

Survey on Modern Graphics Programming and Development Tools

Mr. Tushar Ghude.

Dept. of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India.

Prof. Avinash Shrivasa.

Dept. of Computer Engineering
Vidyalankar Institute of Technology
Mumbai, India.

Abstract—This paper explains Unity3D, which is usually treated as game development software with integrated game engine, and its key features. Paper Also Summarizes three main steps in graphic development: Basics of modeling and animation-Modeling all the geographical entities related to different layers are converted to 3D model by AutoCAD and 3dsMax, Blender software and Rendering which includes lightning simulation, texturing and displaying final image on 2d screen. Normally 3D models are imported into the Unity3D software and scripting is performed using JavaScript language in Visual Programming Language Editor in order to achieve objects and Scenes (virtual reality). To play Unity 3D graphics user need UnityWebPlayer, Unity 3D graphics are a new generation of games, virtual graphic environment with superior 3D graphics. With Active X support users can choose their own way to browse and navigate in the virtual reality.

Keywords— computer graphics, modeling, rendering, rasterization, graphics pipeline, animation, Unity 3D, blender

I. INTRODUCTION

Visual representation of data displayed on a computer screen is called computer graphics. Computer graphics includes images which can be defined as two dimensional representations of three dimensional world, videos (series of images), representation and manipulation of image data by a computer. Various technologies used to create and manipulate images, videos, models the sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content also comes under Computer Graphics.

Computer graphics can be 2D (two-dimensional) or 3D (three-dimensional). Two dimensional computer graphics are usually split into two categories: vector graphics and raster graphics.

Vector graphics are made up of different paths, which are consist of locus of points defined by start point and end point, along with other points, curves, and angles along the way. A path can be a line, a square, a triangle, or a curve. Different shapes and curves are used to create complex diagrams for example engineering drawings created with software like AutoCad. Paths are even used to define the characters of specific typefaces. Many Flash animations also use vector graphics, since they scale better and typically take up less space than bitmap images.

Everything computer's screen, even the text is simply a two dimensional array of pixels. The word "pixel" is derived from

the term "Picture Element". Every pixel on screen has a particular color. A two-dimensional array of pixels [5] is called an image. The purpose of graphics of any kind is therefore to determine what color to put in what pixels. Vector graphics can be scaled without losing quality simply by scaling different objects (lines, curve) by the same size. To scale raster graphics different techniques are available like replication or interpolation in these methods two adjacent pixels along the rows is taken and placed between two pixels and therefore scaled raster image appears patchy or blocky.

II. IDEA BEHIND MODERN 3D GRAPHICS

Since all graphics are just a two-dimensional array of pixels, 3D graphics is thus a system of producing colors for pixels that convince viewer that the scene he or she is looking at is a 3D world rather than a 2D image. The process of converting a 3D world into a 2D image of that world is called rendering [5].

To produce real time 3D graphics rendering algorithm takes an image described in a vector graphics format (shapes) and convert it into a raster image (pixels or dots) for output on a video display or printer, or for storage in a bitmap file format. This process is called rasterization, and a rendering system that uses rasterization is called a rasterizer.

In rasterizers, all objects are made up of empty shells. To create illusion of the third dimension (depth) rasterizers uses a special technique to replace part of the shell with another shell and alter their size that way it creates illusion of depth or how object will look from inside but it simply replaces that part of shell. Also there are techniques to change resolution of grid i.e. one can substitute more triangular shell to represent the area that creates illusion of closer view of the surface and vice versa. Even surfaces that appear to be round are merely triangles if one notices closely enough. An object is made out of a series of adjacent triangles that define the outer surface of the object. This triangular grid defines geometry of object, is often called geometry, a model or a mesh.

The Rasterization process has many phases. Each phase is ordered into a pipeline and rasterization process follows a particular sequence, where 3D object information in a vector format enter from the top and at the end it produces 2D raster image for output on a video display or printer, or for storage in a bitmap file format. This is one of the reasons why rasterization is so amenable to hardware acceleration: it

operates on a single shell at a time and processes information in a particular order i.e. order in which various meshes are submitted will affect final output. Rasterizer only processes next triangular shell only when it finishes with current one that way it covers hidden surfaces correctly while displaying image on the screen though triangles can be fed into the top of the pipeline while triangles that were sent earlier can still be in some phase of rasterization.

