International Journal of u- and e- Service, Science and Technology
Vol.8, No. 12 (2015), pp.305-316
http://dx.doi.org/10.14257/ijunesst.2015.8.12.31

Learning Computer Graphics Utilizing Blocks on Web Environment

Hae-Gul Pyun, Haeng-A An and Jinho Park

Global School of Media, Soongsil University, Seoul, 156-743, South Korea
c2alpha@ssu.ac.kr

Abstract

Computer graphics has been applied for generating digital contents in many fields such as virtual/augmented reality, as well as traditional applications such as visual effects. However, serious mathematical background obstructs people from learning computer graphics. The proposed paper focuses on computer graphics for generating an effective e-learning platform. We utilized a block-based system, which is implemented by WebGL and delivers educational interests and enhanced accessibility. Also, users could practice the theory of graphics using processing language and check their programming result directly.

Keywords*: Computer Graphics, E-learning, WebGL, Block Based, Interactive*

1. Introduction

If people could learn one programming language, it would be easier to learn the other programming languages. Similarly, if people figure out the principle of computer graphics, they can understand more easily in overall computer graphics area. These days, the interest in software education is increasing and we tried to look into adolescents' greater access to computer graphics area. Therefore, this paper analyzes the computer graphics theory and proposes an educational software model that utilizes learning through block making using this theory. Through a block composition method, users can learn computer graphics theory easily and model objects by applying this method.

The whole world seems to be in the grip of software education fever. Finland expanded the SW schools in 2004, the UK designated the SW education as the subject of regular course for the first time among G20 nations, and the US formed a computer programming class as a regular course in autumn of last year. Korea enforces SW compulsory education this year and is planning to adopt this education as regular course. Also, the government has announced the plan for the development of SW professional manpower for the realization of SW society with the Ministry of Science, ICT and Future. SW will become a second world common language in modern society that is worked completely based on computers. Therefore, some programming educational software is emerging for people who are not experts, to increase the accessibility of and teach coding easily. That is, the market of programming educational software will be growing rapidly in future.

Computer graphics are taking center stage in many computer areas. To be specific, we know that computer graphics are used in many parts of our life such as science, engineering, medicine, business administration, industry, arts, amusement industry, and advertisement. Recently, its necessity has risen in 3D printing, modeling, Head Mount Display Oculus and augmented reality used in Google glass.

However, in order to learn computer graphics programming, people need mathematical knowledge because it is based on mathematical calculations such as three-dimensional coordinates, matrix, and vector. Aside from that, because of space perception ability, it is hard to studying computer graphics programming even for people who are already proficient in other computer programming courses. Therefore, if we suggest an

ISSN: 2005-4246 IJUNESST

educational environment wherein users can learn computer graphics programming easily, it will be helpful for manpower training in the computer graphics area.

2. Relative Works

There have been several who tried to propose educational computer programming system based on GUI. The research of MIT Media lab is a typical example. The purpose of this research, named Scratch [2], is providing a computer programming experience for children using a graphic environment.

Scratch is developed in Smalltalk language based on Squeak. Compared to C and C++ language, it helps people learn the principle of programming more easily by block gesture, which works like dragging the blocks and making block towers. The characteristic of the educational programming method, which is based on block helps elementary school students understand programming easily by directional language and has a positive effect on studying immersion [3,4,5].

Recently, educational programming systems based on the web has been developed. Codecademy is for people who want to know coding to be able to learn the lecture online. Every step of the lecture has a goal and people start coding on TextBox following the message. After this stage, people can know the answer as they submit their answer to the server. There is another service named Code.org. The main purpose is the same with Codecademy, but Code.org adopts a method using block coding, which has similarities with Scratch and targets a lower age group [6].

Our model has different aspects compared to the educational programming programs as mentioned above. The former educational programming programs are focused on learning programming languages, but our model is focused on learning the theory of computer graphics. Learning programming languages and learning the theory of computer graphics are totally different areas. Also, they are different structurally and technically because of the computer graphics [7]. Learning programming language acquires grammar but learning computer graphics is focused on acquiring the principle of object, camera, and lights. In addition, after going through enough theoretical study and using a graphic library, users are able to use the rendering tool based web browser that could make code. Extending the [8] paper, we approach computer graphics programming on making functional blocks using WebGL.

