TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

A

Report On

GURKHA ARMY

(A 3D Indoor Game)

Submitted To:

Department of Electronics and Computer Engineering

Pulchowk Campus

Submitted By:

Kiran Timsina (064/BCT/515)

Krishna Prasad Panthi (064/BCT/516)

Suraj Niroula (064/BCT/546)

Date: 25/01/2011

PREFACE:

Many 3d games have been developed with advanced graphics effects. The world of 3d game programming seems to be fascinating, challenging and full of curiosity. So, comes the concept of yet another 3d game with interior environment in first person view which seems to be the basic replica of existing 3d first person shooter games like Quake3D, DOOM series, etc.

Hence, to make such a game, we focus on developing an interior environment inside which a player can move around to complete the given mission of the level by killing the opponents.

ACKNOWLEDGEMENT

We must acknowledge our deep sense of gratitude for the Department of Electronics & Computer Engineering for offering Minor Project as a subject in our syllabus. We are also highly obliged to Dr. Aman Shakya for his valuable assistance and guidance during the development of project.Thanks are due to Er. Manoj Ghimire for his valuable suggestions.

We are also thankful to all other groups doing similar project in 3d games for their active discussions of common problems faced and sharing the probable solutions.

We must acknowledge our appreciation to all the people who directly or indirectly involved in making this project a successful one.

TABLE OF CONTENTS

ABSTRACT

Many advanced 3D games have been released in market with advanced graphic effects. Such games are becoming popular and popular day by day. The field of so fascinating that we are determined to do a project dealing with 3d game.

The field of 3d game programming is interesting, full of curiosity but challenging. This project aims in developing a first person shooter 3D indoor game.

1. INTRODUCTION

1.1 BACKGROUND

The project is about making a 3D interior game named as Gurkha Army. Interior game implies that the game starts and finishes somewhere inside a room remaining within the premises of a building during the entire course of the game.

In first person games, the camera is positioned such that it gives an illusion that the player himself is traversing inside an environment. Likewise, in this project, the protagonist is a first person who navigates inside a building. The game progresses by handling the key events generated by the end user who simulates the first person actions.

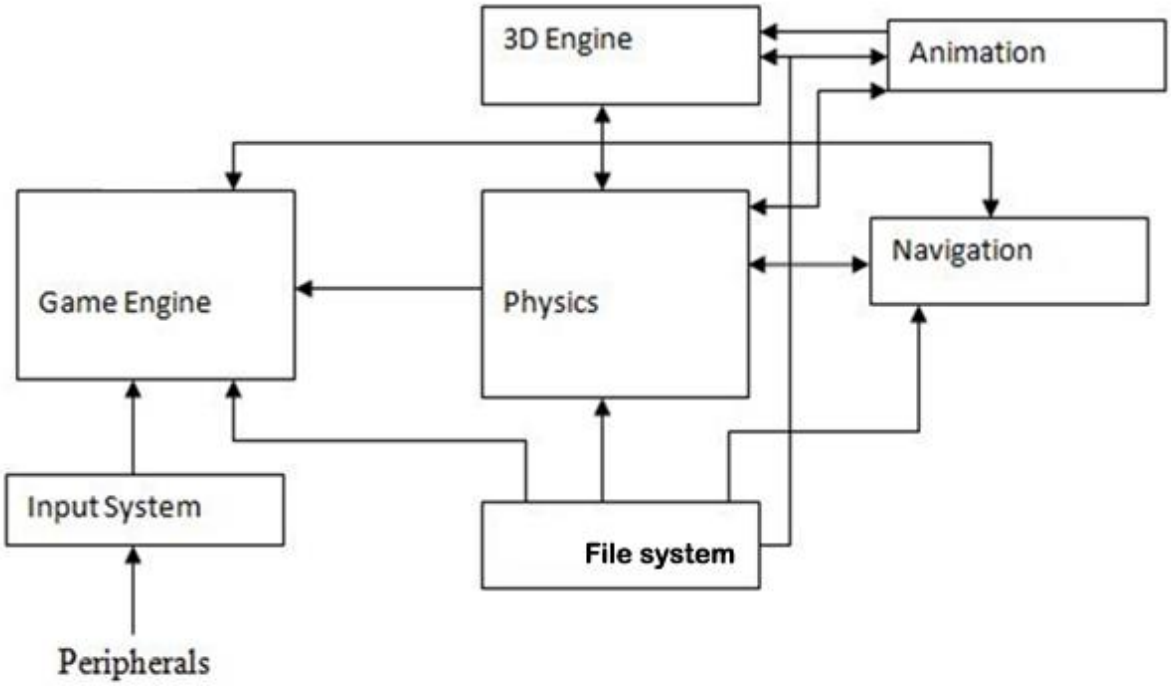
1.2 OBJECTIVES

The major objectives of our project are:

