

A Journey of Learning Computer Graphics

Muhammad Diaztando Haryaputra

Informatics/Computer Science
Institut Teknologi Bandung
Bandung, Indonesia
13514002@std.stei.itb.ac.id

Abstract—The author learns computer graphics as it is a close interest to the main goal; video game development. Using the basic programming language of C with the ability of manipulating the very own frame buffer and using libraries of OpenGL, the author draws the world.

Keywords—algorithm; colors; dots; graphics; lines; model; plane.

I. BACKGROUND

As a computation rookie-expert, it is wise for the author to acknowledge the importance of constant learning. The information technology is a ever changing world where, if one does not catch up, the particularly unlucky person will be dusted off and left as a historical monument, and a human cannot be like a monument, as we live in every moment, moving in from one to another, adapts to each. The author here learned a new experience, working towards the main interest of developing video games, programming a visual candy, a break from command line texts and user interfaces, as the author draws every dot, line, and color by writing code.

II. PREFACE

Computer graphics is a vast area in computer science, mainly covering what is known as computer-generated imagery, drawing dots and lines, using shaders, color filling, ray tracing, and other techniques to make complex imagery with a lot of potential uses. Mainstream medias like films and video games are often has components with computer graphics, if not entirely built with it. Highly complex computer simulations have, like the name, highly complex graphical structures combined with math and high computing power to generate an accurate approximate.

The first time computer graphics emerged was in military, initially as a flight simulator, and became a defense system showing planes flying over an area. It used a modified oscilloscope to display the result [1][2][3]. Then a game show up, a tennis simulator, also was implemented in an oscilloscope. This game is the first interactive video game, featuring interactive graphics [4].

As technology evolved, so does computer graphics. New techniques, problems, and solutions were discovered as time

pass by. For example, rendering has the visibility problems with solutions coming up one by one. First was hidden-line algorithm by Roberts (1963) and Appel (1967), then there was hidden-surface algorithm by Warnock (1969) and Watkins (1970), then finally render sorting by Sutherland (1974) [1].

Today computer graphics can be as advanced as films being made entirely in computer imagery and be as realistic as the budget, while classical imagery like pixel sprites being kept in video games and short videos for entertainment purposes.

III. CONTENT

Next are the topics the author has learned in computer graphics.

A. Frame buffer

Frame buffer is the memory where every pixel information on the monitor is displayed. Manipulating the data inside the frame buffer will change what is displayed exactly on the monitor. There are several ways in accessing the frame buffer. In Linux, as the author used, for example, there is a frame buffer device which can be manipulated directly with a code. The code is then executed in a *tty* terminal, and the display will change to what is written in the code. The language author used is C, from the beginning towards the end, since it's very flexible.

The author mainly uses this initial phase to draw simple dots and lines, then draw basic shapes like triangles, polygons, and circles by connecting the dots together. For circles, there are algorithms to make one efficiently, and the author used *Bresenham's* circle algorithm. It's mainly about looping draw a dot and changing the coordinates mathematically so the dots, which is then connected, form a circle [5].

B. Coloring

Coloring means giving an area a color. An area is the space surrounded by lines. The author learned what is called flood fill in this regard, filling an area starting with one pixel and checking the neighbor pixels if any of them were part of a line and color those which were not. It's a recursive process which effectively fills an area [6]. The colors could leak out

and filled outside the area intended and might fill the entire screen.

C. Clipping

Graphics clipping is a method to select an area and choose what to do with the image in there. To describe it better, the author made a map out of dots and lines, made a box controllable with keyboard, and shows the overlapping area of both bigger on the side. It's like a magnifier. Inside the "mini"-map, the overlapping area is made to be highlighted.

D. OpenGL

OpenGL is a standard of graphics library. It contains specification of what a graphics library should at least can do. There are many implementations of OpenGL. OpenGL is mainly to support multi-platform and does not get attached to a specific graphics card. The author begins exploring OpenGL starting with OpenGL Utility Toolkit (GLUT), which has basic functionality like drawing dots and lines, just what the author needs to draw a 2D scenery which includes an arcing rainbow inside. The rainbow is made by drawing circles with different radius and different colors, and the circles are drawn without *Bresenham's* algorithm this time, because to color in GLUT means coloring each dots, and gradient is made by drawing different color on the dots, depending on how the color are needed to be drawn. For rainbows, difference of colors are on the center of the circle and the edges, so the author draw a triangle fan which is drawn consecutively with mathematical loop until it formed a circle.

E. 3D Modeling and Textures

This is the not the final step in learning computer graphics but this is the point where the author really want to reach. 3D modeling is not as much different than 2D modeling. The author only needs to declare the dots and connect them to form plane, then make multiple planes, finally attach them together.

Textures are images which can be attached to a 3D models or 2D planes to give them what is shown on the image, like putting a wallpaper or giving a face to a head. The process in attaching a texture to a model is exactly as straightforward as telling which point in the model should have which point in the texture, so it was quite easy.

IV. CONCLUSION

In the end, the author has gained a lot of new knowledge about computer graphics. Not only how to program it, but also how it works. From dots and lines to models and textures, the process is surprisingly not hard to understand for the author. This whole journey of learning computer graphics has opened another eye in the author's mind, towards developing video games.

REFERENCES

- [1] *History of Computer Graphics*. Retrieved May 3, 2017, from <https://graphics.stanford.edu/courses/cs248-02/History-of-graphics/History-of-graphics.pdf>
- [2] *A History of Computer Graphics Modeling*. Retrieved May 2, 2017, from <http://www.digitalschool.ca/a-history-of-computer-graphics-modeling/>
- [3] *A Brief History of Computer Graphics*. Retrieved May 2, 2017, from http://deseng.ryerson.ca/dokuwiki/mec222:brief_history_of_computer_graphics
- [4] *Tennis Anyone?* Retrieved May 3, 2017, from <https://web.archive.org/web/20151225121214/https://videogamehistorian.wordpress.com/2014/01/28/tennis-anyone/>
- [5] *The Bresenham Line and Circle Algorithm*. Retrieved May 2, 2017, from https://www.gamedev.net/resources/_/technical/graphics-programming-and-theory/bresenham-line-and-circle-algorithms-r767
- [6] *Flood Fill*. Retrieved May 4, 2017, from <http://lodev.org/cgtutor/floodfill.html>

ACKNOWLEDGMENT

I hereby declare that the paper I am writing is my own, not an adaptation, or a translation of someone else's paper, and not plagiarism.

Bandung, May 5, 2017



Muhammad Diaztando Haryaputra