3. Our Method

3.1. Outline

We focus not on using programming language but the theory of computer graphics. Therefore, this model excluded repetitive and conditional statements to concentrate on the theory. Also, it was designed based on black combination that users need not consider syntax error and compile errors that occur in programming. Furthermore, it provides a step-by-step process for the user to learn systematically. Finally, users can use block modeling to provide contents and feel interesting after learning.

3.2. Core Model

The core of this educational computer graphics software is so that people can understand the principle of graphics easily. Therefore, it has four functions: programming based on web and block, tutorial system, judgement system, to satisfy this purpose. Users access the Internet and follow the tutorial process related to graphics. The tutorials are divided into 3 chapters according to the principal theory and there are goals to achieve in the chapters respectively. If users achieve the goal, the program judges this answer and they can go to the next step if the answer is right.

3.2.1. Web based: The people who are first introduced to computer graphics learn computer graphics by OpenGL, which is the base of computer industry standard application programming interface (API). However, because it has to link to integrated development environment (IDE) to use OpenGL, its accessibility and availability has been considerably lowered. The purpose of our program is to make users understand the principle so complex installation environments should be avoided. Therefore, we propose a web based model that doesn't have to preinstall the extra software and will be accessible anywhere and anytime [9].

3.2.2. Block-based Programming: People who first experience the computer graphics feel it is hard to learn the computer graphics because of the complex library structure. However, people can learn programming not through coding but through the block making, which has a rule of mapping the blocks to core functions in our model. This means that users can focus on only the functions role for them to learn the graphics theory easily.

Therefore, blocks are mapped to API functions that are used in OpenGL. It helps in learning the role of functions by creating block structures and concerns the result of rendering in real time. However, if all of the blocks that build one object are mapped together, the structure will be complex because there are a number of blocks. Therefore, the basic blocks and the explanation of functions are attached in advance, and the principle blocks that help users understand the core principal are used so users can understand them directly.

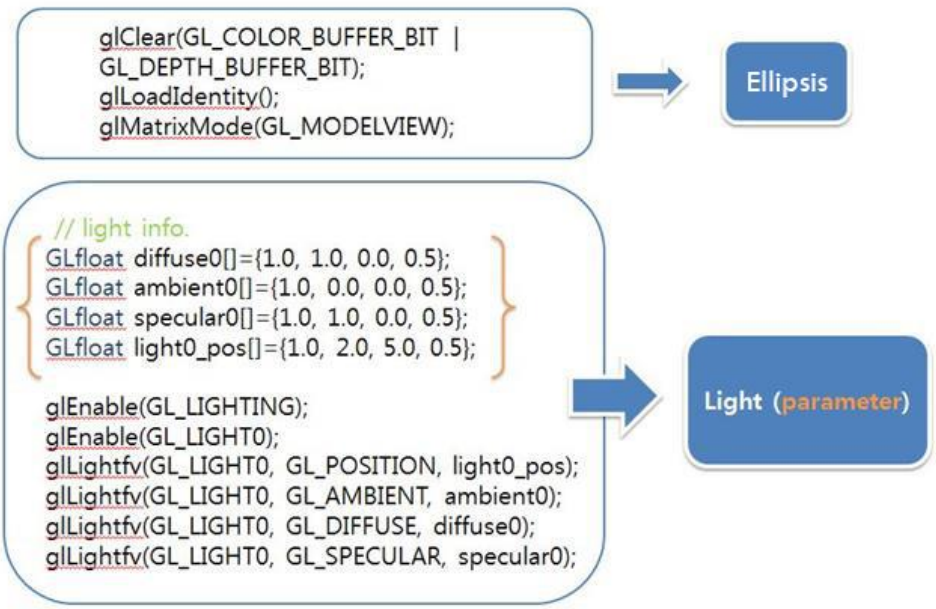


Figure 1. Block Abstract

For example, [Figure 1] is the function that makes lights in OpenGL. Our model uses blocks to reduce repetitive steps and expresses concisely because lights have to set position, ambient, diffuse, specular values using the glLightfy function. Also, the initialization functions such as glClear, glLoadIdentiy, glMatrixMode are described in letters and these are unique only to block combination so users can focus on the principle of light.

Each block has unique parameter values and users combine these blocks to make objects or manipulate these values. Basically, blocks are situated in specific spaces to

make combinations using the drag and drop methods. Also, whenever blocks are combined, the result is rendered in Viewport and users can check their result in real time.

3.2.3. Tutorial System: This paper focuses on making the difficult theory of computer graphics easier to understand. When users begin the tutorial, the web shows an explanation of knowledge related to that chapter. It then displays the result derived from that theory and lets users combine the blocks to make the same result. As a result, users combine blocks by themselves, check the result in real time, and understand the role of each one of the blocks that they operated in their work.