1. To build game engine
    1.1. To manage all resources (such as textures, meshes, and sounds).
    1.2. To handle keyboard input.
    1.3. To manage threads.
2. To develop a 3d modeler that can generate necessary polygons for 3d path by taking top view (in bmp format) as input.
3. To implement the concept of BSP (Binary Space Partitioning) tree for assisting efficient rendering of scene, radiosity and collision detection.
4. To load 3d characters as players of the game.
5. To assign game mission for each level.
6. To embed decision capability in players by the information obtained by real time 3d image processing.

2. DESCRIPTION

2.1 System Block Diagram



2.1.1Game Engine:

Engine, as the name suggests, provides a solid foundation into driving the main concept of any system and thus the desired wrapper can be developed to get unique products. A game engine is an infrastructure for the development of different games. It provides the design perspectives for the game developers and makes development of game easy. Simple FPS game engine constitutes a basic framework (for linking all the component of the engine together), a resource (animated objects, maps, sounds, etc) manager and various forms of control for itself and for the user, rendering handler, sound system.

2.1.2 3D Engine:

In a nutshell, 3D engine is the one that takes control of drawing the scene of game as it progresses.

The scene in the game constitutes of path, 3d characters, game status, etc. All these have to be correctly drawn on the screen to give the current status of the game. All the transformations from the world coordinates to view coordinates are to be handled.

To achieve these, we wrap glaux over openGL.

What 3D engine has to render is determined by the game engine as the game progresses.

2.1.3 Physics

Physics is that deals with collisions between objects. Collision test has to be done between the first person and the walls. Similarly, when the enemies change their state to chase state, collision with the walls has to be handled.

The algorithm for collision detection is implemented from scratch. BSP(Binary Space Partitioning) tree is used to find the current location of character and test for collision with the walls of current leaf is performed to restrict any move that tries to pass through the wall.

2.1.4 Input System

The key events have to be handled to accordingly control the game. Inputs are taken for navigation and firing.

2.1.5 Animation

Animation is used to represent firing state of first person. Animation is also used for different states of enemy such as standing state, firing state, dying state & chase state.
To achieve this, we used 3DsMAX to generate various frames which collectively represent animation

2.1.6File System

The information about path and 3d characters are stored in files in permanent disk.

Information about path is in text file format and that of 3d characters is in 3ds file format.