III. DESCRIPTION OF GRAPHICS PIPELINE

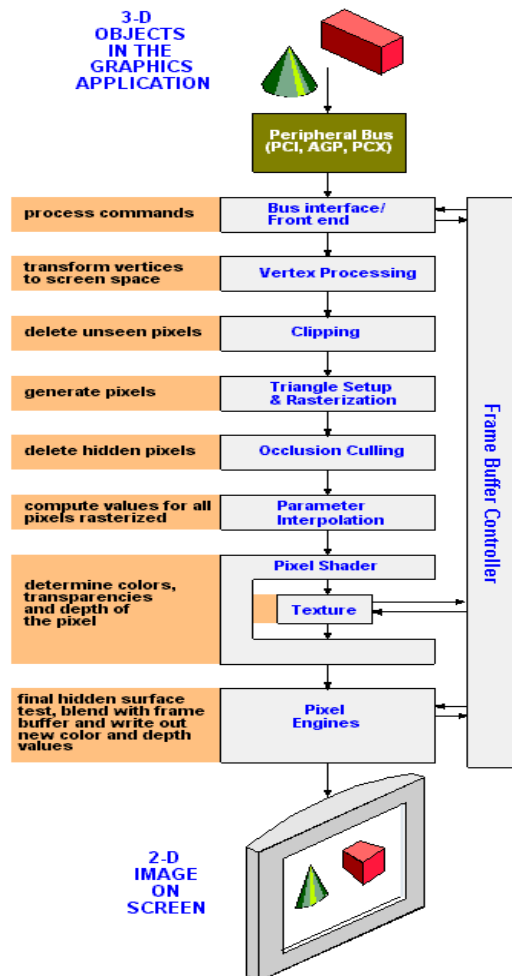


Fig. 1. Graphics pipeline

Graphics pipeline or rendering pipeline refers to the sequence of steps used to create a 2D raster representation of a 3D scene. Every graphics pipeline consists of following main stages:

Stage 1: The first phase of rasterization is Clip Space Transformation. This phase transforms vertex information of each triangular shell into space region. This marks boundary in space, everything falling inside of that boundary will be process and rendered to the output and everything that falls outside will not be process. Object vertices positions in clip space are called clip coordinates. To define a point in 3D space

we need three coordinates X,Y and Z. To define a point in clip space we need four coordinates first three are the same X,Y,Z positions the last coordinate W defines extent of the clip space for this vertex. It marks near and far clipping plane, this clip space can be different for different vertices and since each vertex can have an independent W component, each vertex of a triangle exists in its own clip space. Positive Z axis of clip space goes away from the region and clipping region falls between two cuts on Z axis (Zmin: defines near clipping plane and Zmax: defines far clipping plane), positive X axis to the right, the positive Y axis is up.

Stage 2: Normalized Coordinates: normalized coordinates are obtained by dividing X, Y, Z position of each vertex by W coordinate. Clip space is inconvenient to render on the other hand normalized coordinates can be handled efficiently by transforming clip space into normalized device coordinates. It is essentially just clip space, except that the range of X, Y and Z are [-1, 1]. The directions are all the same.

Stage 3: Window coordinate system in computer graphics is a rectangular area in the users own coordinate system known as the WORLD system. They still represent three dimensional coordinates X, Y, Z but they are bounded by boundaries of viewable window. The bounds for Z coordinate are between 0 (being the closest) and 1 (being the farthest). Vertex positions outside of this range are not visible. Some graphics packages allow programmer to specify output primitive coordinates in a floating-point world-coordinate system representing world-coordinate window and corresponding rectangular region in screen coordinates called the viewport[5][7] into which world coordinate window is mapped.

Stage 4: Scan Conversion: When coordinates of an object are converted into window coordinates, Scan conversion maps object on pixel grid as shown in the following figure. It breaks object according to the window pixels over the output image that the object covers.

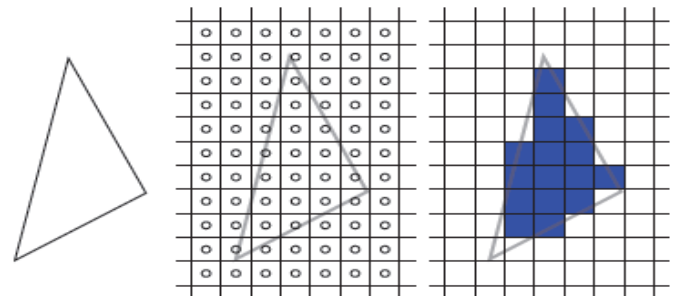


Fig. 2. Scan conversion

The center image shows the digital grid of output pixels; small circle at the center of every pixel represent a sample: a discrete location within the area of a pixel. It is used to determine how much area object covers, fragment is generated if center of the pixel falls under the object boundary i.e. a triangle will produce a fragment for every pixel sample that is within the 2D area of the triangle. This creates a rough approximation of the triangle's general shape as shown in the image on the right.