The tutorial was designed to learn the basic principle, 2D and 3D, Matrix and camera controlling. In 2D and 3D chapter, users can learn how to draw the object. The tutorials progress gradually and users learn 2D at first because it is easier than 3D. In 2D, users learn how to draw point, line, and face and then how to draw 2D objects such as triangles, rectangles, and the like. If users are done with the 2D tutorial, users then move on to learning to make 3D objects such as cuboid.

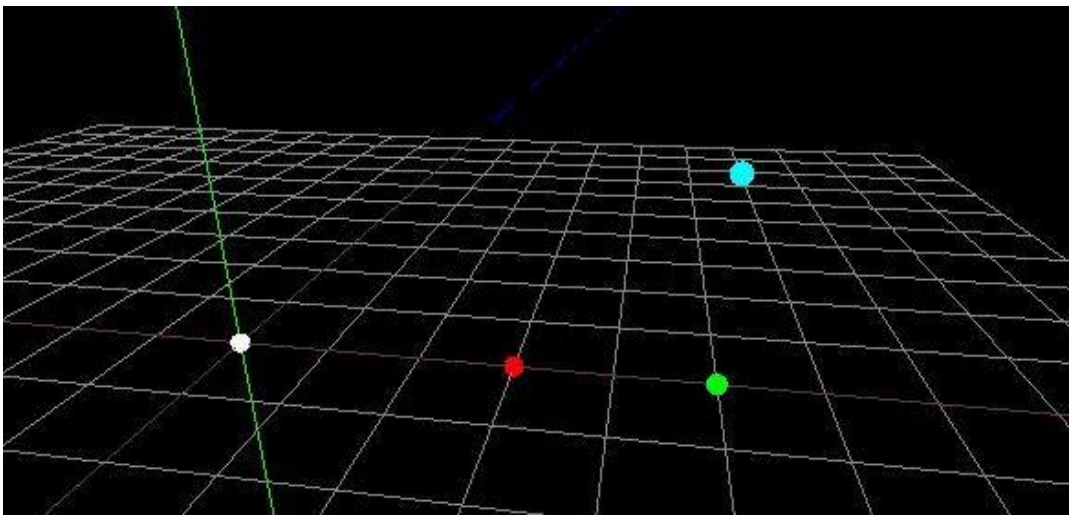


Figure 2. Current Transformation Matrix Visualization

The most important concept in computer graphics, especially in the graphics pipe line, is coordinate transformation. The point of object is drawing on screen following the many steps that the coordinate had in planning, the coordinate that gathered several objects in one scene, the coordinate following to the point of view and so on. The coordinate can be divided into three parts: a model coordinate that was designed expedient to do so, a global coordinate that gathered all of the objects in one scene, and a view coordinate that could change following the point of view. When running the GL program for the first time, the global coordinate corresponds with the model coordinate. However, if there are various changes applied such as moving, rotation and scaling, then the global coordinate and model coordinate would be separated. Also, the view coordinate can be changed by the movement of camera. The designed objects or objects from graphic library can be used to compose the scene directly, but the conversion process is required in general. This process is called model conversion and this means that in general, the shape of objects is changed by applying the transition. Model conversion has a close relation to the view transition and the view transition is the same concept in the camera manipulation, so this will be explained in camera manipulation chapter. Therefore, in this chapter, the movement of objects, rotation and scaling would be explored. The value of status variable is important, which is used in GL. All of the pipeline process is applied based on the current status variable. Therefore, CTM (Current Transformation Matrix) is also important in the transition matrix as a sort of status variable. In other words, the object transition is a stack

of piled up matrixes. It helps the users understand visible transition processing such as CTM. Therefore, the internal realization is stacked and designed by the CTM coordinated using current CTM that is changed in user UI [Figure 2].

In computer graphics using OpenGL, the change of model coordinate is applied to the current transition matrix and the final object is drawn based the current transition matrix. The current transition matrix is a sort of complex matrix. The complex matrix is the final result of multiplying the many matrixes, and always a single matrix. Therefore, it is impossible to predict which processes are applied based only on the current transition matrix that is the final result. That is, the changing process of the model coordinate can't be inversely traced. In this point, the matrix stack is needed. The changing process will be traced using this stack. In OpenGL, the matrix stack is manipulated by glPushMatrix() and glPopMatrix() function and if the current transition matrix needs to be stored, the push function would be called and if the previous transition matrix needs to be restored, pop function would be called. This concept is an essential part in hierarchical structure modeling. Therefore, CTM coordinate mentioned in above designs for users to figure out the understanding by marking the point in rendering space.