2.2 THREADING

2.2.1 List of threads

THREADS

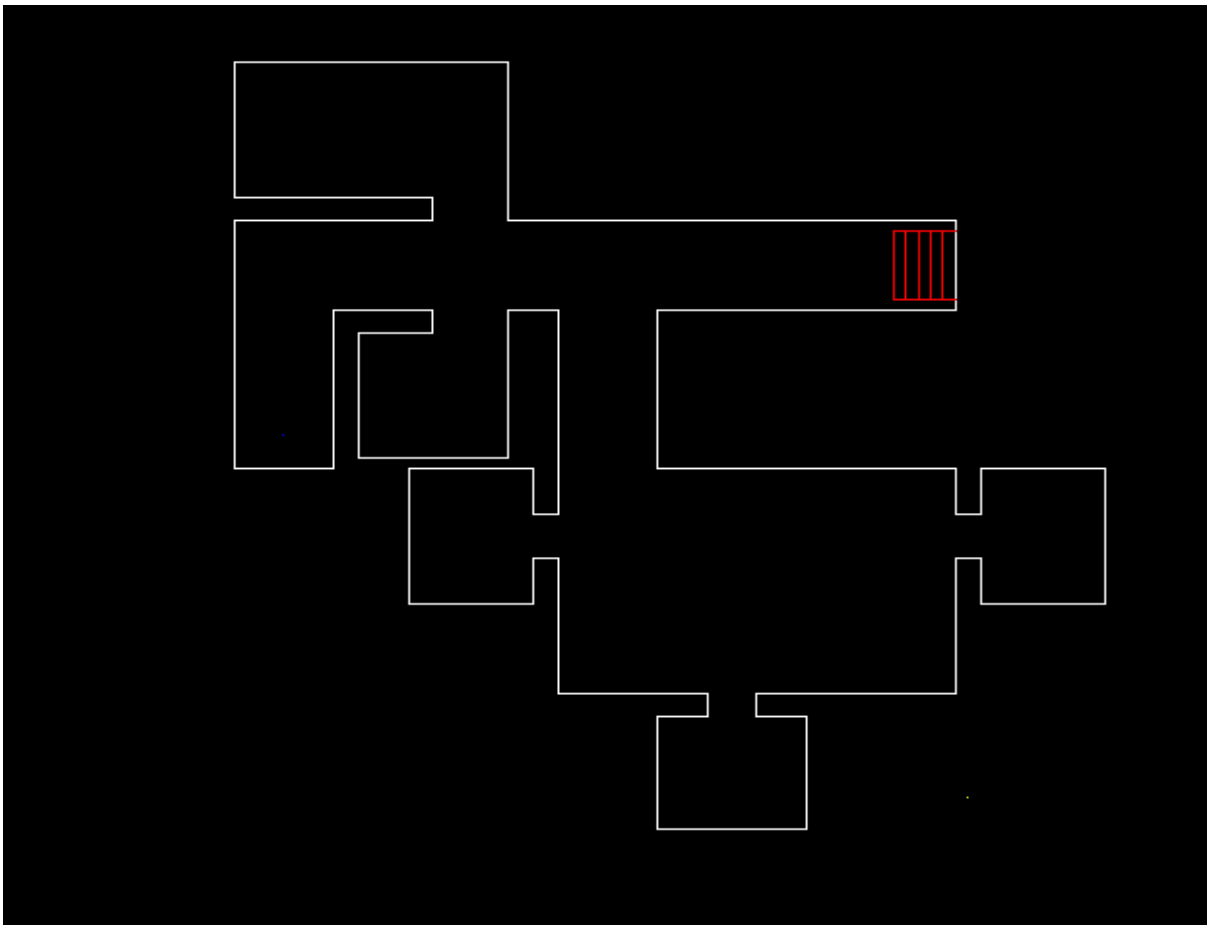
Name	Purpose	Time of creation	Time of destruction	Created By
Main (Render)	Initialisation& Rendering	Beginning of program	End of program	System
Input	Handle keyboard input	Beginning of program	End of program	Main
Sound	Play sound	Beginning of program	End of program	Main
Mesh Loader	Load data about path & 3d objects	Beginning of new game	After finishing the job	Main
Physics	Collision detection	Beginning of new game	End of program	main

2.3PATH

2.3.1 Introduction

- Information about path is found by processing top view of each floor.
- Convention in designing top view
  - White line represent walls
  - Red line represent stairs
  - Blue point indicates the starting point of level
  - Green point/s indicate the floor(1 dot for ground, 2 dots for first floor,& so on)

2.3.2 Sample Path



2.3.3Steps in top view image processing

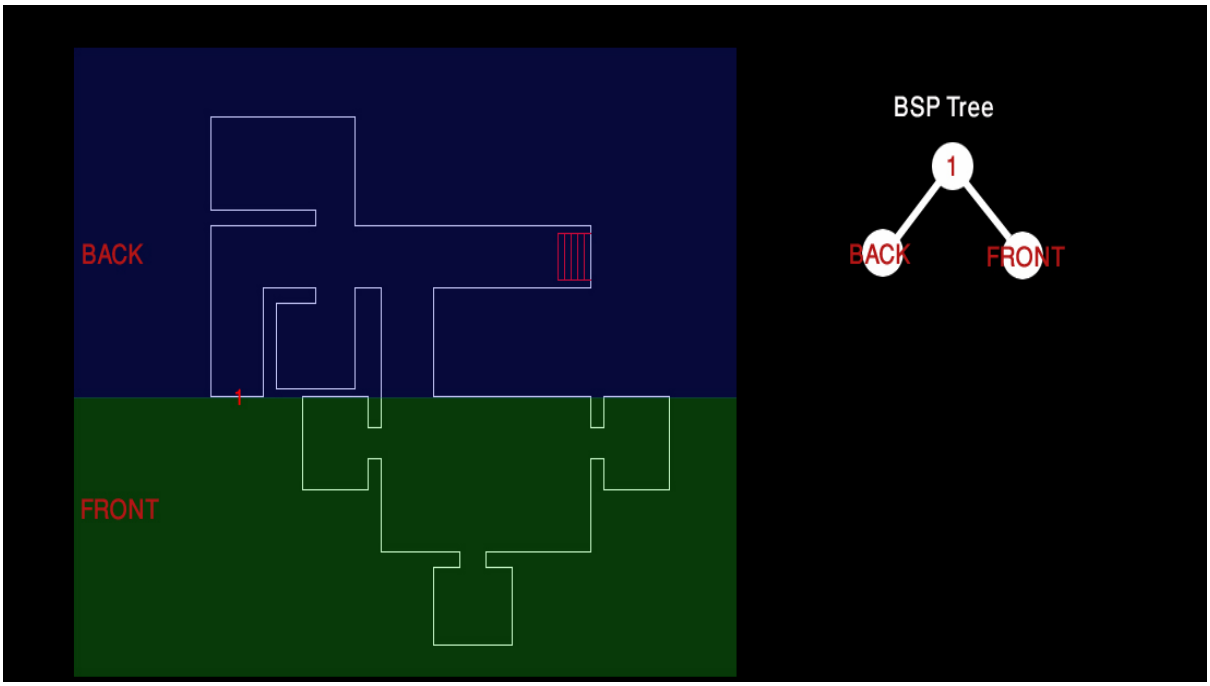
- Find 2D edge points
- Derive normal vectors
- Find 3D coordinates of planes that shall represent wall
- Normalise 3D plane coordinates
- Scale plane coordinates
- Export plane coordinates to text file

