

Bresenham's Lines Algorithm Visualization Using Flash

Hadi Sutopo

Abstract—This paper is intended to develop an algorithm visualization, particularly Bresenham's Line Algorithm for a Computer Graphics course. Algorithm visualization technology graphically illustrates how algorithms work. This visualization can be used to explain how a line is produced in a display computer for education. This research consists of 6 steps which are concept, design, obtaining of content material, assembly, testing, and distribution. During the testing step, the application is run and checked to confirm that it performs exactly what the author has intended and the students can learn Bresenham's Line Algorithm by studying the visualization. Flash movie demonstrate how Bresenham's Line Algorithm visualized by inputting data, performing calculation, and then returning the calculated results back to output text field. Subjects of the research were students of Department of Informatics Universitas Persada Indonesia YAI for implementation of the learning. The data were analyzed using the analytic descriptive method and interpreted in a narrative way based on the research findings. The visualization and the interactivity have been well tested by the students, and the results indicate that students increase their ability to program graphic application.

Index Terms—Computer graphics, multimedia, algorithm, Flash movie, ActionScript

I. INTRODUCTION

In computer science particularly computer graphics, Bresenham's Line Algorithm is an algorithm for line producing. This concept is difficult to be understood for students who study computer science, especially creating coding in programming language.

Therefore one of the solutions is to provide the students algorithm visualization. Many of the phenomena treated in engineering are dynamic and/or three dimensional; often these phenomena are very difficult to be represented in the conventional print media, however it is anticipated that visualization has the capacity to do a much better job. Visualization is gaining its popularity to implement in education institutions either in distance learning or in blended learning. The development of algorithm visualization should be able to observe the condition of student concerned, due to the changes in paradigm of learning which is from teacher-centered learning to student-centered learning [1].

Algorithm visualization uses computer graphics to show the actions of an algorithm step by step. In this research, the algorithm visualization is used to help students understand the concept of Bresenham's line algorithm and creating coding to produce a line.

Our goal is to develop algorithm visualization of Bresenham's Lines Algorithm. The visualization shows all pixels that generate a line according to Bresenham's Line Algorithm. It can be easily understood by students how some pixels produce a line.

A. Multimedia

The word multimedia is a combination derived from multiple and media [2]. We define digital multimedia as any combination of text, graphic (still and animated), sound, and motion video delivered to the user by a computer. The computer is an intrinsic part of multimedia. All these elements - text, graphics, sound, and video - are either computer-generated or transmitted through a computer. Multimedia systems are used in education, presentations, information kiosks, and gaming industry. The power of computers allows users to interact with the programs. Since interactivity is such a powerful concept, many experts in the field of multimedia consider interactivity as an integral part of multimedia.

In a multimedia system, if the user has the ability to control the delivered elements and timing, the system is called an interactive system [3]. There are different devices to provide end-user interactivity. Almost all tools today support the use of keyboard and mouse, button, and even touch screens. Buttons are on-screen objects that will produce some response when the end user clicks the mouse or touches them. The pushbutton control in Windows dialog boxes is an example of a button. Authoring of buttons involves defining the button appearance on screen, the location, and the action when clicked. Assembly tools that support buttons will provide features to do all three things.

B. Visualization

Visualization is presenting data or information that used in science, engineering, medical, business, and others. Numerical simulations often produce data files that contain data values. The data are converted to a visual form that is helpful for the user in analyzing his/her problem. Visualization for science, engineering, and medical presents graphical form information presentation. And the term business visualization is used in connection with data sets related to business, management, social, industry, and other nonscientific areas [4].

C. Bresenham's Line Algorithm

A picture can be described in several ways. Assuming we have a raster display, a picture is completely specified by the set of intensities for the pixel positions in the display. The scene is then displayed either by loading the pixel arrays into the frame buffer or by scan converting the basic geometric-structure specifications into pixel patterns.

Generally, graphic programming creates the basic of

geometric structure, referred to as output primitives. Output primitive is displayed. Output primitive is specific coordinate data and other information that is input how the object is to be displayed. Simple output primitives are straight line, and it is the simplest geometric components of pictures. Additional output primitives that can be used to construct a picture include lines, circles and other conic sections, quadric surfaces, line, curves and surfaces, polygon color areas, and character strings.

