ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 8, February 2013

Android 3D Game with Motion Sensor and Touch Screen Control

Harshil Bhatnagar, Kirtikumar Bansod, Yashi Verma, Divya Naidu, Amreen Khan

Abstract:- This abstract describes an arcade style 3D game which will run on the Android operating system.. The objective of game is to drive a space jet dodging several continuous series of obstacles. The actual game play will be in a 3-Dimensional view. The game contains obstacles of fixed size which will be rendered in 3D in open area and the player has to control the space jet such that it will not collide with any obstacle. As the game player moves forward without collapsing, the high score will keep on incrementing. And once the space jet collides with any of the obstacle, each time the high score will decrement by a factor and a slight push back with a small vibration and music effect will occur. This application uses two user interfaces that is, Motion Sensor and Touch based control. Motion sensor will simulate the real physical effect that is impossible to provide through joystick or any other gaming device. It will use OpenGL game engine which will provide facilities such as graphics, sound, physics, AI functions & 3D rendering capabilities.

Index Terms: **Android, Motion Sensor, 3D Game Open GL, Touch Screen Mode.**

I. INTRODUCTION

Mobile games are played using the technology present on the device itself .Technology includes Accelerometer, Game Engines, Touch Screen, Screen Resolution, GPS, API etc[5].This paper contains development of an android game which includes 3D rendering of objects, Motion sensors(accelerometer) in dual mode as well as touch screen using OpenGL ES 1.0 version [2]. The objective of game is to drive a space jet dodging several continuous series of obstacles. The actual game play will be in a 3-Dimensional view. As the game player moves forward without collapsing, the high score will keep on incrementing. And once the space jet collides with any of the obstacle, each time the high score will decrement by a factor and a slight push back with a small vibration and music effect will occur. Also two slight vibrations will also occur by the device as a notification. This application uses two user interfaces that is, Motion Sensor and Touch based control. The area in which the space jet will move is of four prism high and wide. The area is defined by a 12 x 12 matrix of channels, each of which contains prisms of same structure [14]. The obstacles are prisms occupying fixed channels.

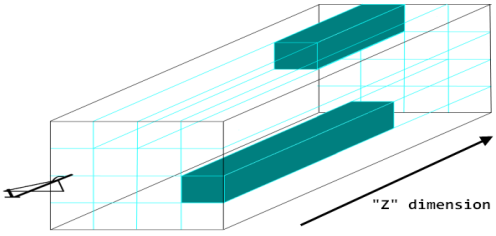

Fig 1. Rendered Objects

Two levels will be offered in the beginning of the game that is Easy and Difficult. The easy level will have few no. of obstacles, lesser speed. The difficult level has a different background from the first level, and also has more obstacles. Accordingly the user's score increments more quickly in the second level. Picture of second level game play is shown below:

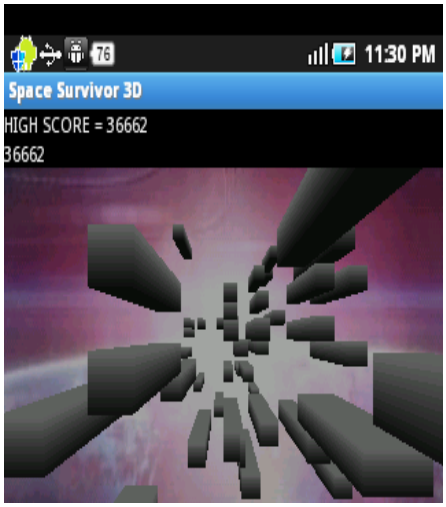

Fig 2: Easy Mode

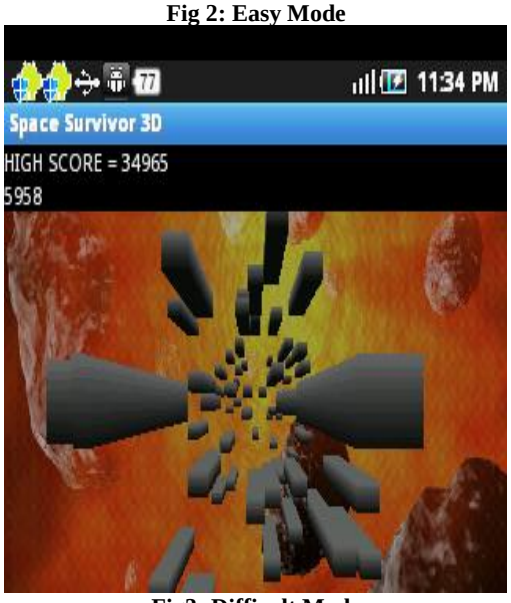

Fig3: Difficult Mode

II. PRESENT WORK

A. User input:

The button-based GUI that prompts the user for these pieces of information is landscape-oriented, as this is the mode of play that is preferred. An image of this button-based GUI is shown below:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 8, February 2013

