

Application Architecture for Collaborative Terrain Editing

Jos Timanta Tarigan¹⁺, Rahmat Widya Sembiring², Maya Silvi Lydia¹, Opim Salim Sitompul¹,
Mahyuddin Nasution¹, Muhammad Zarlis¹

¹ Faculty of Computer Science and Information Technology, Universitas Sumatera Utara

² Management of Informatics, Politeknik Negeri Medan

Abstract. Generating a massive terrain in a 3D game can be exhaustive and costly. Artists as the creator of the content, are required to collaborate with the game designers and testers to be able to create a high quality and playable terrain. Hence, it is important to create a system that allows involving parties to collaborate efficiently. In this paper, we propose a software architecture that allows clients to collaborate in real-time to create a 3D terrain. The system allows multiple users using multiple devices to work on a single content concurrently. Our architecture uses web technologies such as WebGL and WebSocket to allow the solution to be ran on any devices with modern browsers. We present the functional feature and technological aspect of the future developed application.

Keywords: 3D World, Synthetic Terrain, Collaborative Work, Web Based 3D Application

1. Introduction

Modern 3D games most of the time require a 3D world as part of its content. Some games may even need a massive, detailed, and playable 3D world to increase the quality of the gameplay. In modern game industry, generating this type of content can be costly and time consuming. Using modern 3D tools may help the creation process but there are some limitations that can not be achieved by using these tools.

In 3D world, the world can be defined as a terrain to define the base of the character. Since it is mostly massive and wide, the cost of generating 3D terrain is high. Flight-based games such as Flight Simulator and Ace Combat series may require a massive, non-repetitive terrain model. Generating such content using only conventional 3D tools may require a lot of time. Moreover, creating a playable terrain will require additional process. While the creation process depends heavily on artist, it is important to have the created content tested by the designers or tester before it is installed in the game. Hence, it is important for all the involved parties to collaborate during the creation process. Furthermore, most of the tools used in the process does not allow multiple clients with multiple roles to work on an object. This condition creates a process where a content has to be created and tested repeatedly before artists and designers/testers agree the map is aesthetic and playable. [1] shows how common content creation in 3D games is created.

In this paper, we propose an application architecture that introduces a new workflow in for terrain creation process. The proposed system allows multiple clients using multiple devices to collaborate in real time. Our objective is to design a system that allows multiple clients to edit and test a map concurrently. The scope of this paper is to present the architecture, related technology, and workflow of the system.

1.1. Requirement

The objective of the system is to create and render a 3D terrain. Hence, it must the capability of rendering 3D objects. The other requirement is the system must allow multiple clients to work on a single

⁺ Corresponding author. Tel.: +62 81361606734;
E-mail address: jostarigan@usu.ac.id.

object. Consequently, our design must include a network/communication protocol to keep the content consistent among all clients.

As previously described, our current development plan is to allow artist to edit the map and game designer/tester to test it by performing a walkthrough through the map. Hence, our application is required to have two interface; editing mode and testing mode. And the last objective is to allow multiple platforms to run the application. To achieve this requirement, we decided to create a web based application.

1.2. Related Works

There are multiple 3D content creation applications available on the market. 3D Studio max, Autodesk Maya, and Blender are three of the most famous 3D application used in the industry. Some applications such as TerraGen and World Machine are built specifically to generate 3D terrain content. In common game development process, the content created by the previously described software will be imported to the game engine such as Unity, Unreal Engine, and CryEngine. Upon installation, a game tester or game designer will test the game and decides whether the content is playable or not. While all of these applications already have an optimized solution to increase the speed of the development process, there are a lots of improvement needed to create a more effective collaboration between multiple clients.

The introduction of WebGL [2] brought a wide opportunity to render 3D content on web application development. The capability and support on 3D rendering on the web is one of the most promising feature. A survey performed by Evans et. al. [3] shows that while the performance of 3D rendering on the web has been inferior compared to native 3D application, it is still considered as one of the best option to have a cross platform 3D rendering. [4] has developed a WebGL based application to create a 3D world based on ITS Campus in Surabaya Indonesia. [5] proposed a 3D web based online shopping to create a more immersive interaction between the prospective buyer and the objects on the store. Another interesting research has been done by Gulobovic et. al. [6] and Kim et. al. [7]. These two researches built a WebGL based mobile applications and shows how this technology can be ran on both desktop computer and smartphones.

