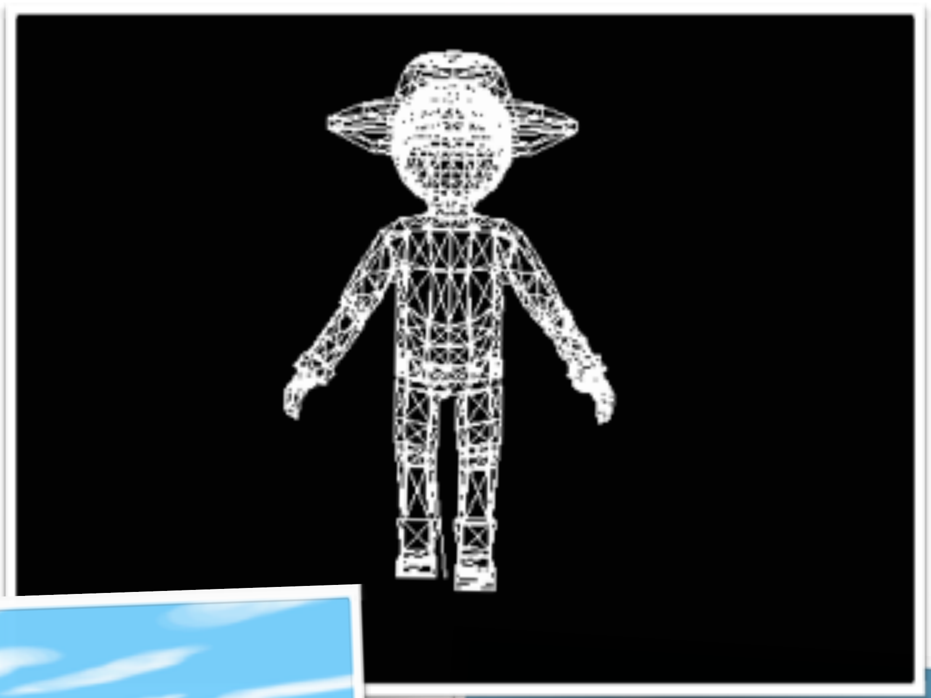
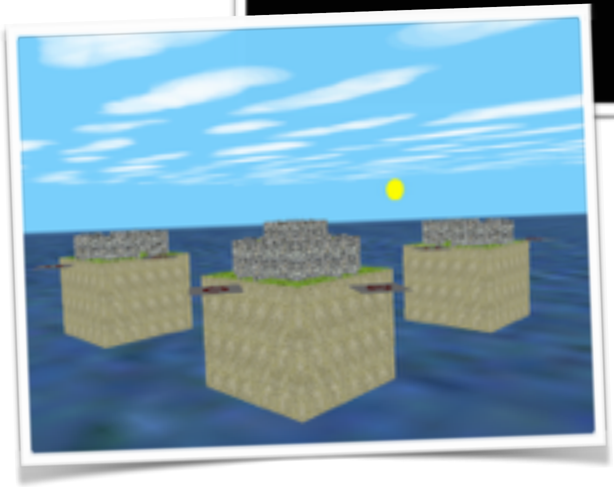
vamshi kolanu
May 21, 2016

Exploring 3-D world





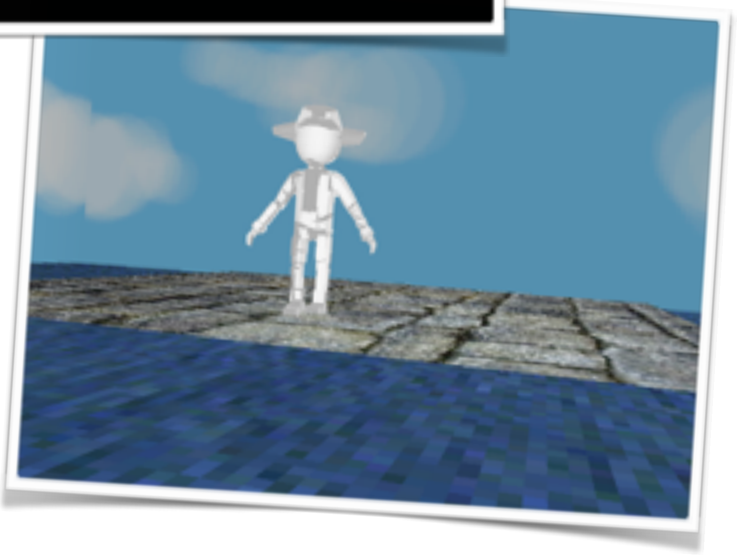


Table of contents:

1. Introduction of the topic.
2. Creating the character using blender and its movement.
3. Creating the 3D world.
4. Viewing the 3D world.
5. Algorithms used for movement of the character in 3D world.
6. Complicated sections of the project.
7. ReadMe
8. References.

Introduction

This project is 3D single player game which has been designed in openGL. In this game the user has been provided the privilege to explore a 3D world. Initially the user is provided with 3 characters Dave, Staurt and Steve among which he has to choose one. This character will help the user to explore the mysterious 3D world.