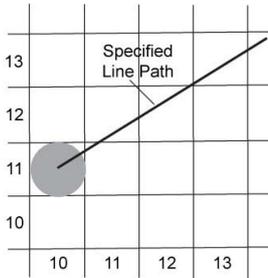


Figure 1. (left) Section of a display screen where a straight line segment is to be plotted

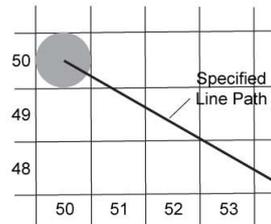


Figure 2. (right) Section of a display screen where a negative slope line segment is to be plotted, starting from the pixel at column 50

Line drawing is accomplished by calculating intermediate positions along the line path between two specified endpoint positions. Some pixels are created in the position between the endpoints [5].

Bresenham's line algorithm is a line algorithm which calculates either X or Y coordinate, and using only incremental integer calculations to produce lines, circles and other curves. Figures 1 and 2 illustrate sections of a display screen where straight line segments are to be drawn. The vertical axes show scan-line positions, and the horizontal axes identify pixel columns. Sampling at unit x intervals in these examples, we need to decide which of two possible pixel positions is closer to the line path at each sample step. Starting from the left endpoint shown in Figure 2, we need to determine at the next sample position whether to plot the pixel at position (11, 11) or the one at (11, 12). Similarly, Figure 3 shows a negative slope line path starting from the left endpoint at pixel position (50, 50). In this one, do we select the next pixel position as (51, 50) or as (51, 49).

II. METHOD

This section presents method of tutorial and testing development that is used in this research. This research uses Multimedia Development Life Cycle (MDLC)[6]. Authoring is somewhat like making a feature film, a movie, and there are many steps to the process. Multimedia Development Life Cycle, a typical multimedia systems development, may involve the following six major steps presented in Figure 1, as follows:

- 1) **Concept.** The objective for the project is defined, and the type of the application is specified. In the movies, this is the stage at which the producer decides the kind of movie to take and the subject to be.
- 2) **Design.** This is the process of deciding in detail what will be in the project and how it will be presented. This stage

includes script writing, storyboarding, making navigation structure and some design steps.

- 3) **Obtaining of content material.** During this stage all the data, audio, video and images for the project are collected in appropriate digital formats. In the course material, this would be the production stage, where all the scenes for the multimedia application are set up with authoring tool Flash.
- 4) **Assembly.** In this step, the overall of the project is built, the tutorial to make puzzle game is assembled, and any interactive features are built. The tool for this stage of authoring is Adobe Flash.

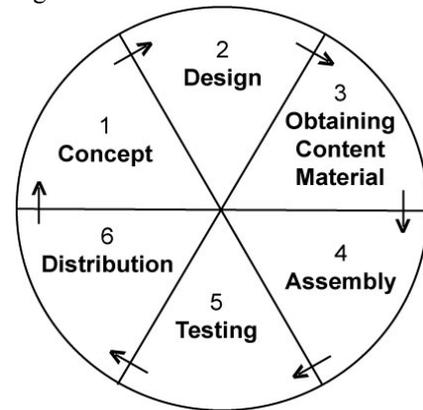


Figure 3. Multimedia development life cycle

- 5) **Testing.** During testing, the application is run and checked to confirm that it performs exactly what the author has intended.
- 6) **Distribution.** In this step, the application is reproduced and delivered to end users for their use. The distribution can be from either C-ROM or web site.

III. DISCUSSION

A. Multimedia Development

To illustrate how to develop multimedia application that contains algorithm visualization in Adobe Flash, the first step is to define the learning objectives [7]. The tutorial involves many topics such as graphic systems, input and output primitives, and graphic programming or scripting.

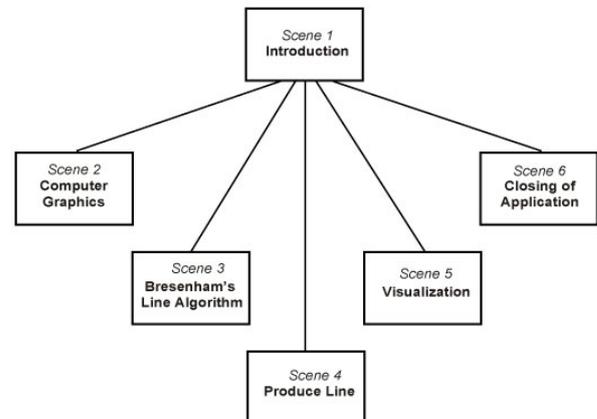


Figure 4. Navigation structure

The second step, developing a navigation structure and storyboard of the Bresenham's line algorithm. First, it

establishes lateral thought processes, helping to break down the navigation structures that are usually embedded in traditional approaches to course delivery. Second, it can result in an overview based on quite abstract design, which in turn generates fresh implementation. Third, it provides a storyboard for identifying relationships between the components. Navigation structure is essential to design an interactive multimedia application shown in Figure 4.