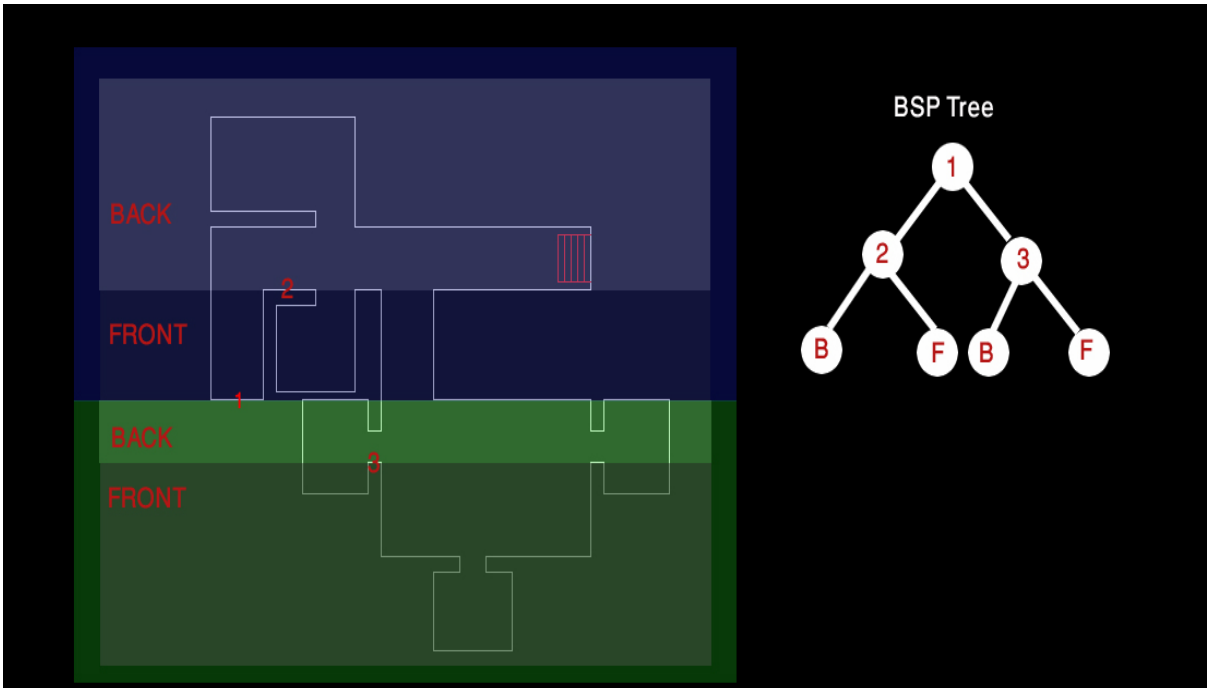
2.3.4 Introduction to BSP tree

A BSP tree is a binary tree whose root node is selected such that the line entered in root node will divide the entire design in almost two equal halves.

All the line lying on the left side or up side of partitioning plane are pointed by the left pointer and all the lines lying the right side or bottom side are pointed by the right pointer.

