Ramrao Adik Institute of Technology

(Department of Computer Engineering)

Mini Project Report

On

Planet Collision Animation

Subject-: Computer Graphics Laboratory

Presented By

Roll No	Batch-Sr. No	Name
18CE2011	A3-4	Sushil Anilkumar Dubey
18CE2007	A3-3	Prithvi Dambal
18CE2019	A3-6	Parth Gujar
18CE2030	A3-9	Samiksha Varpe

Signature of Internal Examiner

Chapter 1: Introduction

Blender is a professional, free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games.

Blender's features include 3D modelling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, motion graphics, video editing and compositing. It also features an integrated game engine.

Blender supports the entirety of the 3D pipeline—modelling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Advanced users employ Blender's API for Python scripting to customize the application and write specialized tools; often these are included in Blender's future releases. Blender is well suited to individuals and small studios who benefit from its unified pipeline and responsive development process

Blender is cross-platform and runs equally well on Linux, Windows, and Macintosh computers. Its interface uses OpenGL to provide a consistent experience.

Blender can be used for the following tasks:

• 3d modelling using polygons

• 3d animation

• video editing (basic)

• compositing (video and stills)

• particles

• liquid simulation

• smoke simulation

• fire simulation

Features of Blender Software: 7.89

• Support for a variety of geometric primitives, including polygon meshes, fast sub divisional surface modelling, Bezier curves, NURBS surfaces, metaballs, icospheres, multi-res digital sculpting (including

dynamic topology, maps baking, re-meshing, re-symmetrize, decimation), outline font, and a new n-gon modelling system called B- mesh.

• Internal render engine with scanline rendering, indirect lighting, and ambient occlusion that can export in a wide variety of formats.

• A path tracer render engine called Cycles, which can take advantage of the GPU for rendering. Cycles supports the Open Shading Language since Blender 2.65.[37]

• Integration with a number of external render engines through plugins.

• Keyframed animation tools including inverse kinematics, armature (skeletal), hook, curve and lattice-based deformations, shape animations, non-linear animation, constraints, and vertex weighting.

• Simulation tools for soft body dynamics including mesh collision detection, LBM fluid dynamics, smoke simulation, Bullet rigid, ocean generator with waves.

• A particle system that includes support for particle-based hair.

• Modifiers to apply non-destructive effects.

• Python scripting for tool creation and prototyping, game logic, importing/exporting from other formats, task automation and custom tools.

• Basic non-linear video/audio editing.

• The Blender Game Engine, a sub-project, offers interactivity features such as collision detection, dynamics engine, and programmable logic. It also allows the creation of stand- alone, real-time applications ranging from architectural visualization to video games.

• A fully integrated node-based compositor within the rendering pipeline accelerated with OpenCL.

• Procedural and node-based textures, as well as texture painting, projective painting, vertex painting, weight painting and dynamic painting.

• Real-time control during physics simulation and rendering.

Chapter-2 Implementation

Technologies used:

Software Specification: Blender 2.79b

Step 1: Open blender software.

Step 2: Add 2 UV spheres (1 big and 1 small) assigning keyframes to locations at frame no. 1 and 150 respectively.

Step 3: Add a torus at angle 90° to the sphere and place it in between the spheres where the spheres intersect i.e. at frame no. 50.

Step 4: Go back to solid view, turn blender render to cycles render and make surfaces of the spheres smooth.

Step 5: Change the rotation coordinates of the lighting source. Now, change the view to object mode, change the strength and colour of both the spheres. Save the work you have done till now.

Step 6: Now, go to the world column and add a background for the planets. Go to the screen layout menu and choose compositing.

Step 7: For the small planet, add shaders as well as converters to fix the colour of the planet before, during and after collision effects.

Step 8: Add another material from the material section and name it large. The functions of the large planet will be similar to that of the small planet. You just need to change the colour and colour strength of the large planet for before, during and after collision effects.