Hearn explained how to make line according Bresenham's Line Algorithm [9] as follows:

- 1) Input the two endpoints and store left endpoint in (x_0, y_0) as the first point, and right endpoint in (x_1, y_1) as the second point.
- 2) Calculate constants dx , dy , $2dx$, and $2dy-2dx$, and gain the starting value for decision parameter as $p_0 = 2dy-2dx$.
- 3) Every x_k along the line at $k=0$, perform the following test: If $p_k < 0$, then the next point to be plotted is (x_k+1, y_k) and $p_{k+1} = p_k + 2dy$. Otherwise, the next point to be plotted is (x_k+1, y_k+1) and $p_{k+1} = p_k + 2dy - 2dx$.
- 4) Repeat step 4 until $x_k = x_1$.

To develop application in Flash, take the following steps:

- 1) Create a new Flash Document.
- 2) Use Text Tool to make four input text fields, and named their variables x_1 , y_1 , x_2 , and y_2 .
- 3) Use Text Tool to make a dynamic text field, and named their variables $output$.
- 4) Create a button to use line algorithm calculating, and performs the pixel coordinates in line producing on the computer display. The button need script as follow:

```

on (release) {
    dx=number((x2)-(x1));
    dy=number((y2)-(y1));
    a=2*dx; b=(2*dy)-(2*dx);
    if (number(x1)>number(x2)){
        x=x2; y=y2;r=x1;
    }
    else{
        x=x1;y=y1;r=x2;
    }
    i=0;
    xArray=new Array();
    while (x<number(r)){
        xa=1.0*x+1;
        x=1.0*x+1;
        xArray[i]=xa;
        i++;
        outputx=xArray;
    }

    j=0;
    p=(2*dy)-dx;
    yArray=new Array();
    while (y<number(y2)){
        if (p<0){
            ya=1.0*y;
            y=1.0*y;
            yArray[j]=ya;
            p=1.0*p+(2*dy);j++;

```

```

        }
    else{
        ya=1.0*y+1;
        y=1.0*y+1;
        yArray[j]=ya;
        p=1.0*p+b;j++;
    }
}

outputy=yArray;
if (number(x1)>number(x2)){
    output="x"+outputx.reverse()+"x2+newline+"y"+y2
}
+
    outputy;
}
else{
    output="x"+x1+"x2+newline+"y"+y1+"y2+
outputy;
}
}
}

```

- 5) To visualize the line producing, create a new button. The button need the following script:

```

on (release) {
    a=x1*5+350;
    b=y1*(-5)+250;
    c=x2*5+350;
    d=y2*(-5)+250;
    _root.createEmptyMovieClip("garis",1);
    _root.garis.lineStyle(2);
    _root.garis.moveTo(a,b);
    _root.garis.lineTo(c,d);
}

```

- 6) Select Control > Test Movie to view the running application.

Flash movie is provided three action button to calculate, to visualize, and to reset as we can see in Figure 5 presents the stage layout in Flash.

B. Testing

The fifth step is Testing, the application is run and checked to confirm that it performs exactly what the author has intended. In the application, this is similar to screening, where the application or parts of it are viewed and approved by the lecturer or his or her assistance. The next Flash movie demonstrate how Bresenham's Line Algorithm visualized. Flash take the input data, perform calculation, and then return the calculated results back to output text field. The figures bellow present a Flash movie with input text fields, output text fields with pixel coordinates calculated by Bresenham's Line Algorithm and line to be produced.

The first testing is producing line in the first quadrant. Figure 6 shows a slope line path starting from the left endpoint at pixel position (10, 5) to the right endpoint at pixel position (20, 20). Similarly, the line is produced by starting from the right endpoint at pixel position (20,20) to the left endpoint at pixel position (10, 5).