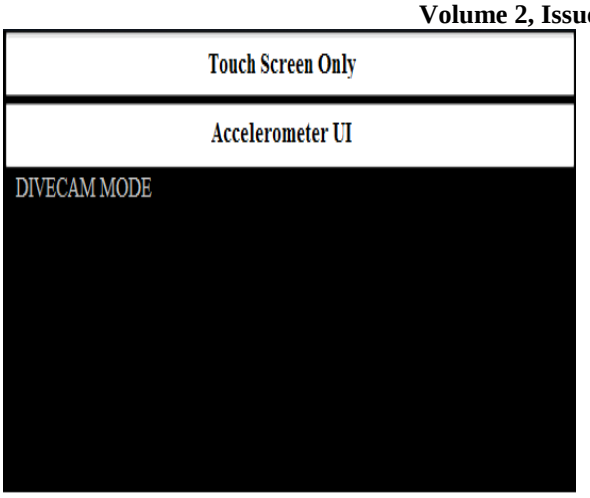

Fig 4: Game Modes

1) Touch screen:

The touch screen input mode operates in a manner that should be familiar to Android users. The touch screen user interface is built around whole drag events using fingers. To direct space jet in X direction it has to drag the screen left-right and vice versa for y direction. If the player attempts to go beyond the boundary of the game play, a vibration will occur as a notification that the user can't go beyond this position.

2) Motion Sensor:

In this mode player has to hold the handheld device in landscape mode. One can compare this style as holding yoke of an airplane and controlling its direction to create a sort of virtual reality. This mode is treated as "Vertical Mode" As shown in the image below.

Fig 5: Vertical Mode

Another mode "Parallel mode" which is a simulation of a driver holding underwater camera and pointing it in the direction where the driver moves. The user can play the game by holding the device parallel to the ground. This is not supported by many motion sensor games in android platform.

Fig 6: Parallel Mode

The selection between "Parallel" mode and "Vertical" mode is made by the user at the starting of the game. The activity transfer to the main game will not proceed until the device is held in right position. Portrait mode play is possible but landscape mode is more interesting as it covers most of the area of the game play. Also," perpendicular mode" is not available when the device is held in portrait orientation.

B. Prism series and lifetime:

The obstacles in the game will be displaying in a continuous manner. Obstacles are shaped like something obstruction in space. The prisms which are treated as obstacles are generated in the form of a group which is near proximity to the user when application gets starts. Another similar group of obstacles would be generated when the first group of obstacles got passed .This process creates a stream of obstacles which would be continuous and user will pass through the gaps of obstacles.

C. Camera Work:

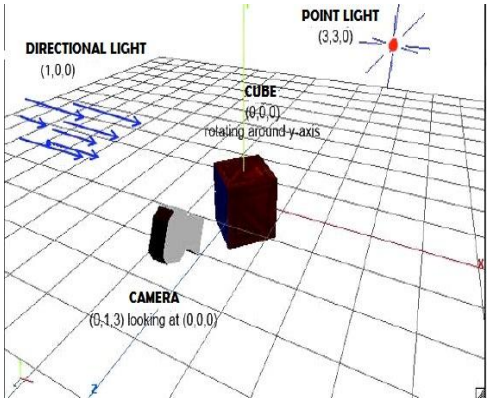

Fig 7: Camera System

The camera system of game will consist of two modes. The distance from the camera to the near clipping plane, the distance from the camera to the far clipping plane., it will represent its position in world space, it's rotation around its x-axis (pitch) , it's rotation around its y-axis (yaw), this .Its rotation around its z-axis (roll), . Same camera system is used in this game in which every object looks differently at different angles.[13],[14]

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 8, February 2013

D. Role of Light:

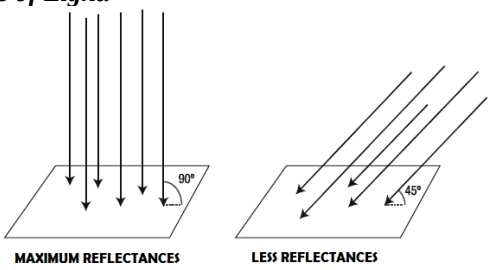

Fig 8: Lightning

By using lightening in OpenGL ES the game will have a nice 3D touch. The light source's type. The light source's color or intensity. The light source's position and direction relative to the lit object and the object's material and texture depending on this factors lightening can change the color of an object. In the same way the objects which have been used in this game are having different shades at the different level of visual perception. [13]

E. Background:

The background which we have used in the game play will be constant this can be accomplished by using flat OpenGL plane at a fixed distance from the camera. The obstacles will occupy the specific part and the remaining part will be occupied by the flat plane. The game will have a background music which will make the game play more interesting, as well as vibrations whenever a collision is made to give a real effect during the gameplay.[14]

F. Process Control

In android operating system if any application gets started then the process will be organized in the order that it do not usually exit once started. Rather the application will get suspended and will get resume at some other point. This application will belong to this process management of getting suspended and restart. Full credit is given for high scores set. An accidental application kill can results into loss of current high score.