The 3D world has many obstacles and randomly moving objects which the user has to cross to explore different sections of the 3-D world. Different views will help the user to easily navigate from one part of the world to another. The character also has to privilege to view in all the possible directions i.e. a 360 degree view.

The game ends when the user falls into the river while trying to explore the world. I hope you will have fun playing the game :) !

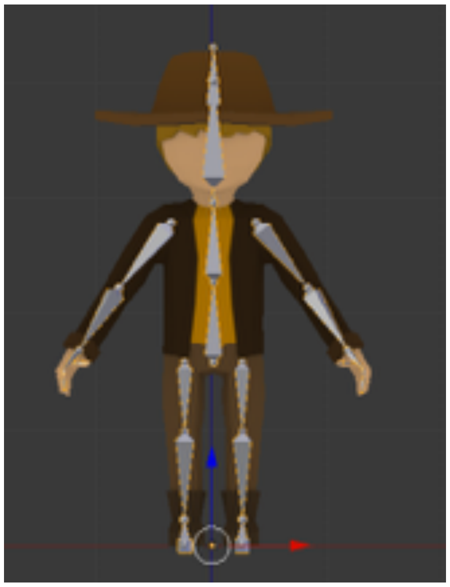
Creating the Character.

Character has been created using a .blend object. Blend file has been converted to a object file which contains the vertex count, face count and vertexes of the character. There were 2 challenges while plotting the blend object.

1. Movement of different body parts of the character.

2. Co-ordination between different body parts of the character while walking and jumping.

This object file is being used to plot the character in openGL. This can't be implemented directly because plotting the blend object as a whole makes it impossible to move various parts of the character separately. So, I have used the blender tool to divide the blend object into different parts which are as follows: left_upper_arm, right_upper_arm, head, body etc. These parts of the character were imported separately and then plotted on the screen using openGL functions.

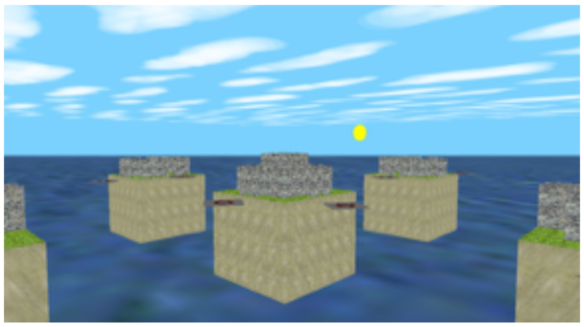


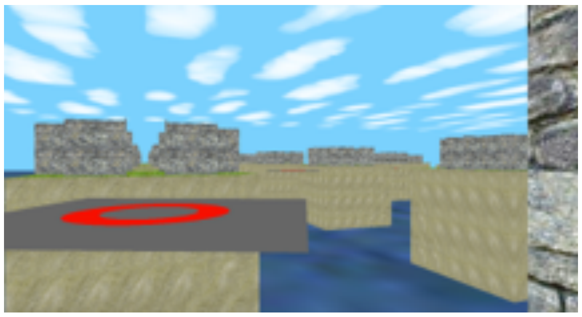
Now, there has to be synchronization between different parts of the character while he is walking in the 3-D world. To solve this issue physics equations were used to calculate the angle by which his hands and legs should move when the character travels unit distance. Accordingly, these values were substituted in the openGL equations to render the character on the screen.

Creating the 3D World.

The complete game lies in the concept of exploring the 3-D world. The character start from a initial position and the game ends when the character falls into the river or reaches the final point.

The 3D-world contains of cliffs, sun, a river and moving horizontal tiles. The character is situated on one of the cliffs at the beginning of the game. There are many cliffs in the game and the character has to move from one cliff to another cliff to reach the end point. The horizontal titles randomly move from one cliff to another cliff which help the user to travel from one cliff to another.





The character has to jump on the horizontal titles to move between cliffs. The game ends when the character falls from the cliff into the river while moving from one cliff to another cliff.

The sun is a point source light object which illuminates the complete world with diffuse and specular lighting. Texture mapping has been used to map grass, walls, cliff, rivers, sky, sun etc elements of the object. The texture mapping has been done by loading the bmp image on the quad objects which are being drawn to create the world.

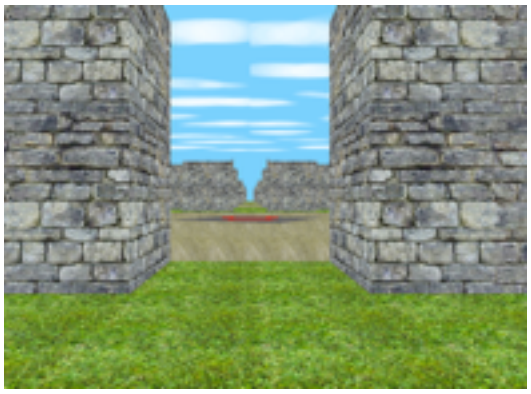
Viewing the 3D world.