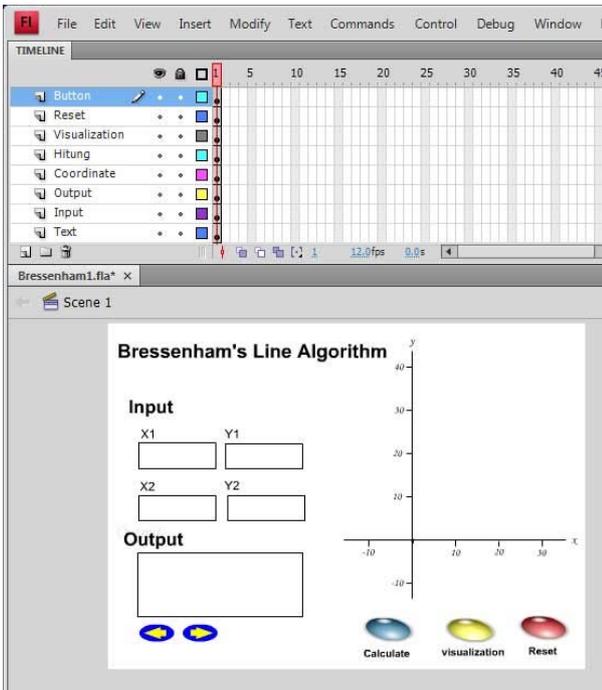


Figure 5. Stage layout in Flash document

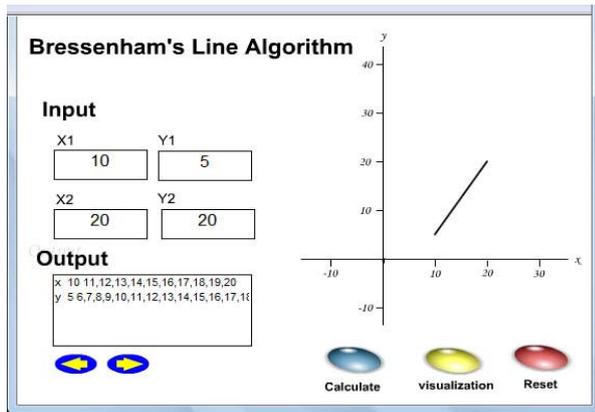


Figure 6. Flash movie visualizes Bresenham's Line Algorithm producing line in the first quadrant

The second testing is producing line in the second quadrant. Figure 7 shows a slope line path starting from the left endpoint at pixel position (10, -5) to the right endpoint at pixel position (20, -10). Similarly, the line is produced by starting from the right endpoint at pixel position (20,-10) to the left endpoint at pixel position (10, -5).

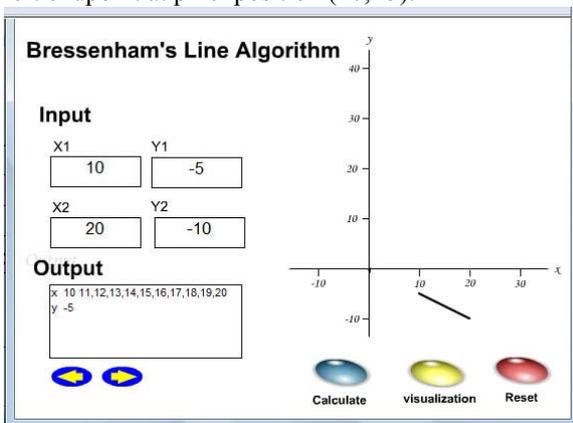


Figure 7. Flash movie visualizes Bresenham's Line Algorithm producing line in the second quadrant

The third testing is producing line in the third quadrant.

Figure 8 shows a slope line path starting from the left endpoint at pixel position (-10, -5) to the right endpoint at pixel position (-5, -10). Similarly, the line is produced by starting from the right endpoint at pixel position (-5,-10) to the left endpoint at pixel position (-10, -5).

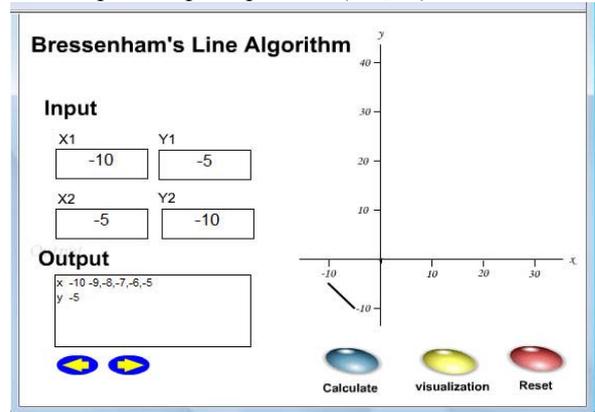


Figure 8. Flash movie visualizes Bresenham's Line Algorithm producing line in the third quadrant

The fourth testing is producing line in the fourth quadrant. Figure 9 shows a slope line path starting from the left endpoint at pixel position (-10, 10) to the right endpoint at pixel position (-5, 20). Similarly, the line is produced by starting from the right endpoint at pixel position (-5, 20) to the left endpoint at pixel position (-10, 10).