Step 9: Save your work. Go back to the default screen and apply soft body to the large planet and collision to the small planet. Due to this, a band will be seen at the intersection of the two planets which shows that the planets are burning due to the collision.

Step 10: Once you are done here, go to the compositing screen and limit orange colour to the intersection part. Now, to add some particles that fly off when collision takes place, go to default screen, add 2 icospheres and add colour to them.

Step 11: Add particles from the particle section for both the planets as well as the torus. At last, change the view to rendered and rotate the planets at -45° and adjust the focal length.

Step 12: Finally, render your animation completely using the dimension you wish.

Chapter-3 Screenshots

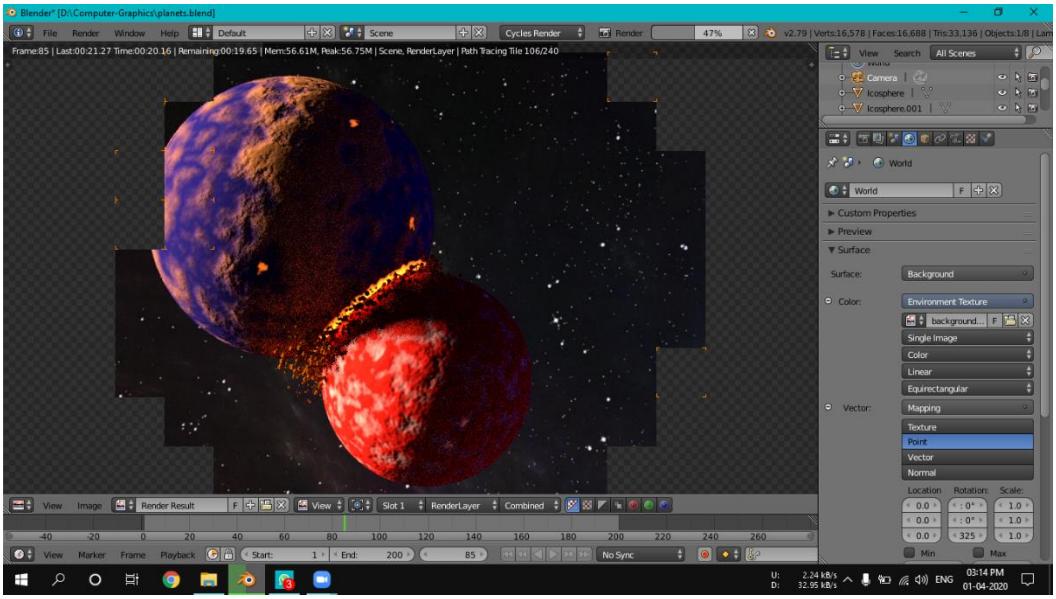

Figure :- 1 Rendering

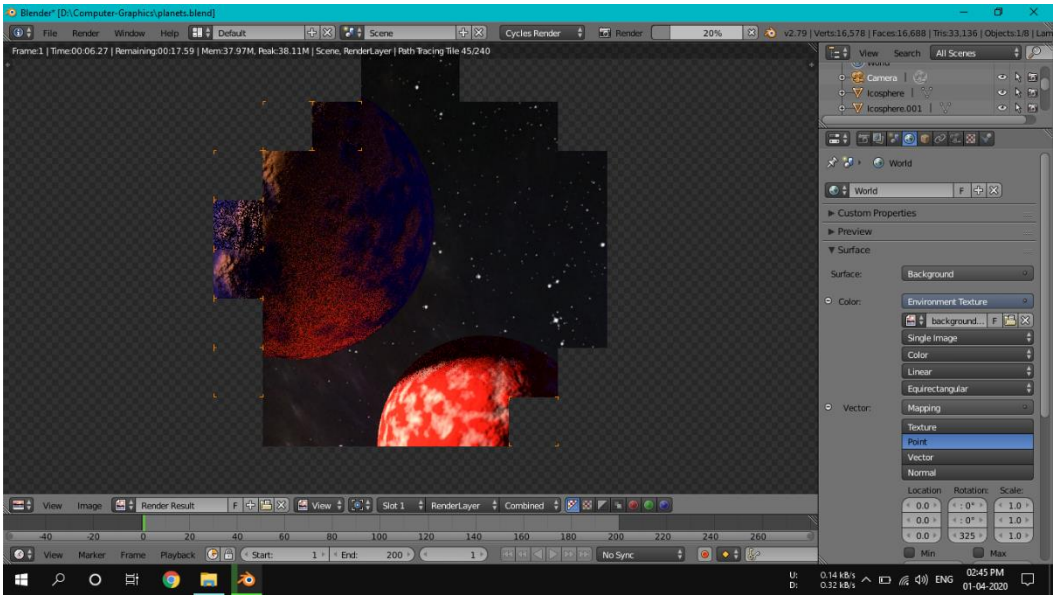

Figure :- 2 Rendering

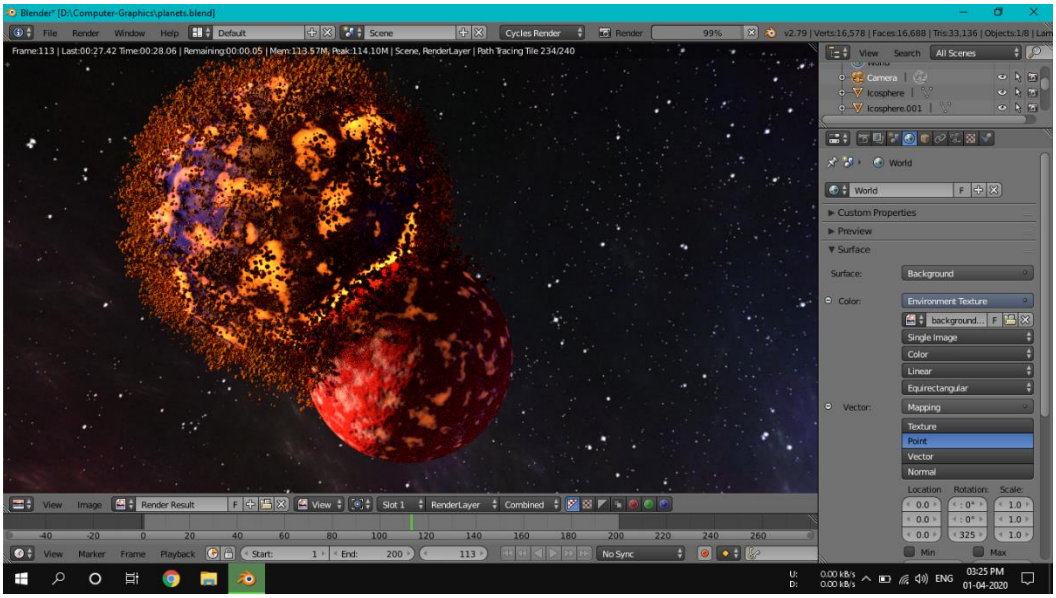

Figure :- 3 Rendering

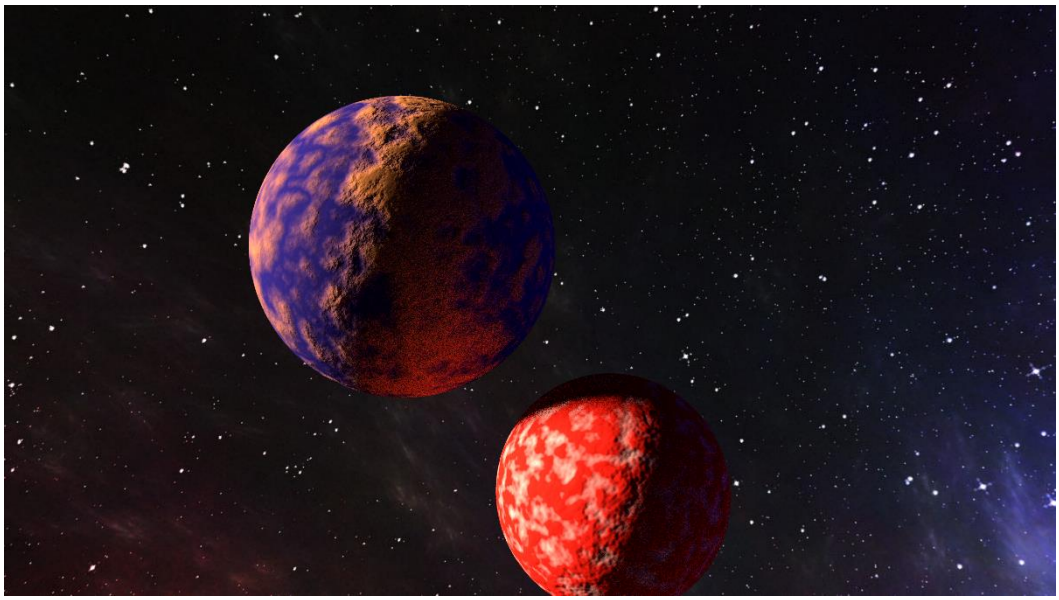

Figure :- 4 Animation GUI

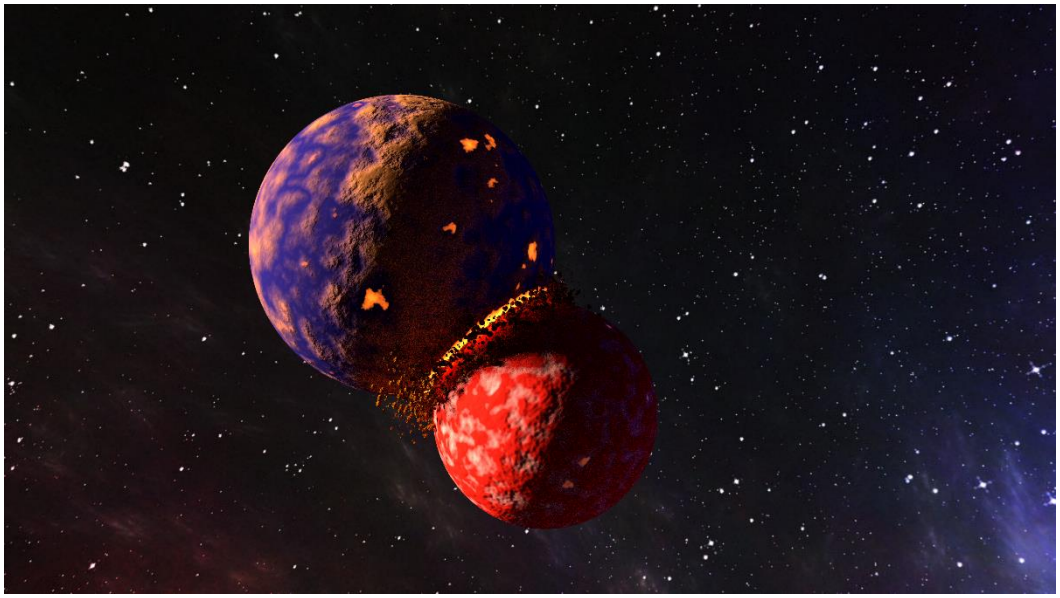

Figure :- 5 Animation GUI

Figure :- 6 Animation GUI

Chapter-4 Conclusion

In this project we are able to create a 3d animation such as collision of two extra-terrestrial bodies by using blender software. By using blender software, we are able to create animation. We can use more tools and different menus in blender to create an effective model.

First, we can create a planetary body, apply different tools to create it as a planet giving texture designs. Then duplicate the same planet in smaller size two show to different bodies. On the bottom we can create one panel which is views as a surface. Then two planets are collided .to set we apply camera view to it. Then rendered the whole view. After going to camera view, we save the frames of the animation.