The view coordinate is defined by three factors in GL. These are the position, target position, and orientation of camera. Because the orientation of camera is that through which the same object is observed, this object can be observed vertically or horizontally. The view coordinate is determined by gluLookAt function in GL. However, users find it hard to use because gluLookAt function has 9 parameters. Therefore, by separating the camera function, position, target, and direction of camera, users will feel it is easier to use this function. Using perspective and orthogonal that is the concept of projection, users learn the camera. In tutorial, users use perspective and orthographic camera and understand the difference between two cameras directly.

Users can learn the control of light in the last chapter. Applying the light and lighting is the works that show the color of object. That is, this is the process that calculates the power of light at a fixed point by concerning the light source and characteristic of object. Lighting is an important part of the computer graphics because the figure of the object can only observed by lighting. The parameters of functions are composed by lighting model in OpenGL, but by the conception, positioning, direction, and kinds of light and lighting in tutorial. Also, as mentioned above, the complex lighting function is simplified as block.

3.2.4. Evaluation System: Our model needs a method that checks the users' understanding and ability to follow the tutorials because the purpose of this model is education. Therefore, each tutorial has a goal to achieve, and the model evaluates the results that are combined as our intent [Figure 3].

All objects can be described by the combination of vertex in computer graphics. This vertex has x,y,z coordinates and means the front or back of object in that pointed order. This is used in evaluating and learning objects. For example, a triangle can be made using three vertex A,B,C. These three vertex can be combined 2^3 numbers. If users know the order to draw object is CCW method and given start vertex A, they can evaluate that the answer is drawing the vertex A,B,C counterclockwise. It does not matter whether they follow the order or not, but the shape of a rectangle can be changed to follow the order. Therefore, it evaluates the parameter value by checking the order of drawing vertex and the value, which was in the range of our intention in the case of light or camera.

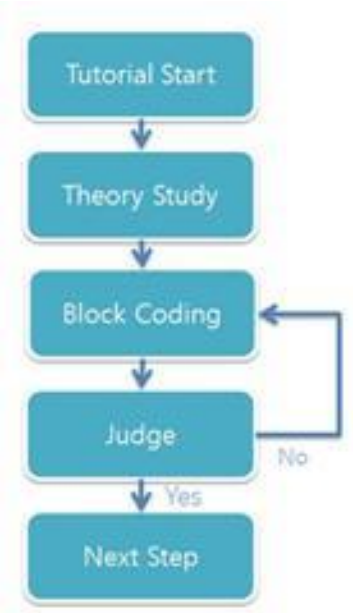


Figure 3. Tutorial Flow Chart

3.2.5. Processing Web: After learning a broad understanding of graphics through tutorials, it produces the chance to use code that is really graphics programming out of block-based programs. Using graphics libraries such as OpenGL, an integrated development environment is required and operates together between the integrated development environment and graphics library following the operating system. OpenGl is an open source but most tools of building integrated development environments is a business program and the process of working together is complex. Therefore, we develop the web tool that could be coded in web browsers.

In rendering tools, users could practice the theory of graphics using processing language. Processing is a computer graphics library that was developed by Casey Reas and Ben Fry in MIT media research group and opened in 2001. The graphics factors can be expressed easily through simple lines of functions. Also, the method of code is similar to OpenGL, so users can understand the concept of graphics programming more easily. The tool we used is based on web browsers so we should use javascript engine. Therefore, among processing libraries, we use processing.js, which supports the javascript engine. Processing.js can compile the processing language in web browsers in real-time and rendering.

After writing in the left input processing code window, click the run button and then source code is compiled internally by processing.js and rendered in the right output window. The additional functions are code assist that complete the supported code automatically, full screen mode, searching character list that all of these functions can be seen in integrated development environment, and processing grammar can be referenced in reference page directly [Figure 4].