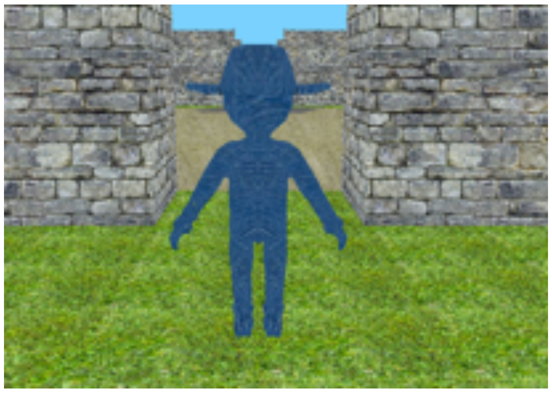
This 3D world can be viewed through various cameras. There are many viewing options to help users navigate through this 3-D adventure world. Various 3D viewing positions which are provided to the user are:

1. First person view
2. Third person view
3. Tower view
4. Helicopter view

All these views are implemented by including a proper lookout vectors in the game.

Apart from the above mentioned views, there are also lookup and look-down functions in the first person view which can be used to view the entire world.





First person view ***Second person view***

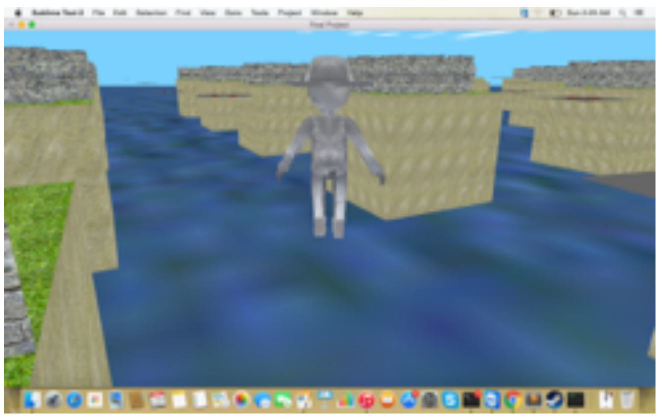
Algorithms used for movement of the character in 3D world.

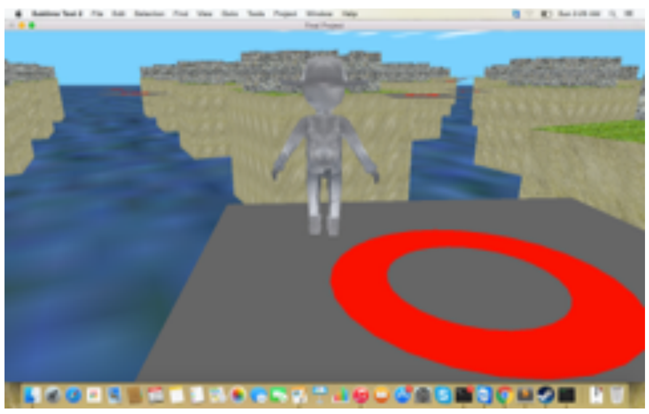
The character should detect collision with the walls of the 3D world. This project uses collision detection algorithm to keep check whether the user passes through the walls or not.

For this part of collision detection, it is assumed that the character is inside a box and we calculate the distance between the box and wall. If the distance between the wall and character is less than zero, which implies that wall and character are colliding with each other. Therefore, we shouldn't allow character to move forward.

Apart from collision detection basic physics concepts have been applied for the movement of the character. Hands and legs of the human dynamically update depending upon the values which are computed using simple physics equations.

The character can jump and translate from the cliff to horizontal tiles. Detecting whether the character lands on the horizontal tile or not is also done using collision detection algorithm in vertical direction. Following Images show the one in which character is standing on a tile and falling from the cliff .





Complicated sections of the project.

1. Synchronizing blender object during the movement of the object

2. Detecting collision of the character with different parts of the world. (Collision detection)

3. Creating a 3D world and movement of horizontal tiles in the 3D world.

4. Jumping of the robot and detect whether he is landing on the horizontal moving tile or not.

5. Constantly updating the 3D world i.e moving horizontal tiles .

READ ME

Initially select one of the character for playing the game.

Character selection can be done using ‘1’ or ‘2’ or ‘3’ keys corresponding to the character position.

Start the game :) !

1. Move the object using UP, DOWN keys.

2. Rotate the object using LEFT, RIGHT keys.

3. Character can jump using ‘j’ or ‘J’ key.

4. Character can Look up and Look down using ’w’ and ’s’ keys. But this feature is only enabled in first person view mode.

5. Change the view modes by clicking on ‘v’ or ‘V’ keys. Initial view that is displayed in the second person view. Second view is the tower view. Third one is the helicopter view followed by first person view

6. In helicopter view camera can move forward and backward using ‘z’ and ‘x’ keys

7. The game ends as soon as the character falls in the river.

References

1. http://ksolek.fm.interiowo.pl/Blender/
2. https://www.youtube.com/watch?v=DiIoWrOlIRw
3. http://www.videotutorialsrock.com/
4. http://nehe.gamedev.net/tutorial/collision_detection/17005/