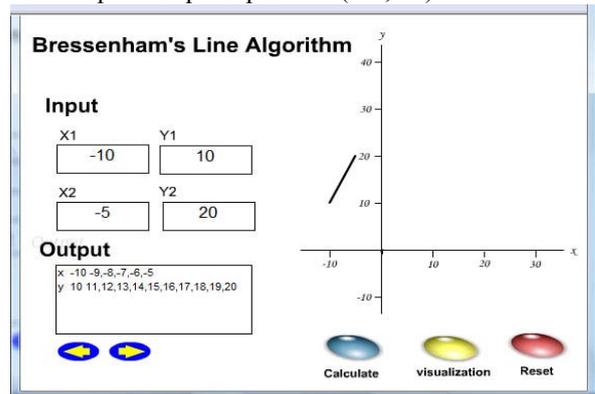


Figure 9. Flash movie visualizes Bresenham's Line Algorithm producing line in the fourth quadrant.

Subjects of the research were students at the Department of Informatics Universitas Persada Indonesia YAI for implementation of the learning. The data were analyzed using the analytic descriptive method and interpreted in a narrative way based on the research findings. The visualization and the interactivity have been well tested by the students, and the results indicate that students increase their ability in creating coding according to Bresenham' line algorithm to produce a line.

IV. CONCLUSION

Through this paper the Bresenham's Line Algorithm visualization has been presented. Some details about graphic programming that should be learned by students have been described. The visualization and the interactivity have been well tested by the students at the Department of Informatics Universitas Persada Indonesia YAI. Adobe Flash is a timeline-based, authoring and object-oriented programming

tools can be used to develop a scientific visualization.

REFERENCES

- [1] Sfenrianto, "A Model of Adaptive E-Learning System Based on Student's Motivation". ICCIT-09 proceedings, pp. 133-138 [International Conference on Creative Communication and Innovative Technology, Indonesia, 2009].
- [2] G. Bhatnager, S. Metha, and S. Mitra, *Introduction to Multimedia Systems*. London: Academic Press, 2001.
- [3] P. K. Anleigh and K. Thakar, *Multimedia Systems Design*. Upper Saddle River: Prentice Hall, 1997.
- [4] Jr. Hill, *Computer Graphics Using Open GL*. Upper Saddle River: Prentice Hall, 1990.
- [5] J. Voley et al, *Computer Graphics Principles and Practices*. New York: Addison Wesley, 1996.
- [6] A. C. Luther, *Authoring Interactive Multimedia*. Boston: AP Professional, 1994.
- [7] A. H. Sutopo, *Multimedia Interaktif dengan Flash*. Yogyakarta: Graha Ilmu Publisher, 2005.
- [8] D. Franklin and J. Makar, *Macromedia Flash MX ActionScripting Advanced Training from the Source*, Berkeley, CA: Macromedia Press, 2002.

- [9] D. Hearn and P. Baker, *Computer Graphics C Version*. 2nd edition. Upper Saddle River, NJ: Prentice Hall International Inc., 1996.



Hadi Sutopo was born in Cilacap, Indonesia on April 15, 1945. He is Doctor of Education in Educational Technology of the Jakarta State University, graduated in November 2009. In 1998 he earned Master of Information Systems at Post Graduate Program Gunadarma University, Jakarta. In 1995 he graduated from the Universitas Persada Indonesia YAI, Jakarta, majoring in Informatics.

Hadi is currently a lecturer of Multimedia at the Universitas Persada Indonesia YAI and some other universities in Jakarta, Indonesia since 1998. He wrote many books on multimedia and information technology. The books are Pemrograman Berorientasi Objek dengan Java (Yogyakarta, Indonesia: Graha Ilmu, 1995), Desain Buku dengan Adobe InDesign Jakarta, Indonesia: Elex Media Komputindo, 2006), and Pemrograman Flash dengan PHP dan MySQL (Yogyakarta, Indonesia: Graha Ilmu, 2007). Current interest research is information technology especially multimedia.

Dr. Hadi Sutopo, MMSI is a member of Indonesian Association of Educational Technology (IPTPI) and Association of Education and Communication Technology (AECT). He works in editorial team of Educational Technology Journal and Multimedia Journal in Jakarta, Indonesia.