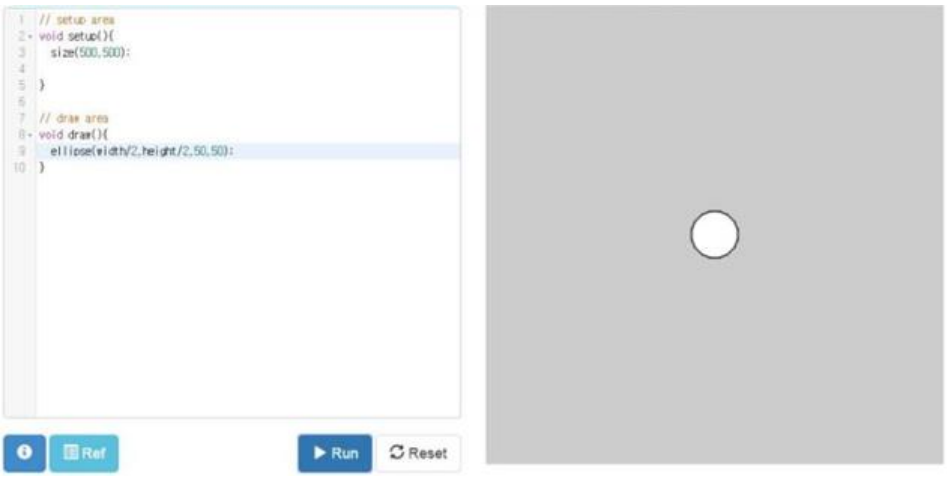


Figure 4. Processing Web IDE

3.3. System Architecture

3.3.1. Front-End Architecture: Front-End is building based on HTML5 because it runs in a Web browser. The composition of its layout uses a bootstrap that supports responsive web and realizes graphics using WebGL. WebGL is the web version of OpenGL, and it is used mainly in realizing computer graphics in web browsers based on OpenGL ES 2.0 that is the reduction version of OpenGL function [10]. Also, Three.js is the library that used to realize WebGL easily [Figure 5].

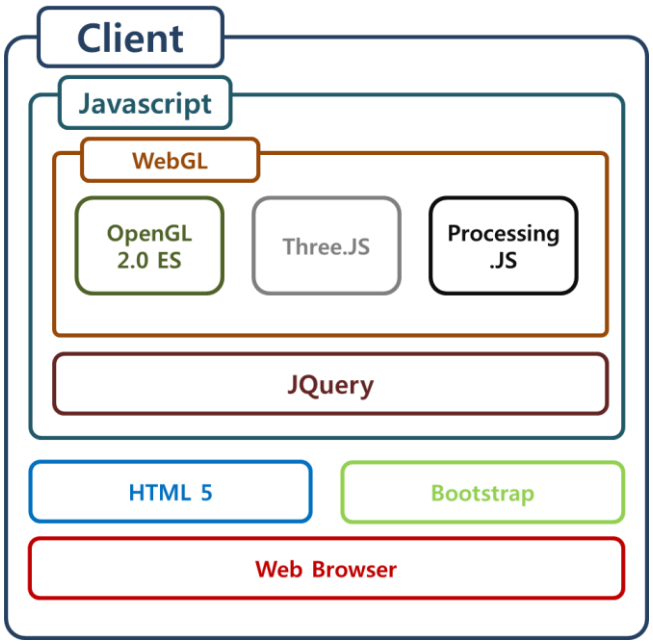


Figure 5. Front-End Architecture

3.3.2. Back-End Architecture: The composition of server is Node.js [11] and Express Web FrameWork and using these structural elements, the data of users' data and realization of application is saved by MySQL that is RDB (Relational Database). Node.js is operated by Single Thread and because of this, Node.js built in cluster module is used in applications to increase the System utilization by clustering [Figure 6]. Node.js is operated independently though composed by clustering because of a single thread; so one

must use external storage to manage the session information data. Using this storage, each of the clusters share the data. These sharing data management do not make use of the aspect of efficiency problem and instead use the memory cache in the case of database that makes use of the hard disk.