Stage 5: Fragment Processing: Attributes output from the Vertex Processing stage include: colors resulting from lighting calculations, Texture coordinates and fog coordinates. The

Fragment stage accepts the varying values from the rasterizer. It uses them to determine the final color of the fragment. It takes a fragment from a scan converted triangle and transforms it into one or more color values and a single depth value. Raster Operations merge the generated fragment with the pixel already found in the frame buffer. The order that fragments from a single triangle are processed in is irrelevant because the triangle lies in 2D space its fragment cannot possibly overlap. However, the fragments from another triangle can possibly overlap, Second rendering object can obscure the first even if its farther from the view. That's why the fragments from one triangle must all be processed before the fragments from another triangle.

Stage 6: Shader: A Shader[2][5] is a program design to run as a part of rendering operation, it executed at certain points. Shader stages represent hooks where a user can add arbitrary algorithms to create a specific visual effect. Shader defines set of algorithms that determines how 3-D surface properties of objects are rendered, and how light interacts with the object within a 3-D computer program. Newer GPUs (Graphics Processing Unit) calculate shaders where previously it was an algorithm calculated by the CPU.

IV. GRAPHICS PROGRAMMING PROCESS

Graphics programming process consists of three main steps. They are modeling, animation and rendering. In 3D computer graphics, 3D modeling [3] is the process of developing a mathematical representation of any three-dimensional surface of object via specialized software. Three dimensional model can be represented as a two dimensional image on computer screen using process called 3D rendering or it can be used in a computer simulation with physics engine to simulate real world object interaction. The model can also be physically created using 3D printing devices. There are three popular ways to represent a model:

Polygonal modeling - Vertices of an object in 3D space are connected by line segments, it forms polygonal mesh. These polygons together form polygonal mesh approximating surface of the object. This type of modelling is called polygon modeling. Creating 3D models using polygon meshes is relatively easy, also it is easy to handle. Computer can process low polygon model very fast but low polygon models appear blocky as polygons are planar and can only approximate curved surfaces using many polygons. On the other hand as number of polygons increase model becomes more realistic (it closely approximates curved surface) but with increase number of polygon and vertices processing overhead increases.

Curve modeling - Surfaces are defined by curves, which are influenced by weighted control points. The curve follows (but does not necessarily interpolate) the points. Increasing the weight for a point will pull the curve closer to that point. Curve types include non uniform rational B-spline (NURBS), splines, patches and geometric primitives.

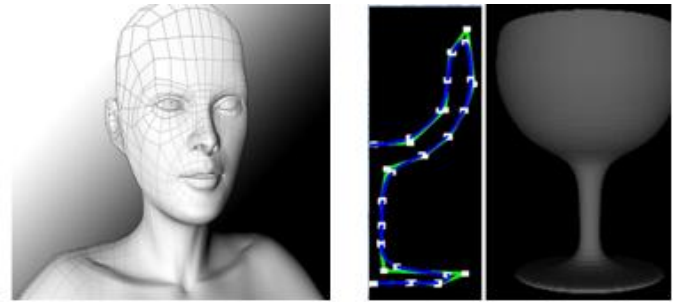


Fig. 3. Polygon modeling and curve modeling

Digital sculpting- also known as Sculpt Modeling or 3D Sculpting, is the use of software that offers tools to push, pull, smooth, grab, pinch or otherwise manipulate a digital object as if it were made of a real-life substance such as clay.

Animation: Computer animation is essentially a digital successor to the stop motion techniques used in traditional animation with 3D models and frame-by-frame animation of 2D illustrations. To create the illusion of movement, an image is displayed on the computer monitor and repeatedly replaced by a new image that is similar to it, but advanced slightly in time (usually at a rate of 24 or 30 frames/second). This is achieved using double buffering

The trick behind double buffering is that program keeps a copy of the screen in main memory. When it is done updating this buffer, it copies the whole thing to the video buffer and segment. This has a number of advantages: No flickering, Main memory is much faster than Video memory, the user cannot see the frame being update, even at low FPS (frames per second). Of course, the main disadvantage of it is that it costs an extra memory that might be needed elsewhere.

