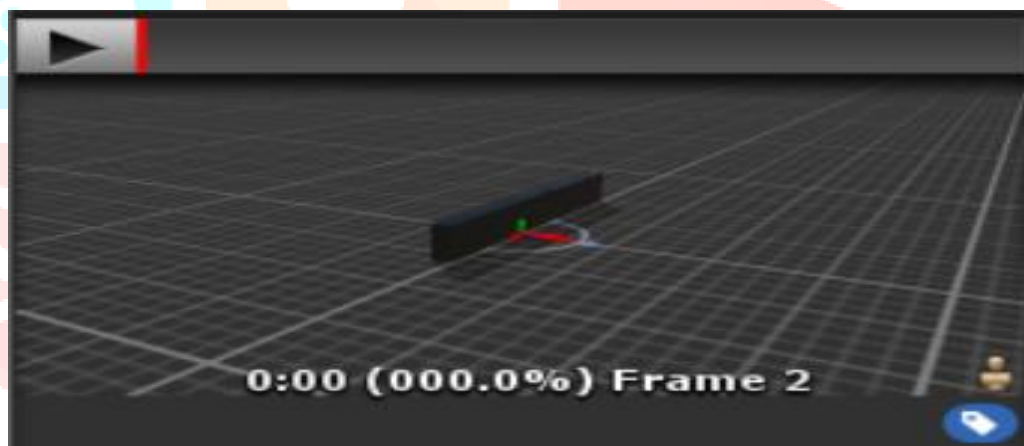


Fig: States of door

4. NPC

NPC is any character player cannot control. A game developer can add a behavior to NPC like run, jump, attack, defend, dodge. The navmesh agent is responsible for to move a NPC in a scene. It helps an NPC to find a path towards a target which should be present in its radius. The navmesh agent is a simplified geometrical plane which enables NPC to navigate towards various objects in the scene. The navmesh obstacles are use to block NPC to reach their target. The aqua blue part of the scene is the navigation area of the navmesh region in the above region.To implement NPC a prefab is required. The prefab should be a rigidbody object in unity.

5. Materials In Unity

Materials are used to define how an object will be displayed in the scene. The properties of the material determine by its shader. A shader displays a combine output of the texture and lighting of the object after it is rendered. Textures in unity are images which are given as reference to a material of a game object. Textures determines the reflectivity and roughness of material.

6. Particle Systems in Unity

A particle system is made of small images or meshes that are displayed and spreads widely in the scene of unity. Each particle, in particle system has a determined lifetime of few seconds. A particle system emits particles at random position in different shades like box, sphere, cone or custom shape. An emission rate in the particle system decides how many particles are emitted per second. The velocity of a particle can be changed by external forces and gravity applied to the system. The particle system can be used to represent a waterfall, smoke from a fire, dust particles blown by a wind zone.

V. CONCLUSION

In this paper we have presented the techniques to develop a 3D FPS game. This section will describe a brief summary of all techniques mentioned in the different sections of the paper. In Section I, we have mentioned the game story, pre-production process and justification of softwares used for development. In Section II, we have discussed about the versatility of our game and already developed games. In Section III, we have mentioned techniques used to create 3D models which will be implemented in the game. These techniques include modeling, texturing, skinning and rigging, animation. In this Figure (A) is an example of a model developed for the game. Further in Section IV, we have included developing a game in Unity Game Engine. It includes visual of Game User Interface, animating objects in Unity, implementation of non-player characters in the scene, materials, shader and particle systems in Unity.

VI. ACKNOWLEDGMENT

We express sincere thanks to our guide Prof. Amarja Adgaonkar and co-guide Prof. Hasib Shaikh whose supervision, inspiration and valuable guidance helped us a lot to complete our work. Their guidance proved to be the most valuable to overcome all the hurdles in the fulfillment of this paper work. Also we are thankful to all those who have helped us in the completion of paper work.

VII. REFERENCES

1. In-game adaptation of a navigation mesh cell path by [Thomas P. Hartley](#) School of Technology, University of Wolverhampton, UK, WV1 1EL [Qasim H. Mehdi](#) School of Technology, University of Wolverhampton, UK, WV11EL
2. C# interpreter and unity 3D for educational programming games by [Paul Smith](#) School of Technology, University of Wolverhampton, UK, WV1 1EL [Thomas P. Hartley](#) School of Technology, University of Wolverhampton, UK, WV1 1EL [Qasim H. Mehdi](#) School of Technology, University of Wolverhampton, UK, WV1 1EL
3. Research of Traffic Safety Simulation Based on 3DS MAX by [Wen-bo Wang](#) Dept. of Comput. Sci. & Technol., Dezhou Univ., Dezhou
4. Research on the Application of 3DSMAX in Animation Design by Dan Zhang College of Arts University of Science & Technology Qingdao, China qdzhangdan@163.com
5. Automatic Rigging and Animation of 3D Characters by Ilya Baran & Jovan Popovi Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology
6. UNITY 3.X GAME DEVELOPMENT ESSENTIALS
By Will Goldstone
7. GAME DEVELOPMENT WITH UNITY, SECOND EDITION
By Michelle Menard, Bryan Wagstaff