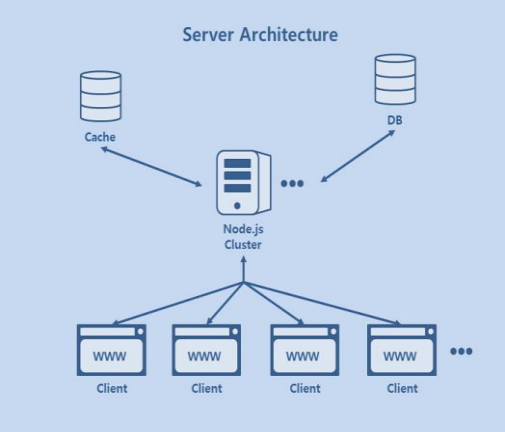


Figure 6. Node.js Clustering

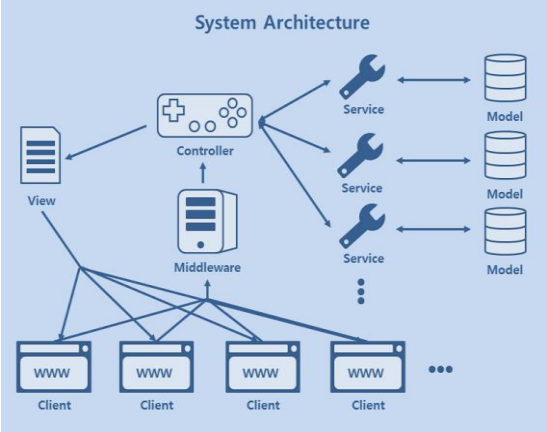


Figure 7. MVC(Model-View-Controller) Design Pattern

Basically, the composition of system software is defined using MVC (Model-View-Controller) pattern and also adds a service layer to solve the problem of mass in the MVC pattern's controller part and overlapping code. The user certification, and log and error management use middleware. When requiring clients, Node.js route dispatcher calls the suitable middleware after managed precedency parts (logging/certification). Controller calls the needed services, updates this data in View and responds, which ends the requiring process [Figure 7].

4. Result

4.1. Tutorial User Interface

The tutorial page consists of four sections [Figure 8]. The first section is about the explanation on how to combine the block and the theory of computer graphics. Users have to make the same renderer screen of the 4th section by combining blocks.

The center section has blocks that can be used in that chapter and users can combine blocks by dragging section two blocks to section three. Users can set the block's parameters in the Medal window, which can be seen by clicking EDIT button. If the block is set in 3rd section, the result can be seen in the 4th section's renderer section following the kind of blocks.

The evaluation algorithm operates when the renderer window coincides with the expected window, and then click the run button on section 1. Users can go to the next step if they set the right combination of blocks and parameters. If not, users can check the message that helps them to know which part was wrong. Additionally, users can share their information through the Q&A board among themselves.

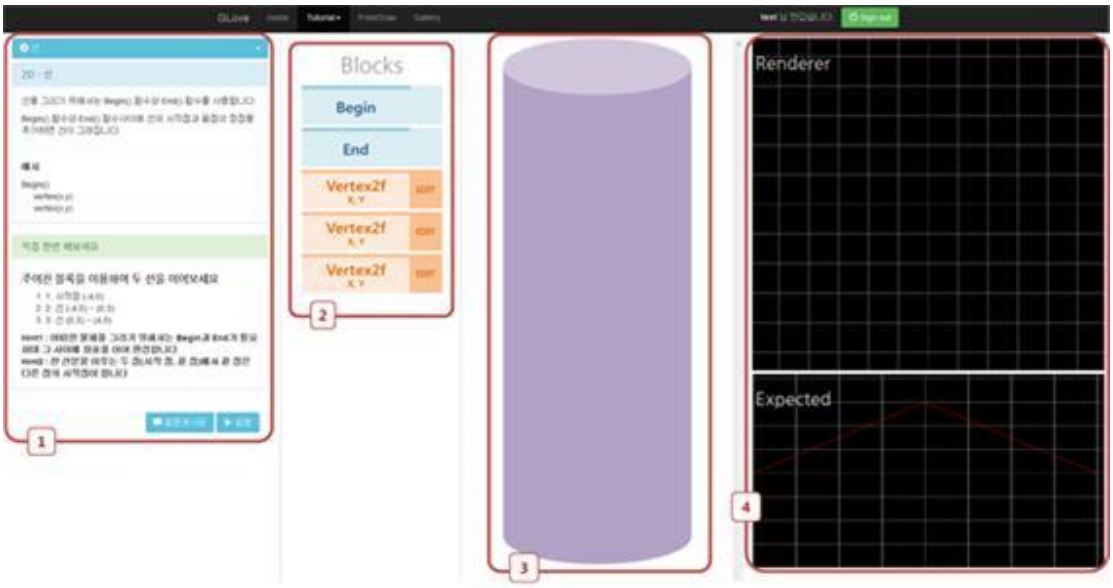


Figure 8. Main User Interface

4.2. Tutorial Content

The tutorial has 16 steps and there are 20 blocks that can be used in the tutorial [Figure 9]. Each block is mapping to each function and carries out the rendering function. Translation (Translate, rotate, scale) blocks are sensitive to the order and the result can differ following the order.