For 3D animations, all frames must be rendered after modeling is complete. For 2D vector animations, the rendering process is the key frame illustration process, while twined frames are rendered as needed. For pre-recorded presentations, the rendered frames are transferred to a different format or medium such as film or digital video [10]. The frames may also be rendered in real time as they are presented to the end-user audience. Low bandwidth animations transmitted via the internet (2D Flash, X3D) often use software on the end-users computer to render in real time as an alternative to streaming or pre-loaded high bandwidth animations.

V. DEVELOPMENT TOOLS- UNITY 3D ENGINE AND BLENDER

Unity is a cross-platform graphic engine with a built-in IDE developed by Unity Technologies. It is used to develop video games for web plugins, desktop platforms, consoles and mobile devices. It grew from an OS X supported game development tool in 2005 to a multi-platform graphics engine [4][6]. The latest update, Unity 4.2.2, was released October 2013. It currently supports development for iOS, Android, Windows, BlackBerry 10, OS X, Linux, web browsers, Flash, PlayStation 3, Xbox 360, Windows Phone 8, and Wii U. Two versions of the engine are available, Unity and Unity Pro.

Unity3D Engine Internal Analysis

The component model is applied in the Unity3D graphic development, which provides a scalable programming architecture. Software function modules can be reused conveniently in this component model. Each entity in a scene is called as an Object, which has the characteristics of a container.

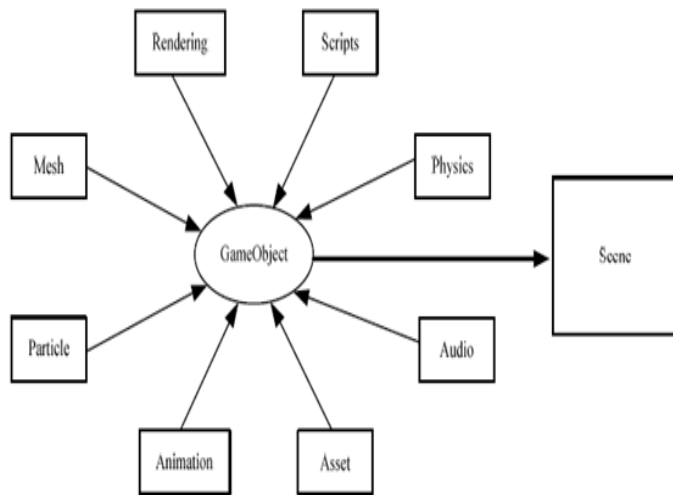


Fig. 4. Unity component model

According to the needs of software many different components are added into an Object. A component can be taken as a collection with a group of related functions, which can be accessed through an interface. For example, a script is a component and its role is to offer a logical operation on an object, and the Box Collider component [1] in Unity3D. Collider mesh specifically provides a support for objects collision detection. Unity3D has many predefined components. Programmers can combine some of them to create a feature rich Object.

Any graphic developed is composed of one or more scenes, each scene includes one or more Objects, and moreover, every Object is composed of some components or child Objects. In development, besides directly using Objects predefined in Unity3D, programmers can create an empty Object with the information about position, rotation and scale of an object, and then add scripts or other components into it. Prefab in unity 3D is a technology like a template, in order to provide the same type of object management. A prefab contains both objects and resources. If the same type of objects needs to be created, a Prefab can be used in this situation. All Objects will be updated simultaneously when its Prefab is changed [1].

The above mechanism of Prefab can greatly improve the maintenance efficiency of graphic simulation or gaming software. To observe the impact of an object's state, programmers can dynamically change the configuration parameters of a component when the application is running. After the Software exits, all its initial parameters will be reset. In order to better grasp the impact of elements, details of differences between design and actual running affect can be discovered by using the game view and scene view at the same time.

Blender

Blender [2] is a free and open-source 3D computer graphics software product used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, UV unwrapping, texturing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting [7][9], Photorealistic Rendering, realistic materials, animating, match moving, camera tracking, video editing and compositing. It also features a built-in graphic engine.

Photorealistic Rendering- Blender features a powerful new unbiased rendering engine called Cycles that offers stunning ultra-realistic rendering. The built-in Cycles rendering engine offers: GPU & CPU rendering, real time viewport preview, HDR lighting support, Permissive License for linking with external software

Fast UV Unwrapping- Blender can easily unwrap mesh data, and use image textures or paint directly onto the model. Blender allows for: Fast Cube, Cylinder, Sphere and Camera projections, Conformal and Angle Based unwrapping (with edge seams and vertex pinning), Painting directly onto the mesh, Multiple UV layers UV layout image exporting. Models can be exported to another applications like Unity 3D.