2.3.5Sample BSP Tree Creation





2.4 CHARACTERS

The frames for all 3D characters are edited in 3DsMAX. Then, individual frames are exported in 3ds file format

The frames for first person include single frame consisting of hand & rifle for first person to represent idle state and several frames to represent the animation while firing

The frames for enemy consists of several frames to represent the animation for different states Stand, Die, Chase & Fire.

2.5 CULLING

A path can consists of thousands of polygons. Rendering becomes slow if all such polygons are rendered

So, some sort algorithm is needed to eliminate unwanted set of polygons.

Culling Algorithm

1.Get the position of first person

2.Traverse along the BSP tree to find the current leaf containing the first person

3.Find other leaf that are potentially visible from the current leaf using PVS(Potentially Visible Set) adjacency matrix

4.Render only those leaves found in above two steps.

2.6 COLLISION DETECTION

Testing for collision with all the polygons of path can be inefficient and in fact not necessary. So, an algorithm is needed to reduce the number of polygons that needs testing

Algorithm

1.Get the position of first person

2.Traverse along the BSP tree to find the current leaf containing the first person

3.Test for collision with only those polygons that form the current leaf.

2.7 STATE DIAGRAM

2.7.1 Definition- State machine is a system that has finite number of states and predefined procedures for changing the states.

2.7.2Set of possible states for an enemy

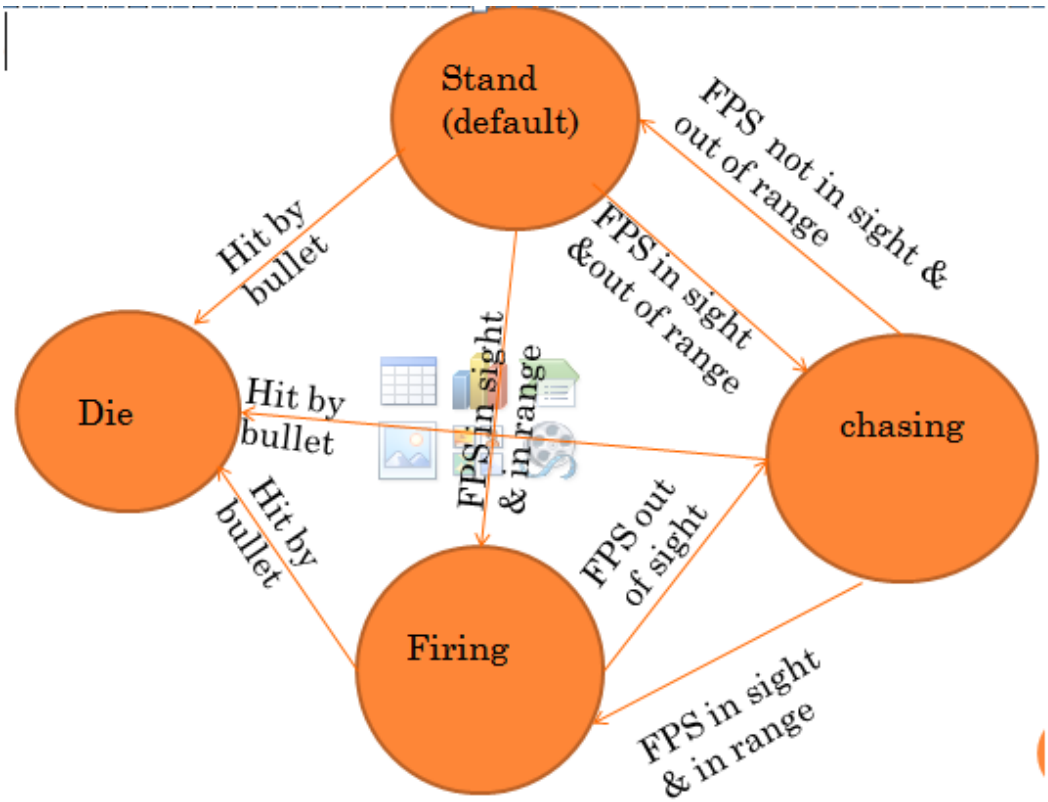
Stand

Chase

Fire

Die

2.7.3 State transition diagram



3. Conclusion:

We could develop a program that allows to navigate in the game field. The first person can fire aiming at the enemy and there are multiple floor level within game level.

4. References:

1. Binary Space Partitioning Trees and Polygon Removal in Real Time 3D Rendering- a master thesis by Samuel Ranta-Eskola

2.Tricks of the 3D Game Programming Gurus - Andre