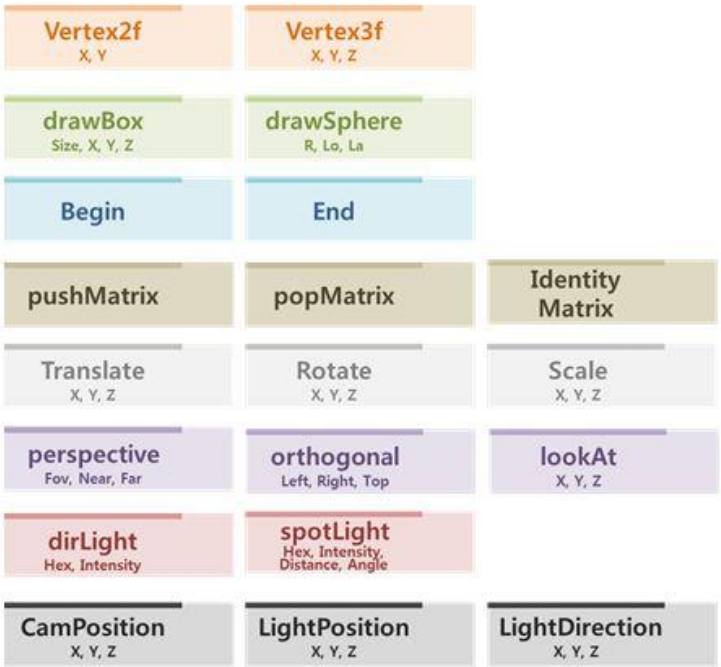


Figure 9. Tutorial Blocks

The required blocks are different in each chapter in tutorial [Table 1]. According to the above table, this model stores a database to arrange the blocks, road the chapter information that matches the block information on the client, is directed to realize exact functions to use the number and kind of all of the blocks. Each of the block parameters

has a permitted type and exception handling is designed not to set the values out of range using regular expression.

Table 1. Block List

Stage	Learning Purpose	Using Blocks	Steps
2D	Vertex	vertex 3	Step 1
	Line	begin 1, end 1 ,vertex 3	Step 2
	Triangle	begin 1, end 1 ,vertex 3	Step 3
	Rectangle	begin 1, end 1 ,vertex 4	Step 4
3D	Low API Box	begin 2, end 2, vertex 8	Step 5
	High API Box	drawBox 1	Step 6
	High API Sphere	drawSphere 1	Step 7
Matrix	Translate	drawBox 3, translate 3, idenity Matrix 1	Step 8
	Rotate	drawBox 1, translate 1, rotate 1, idenity Matrix 1	Step 9
	Scale	drawBox 1, scale 1, idenity Matrix 1	Step 10
	Push, Pop	psuh 2, pop 2, drawBox 3, translate 2	Step 11
Camera	Perspective	drawBox 1, perspective 1	Step 12
	Orthographic	drawBox 1, perspective 1, orthogonal 1	Step 13
	Position, LookAt	drawBox 3 , camera position 1, lookAt 1	Step 14
Light	Directional Light	directional light 1, lightDirecton 1, light position1	Step 15
	Spot Light	spot light 1, lightDirecton 1, light position 1	Step 16

4.3. Object Modeling

Finally, users can model the object they want if they clear all the tutorials. The using method is the same to tutorial and the order of combined block or parameter's values can gain the same result that is coded in OpenGL or Processing. Therefore, users can learn not only the theory using blocks, but also how to use programming libraries such as OpenGL [Figure 10].