Camera and Object tracking- Blender includes production ready camera and object tracking. Allowing user to import raw footage, track the footage, mask areas and see the camera movements live in 3d scene.

These are few images of the modeling project that has been completed under this survey.

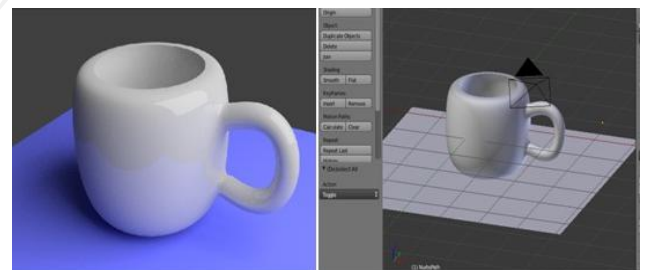


Fig. 5. lighting and shadow effect

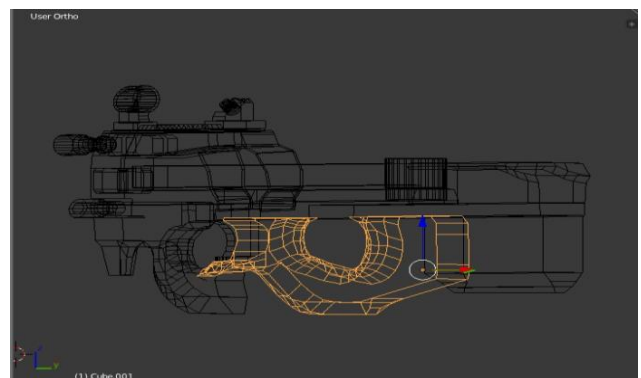


Fig. 6. Modeling project- wireframe view

REFERENCES

- [1] Research on key technologies base Unity3D game engine Jingming Xie Inf. Eng. Inst, Guangzhou Panyu Polytech. Coll, Guangzhou, China. 17 July 2012 Computer Science & Education (ICCSE), 2012 7th International Conference.
- [2] www.blender.org
- [3] Immersive 3D modeling with Blender Takala, Tuukka M. Aalto University, Department of Media Technology, Finland Makarainen, Meeri; Hamalainen, Perttu 17 March 2013 3D User Interfaces (3DUI), 2013 IEEE Symposium.
- [4] <http://www.unity3dstudent.com/>
- [5] <http://www.glprogramming.com/>
- [6] <http://walkerboystudio.com>
- [7] The method of three-dimensional scene modeling and implementation using VC++ and OpenGL Liu Lixin School of Optoelectronics, Changchun University of Science and Technology
- [8] A New Method of Virtual Reality Based on Unity3D Sa Wang, Zhengli Mao, Changhai Zeng, Huili Gong¹, Shanshan Li, Beibei Chen. The Key Lab of Resource Environment and GIS, Capital Normal University, Beijing, China.
- [9] An Augmented Reality Environment for Learning OpenGL Programming 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic
- [10] <http://www.princeton.edu> (princeton university computer graphics and animation)
- [11] Learning Modern 3D Graphics Jason L. McKesson Programming computer science education <http://www.arcsynthesis.org/>

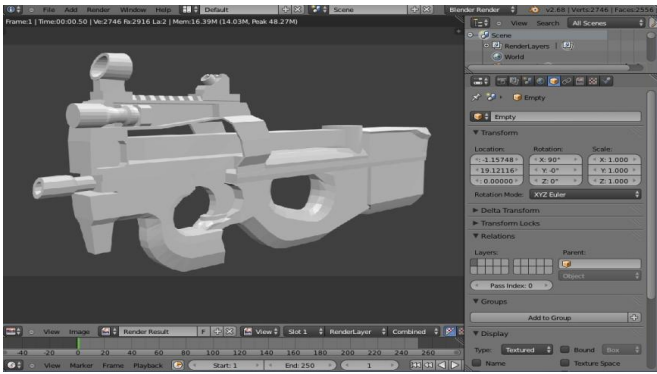


Fig. 7. Modeling project- Final 3D model

VI. CONCLUSION

The rendering pipeline involves a large number of steps, dealing with a variety of math operations. It has stages that run actual programs to compute results for the next. Modern technology supports solid 3D graphics development with modeling tools and graphic engine used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications graphics programming is all about modeling real world objects into basic polygons and then scripting behavior for every object you create. Development tools such as unity 3D can be use for creating 3D environment, scenes as well as to control spontaneous exploration and observation of it and with this new technology users can choose their own way to browse and navigate in the virtual reality.

IJERT