III. RELATED WORK

Many game developers have worked on the android platform to develop 3D games. Robert Green [14] has been working on this specific field and is continuously trying to develop 3D games on android platform; For developing games we must choose the language on which we have to work. We can work on a specific programming language Android is a Java-based environment. This is nice for new developers as Java is widely accepted as a much easier language to get started in than C++, which is the norm for mobile development and is what he use now. Google has also done an excellent job with documenting the API and providing examples to use. There is an example to show functionality for almost 100% of the API, called API Demos. In order to get the higher learning curves and just to solve competing complexity on performances ,Robert Green has chosen 3D game platform.2D games have a much lower learning curve and generally are easier to get good performance.3D games require much more in-depth math skills and may have performance issues if developer is not very careful. They also require the ability to use modeling tools like 3D Studio and Maya if intend to have shapes more complex than Boxes and Circles. Android supports OpenGL for 3D programming and there are many good tutorials on OpenGL that one can find to learn it For building a high quality models coding practice is necessary that applies to all software development but it is particularly important in game development. Debugging can get very difficult in a stateful, real-time system. Keep methods small and the general rule of thumb is that each method should have 1 and only 1 purpose. For programmatically drawing a background for a scene, there may want a method called "draw Background." Things like that will make the methods easy without making it too complex to understand. Dan Ruscoes [13] stated Android is the first mobile platform he has developed for and getting started was a lot easier than he expected it to be. He has been using Ubuntu, Linux, running Eclipse with the ADT Plugin and testing on an HTC Evo 4G as well as several emulators. The Android emulators, perform terribly when testing games. Despite this, they are very useful when testing a game for various screen resolutions. A Motorola Xoom emulator helped to fix an issue in one of the games. The online Android documentation is excellent and plenty of tutorials for specific concepts can be found via a quick trip to Google. Martin-Dorta [1] brought attention to the pedagogical potential of mobile devices, and various experiments have tested tools in handheld devices, indicating that they strengthen and support learning in different fields. The purpose of the work presented in Martin-Dorta paper was to design a 3D construction mobile game for 3D spatial visualization training, with the aim of familiarizing users with a 3D environment and improving their understanding of the 2D-3D relationship. Martin's paper describes the architecture of the system that they developed, which includes several different applications: A mobile game played on touch screen devices to permit students to solve exercises about building models with unit-sized cubes in a 3D environment, **A** PC application, to permit the teacher to design exercises and manage users, and a server application and a database to collect and manage data. The J2ME Mobile 3D Graphics library is used and development is done in Net beans.

.

IV. CONCLUSION

Our experience with Android and its OpenGL implementation has been positive. This implementation is a powerful and accessible one. The Eclipse IDE and Java language are economical and familiar choices, whose operations and semantics should be familiar to many developers. The results show a high degree of satisfaction with the game and that users realize how useful this game can be for the proposed aims. The accelerometer and touch screen input devices are somewhat more novel technologies. We hope that the game here offers some useful techniques for dealing with these forms of input in future game development.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 8, February 2013

REFERENCES

[1]. A 3D Educational Mobile Game to Enhance Student's Spatial Skills by Martin-Dorta: (IEEE PAPER, Date of Conference: 5-7 July 2010).

[2]. Design and Implementation of the Game Engine Based on Android Platform by Wu Yan-hui (IEEE paper, Date of issue: 16-18 Aug. 2011).

[3]. http://www.droidnova.com/android-3d-game-tutorial-part-i,312.html.

[4]. http://android-developers.blogspot.in/2010/06/game-development-for-android- quick.html.

[5]. http://en.wikipedia.org/wiki/Game_engine

[6]. http://www.androidapplicationdevelopment.biz/android-3d-game-development.html.

[7]. http://www.javacodegeeks.com/2011/06/android-game-development-tutorials.html.

[8]. http://p-xr.com/android-tutorial-opengl-3d-game-development-basics

[9]. http://www.rbgrn.net/content/215-light-racer-3d development-journal

[10]. http://www.flipcode.com/archives/3D_Geometry_Primer_Chapter_2-Issue_01_Appendix.shtml

[11]. http://ruscoe.org/indie-game-development-for-android/

[12]. http://www.flipcode.com/archives/3D_Geometry_Primer_Chapter_2-Issue_01_Appendix.shtml

[13]. Adventures in Indie Game Development for Android by Dan RUSCOES

[14]. Getting started in android game development by Robert Green:

[15]. Beginning Android Games-Mario Zechner

[16]. Learning Android Game Programming: A Hands-On Guide to Building Your First Android Game by Richard A. Rogers.