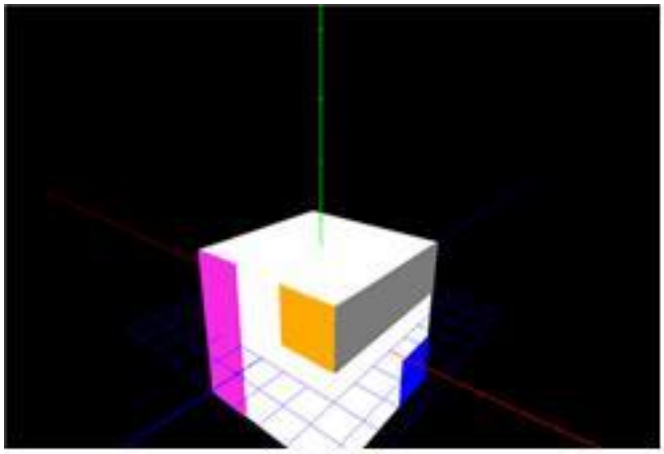


Figure 10. Object Modeling

5. Conclusion

This paper proposes a software model that can help users learn more about computer graphics by combining blocks through the web whenever they want. The main purpose of most educational software is teaching programming languages, but the purpose of this model is to study the theory of computer graphics. Users can learn theory and not be subordinate to programming languages, and also make graphics programming following the environment. Furthermore, users can understand directly as they can find out the result of rendering whenever the blocks are combined. However, some programming statements such as strong condition and repetitive statements are not included because this model is focused on the graphics theory. Therefore, animation that uses condition and repetitive statement has beem excluded, although this can be realized by applying the theory and languages that users learned in this model.

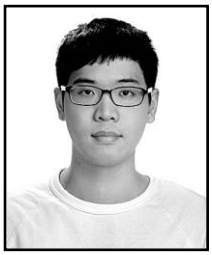
We conducted the beta test with 20 users who learned the internal alpha test and programming language during developing period. Even the user experiences a little bit of computer graphics that are easy to understand using conceptual knowledge through blocks; their concept was confirmed by real-time rendering and novice users could also learn on a level where it can create the desired object. The assessments that block based system allow are mainly increasing users' interest and enhanced engagement achieved. These are not yet complete, but it has been pointed out that the lack of UI aspect is being planned to be compensated. In addition, the problem of theoretical contents, improving directly user interface and quantitative test, and so on, are needed to be developed to include in future research. We expect that the interest of computer graphics areas will be increasing, and if people can learn graphics theory more easily, the accessibility of computer graphics will also be increasing in the future.

References

[1] S. K. Goo, "Computer Graphics, Intuition and Idea", Journal of Korean Society of Media and Arts, vol. 11, no. 1, **(2013)**, pp. 77-88.
[2] M. Resnick, "Scratch: programming for all", Communications of the ACM, vol. 52, no. 11, **(2009)**, pp. 60-67.
[3] J. B. Song, S. H. Cho and T. W. Lee, "The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability", Korea Association of information education, vol. 12, no. 3, **(2008)**, pp. 323-332.
[4] K. M. Ahn, W. S. Sohn and W. C. Choy, "The Effect of Scratch Programming Education on Learning-Flow and Programming Ability for Elementary Students", Korea Association of information education, vol. 15, no. 1, **(2011)**, pp. 1-10.

[5] S. H. Cho, J. B. Song, S. S. Kim and K. H. Lee, "The Effect of CPS-based Scratch EPL on Problem Solving Ability and Programming Attitude", Korea Association of information education, vol. 12, no. 1, **(2008)**,pp. 77-88.
[6] J. Maloney, "Programming by choice: urban youth learning programming with scratch", ACM SIGCSE Bulletin - SIGCSE, vol. 40, no. 1, **(2008)**, pp. 367-371.
[7] B. Chen and H. H. Cheng, "Interpretive OpenGL for computer graphics", Computers & Graphics, vol. 29, no. 3, **(2005)**, pp. 331-339.
[8] H. G. Pyun, H. A. An and J. Park, "WebGL based E-Learning Platform on Computer Grphics", International Workshop Education, Advanced Science and Technology Letters, **(2015)**; Jeju, Korea.
[9] E. Angel, "Teaching a three-dimensional computer graphics class using openGL", ACM SIGGRAPH Computer Graphics, vol. 31, no. 3, **(1997)**, pp. 54-55.
[10] J. Congote, "Interactive visualization of volumetric data with WebGL in real-time", Proceedings of the 16th International Conference on 3D Web Technolog, **(2011)**.
[11] S. Tilkov and S. Vinoski, "Node. js: Using JavaScript to build high-performance network programs", Computers & Graphics, vol. 14, no. 6, **(2010)**, pp. 80-83.

Authors



Hae-gul Pyun, He is an undergraduate student of Global School of Media, Soongsil University. He is currently working as an undergraduate research assistant of Visual Graphics Lab at Soongsil University.

Haeng-A An, She is an undergraduate student of Global School of Media, S oongsil University. She is currently working as an Undergraduate research assistant of Visual Graphics Lab at Soongsil University.

Jinho Park, He received his B.S. and M.S. degrees in Applied Mathematics in 1999 and 2001, respectively, and Ph.D. in Computer Science in 2007 from Korea Advanced Institute of Science and Technology. He is an assistant professor in the Global School of Media, Soongsil University, South Korea. His research interests include fluid animation and scientific visualization.