While the web technologies are mostly built for a client-server type of application, it is possible to perform a client-to-client web application. The recently introduced WebSocket [8] allows developer to create a client-to-client communication protocol. [9] is an example of WebSocket implementation that uses this technology to perform a new compressed data transfer between clients.

These web technologies allow a more complex web application development. [10] proposed a framework that can be used a foundation to build an 3D online games. [11] is an implementation of WebGL to show biology-related 3D data while [12] introduced a new way to introduce mathematics formula by using WebGL to visualize the content on a web based application.

2. Application Architecture

In this section, we introduce the overall architecture of the proposed application. We built this design based on the previously described requirements and in regards with the recent technology and previous related works on the technology.

2.1. System Design

Based on our current requirement, the program will have two type of interface; editor and tester. User can freely change the mode in the application. In editor mode, users perform map editing by painting a 2D image which will be used as a height field [13]. The application allows users to paint a 2D image which represent the height field of the 3D terrain model. Similar to a traditional paint application, user may edit and change the height of the map by manipulating the pixel of the 2D image. Switching the application to a tester mode allows user to perform a walkthrough through the map and interact directly to it. User will able to perform a basic movement (forward, backwards, left, and right). Other features such as gravitation and game mechanics will be added in the future.

2.2. Communication Protocol

The proposed application uses WebSocket as a communication standard between client. Since we allow multiple clients to edit a single content, we have to create a communication protocol to guarantee the data consistency amongst client. [14] introduces an interesting research regarding consistency maintenance protocol in collaborative graphics editing systems. Based on these research, we built our own protocol for our application. Our proposed protocol consists of three steps; initialization, editing process, and serialization.

Figure 1 illustrates the collaboration initialization process. The initialization defines the beginning of the collaboration process when a user sends a request to edit a map in the map server. If there is no collaboration process on the requested map, the server will set the client as the map's server and store its IP address in the database. If a new client request for the same map, the server will send the map's server IP address and the client will collaborate with the map's server during the terrain creation process.

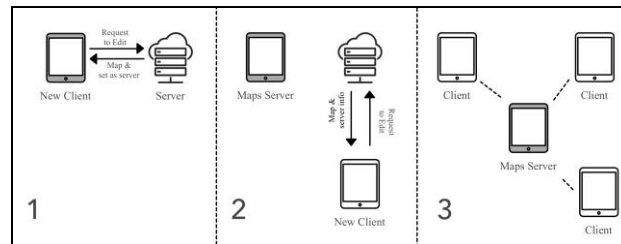


Fig. 1: Collaboration Initialization Process Diagram

During the collaboration process, one client will act as a Map Server. Any changes made by one of the client is sent to the server before it is applied on the content. The server will then sort the changes and sent them to all collaborating clients. To avoid inconsistency in editing sequences, a sequence number will be assigned in each changes. This process is done by the map's server. Should a client missed a sequence during the editing process, it may request the server to resend the missing sequence. This case is quite common in an unstable network environment. Figure 2 illustrates the previously described data synchronization process.

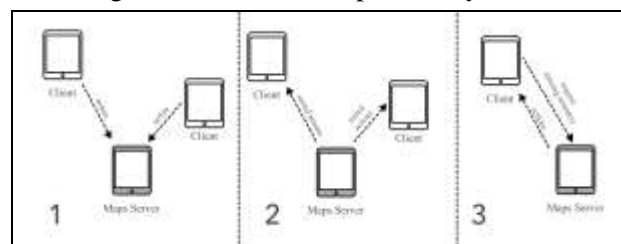


Fig. 2: Data Synchronization Process Diagram

The collaboration session is ended when the server exits the session. When this happened, the remaining clients will also quit the session. If those clients decided to continue to work on the map, they have to create a new session and the first user that reached the server will act as the new map server.

3. Terrain Generation

The objective of this application is to generate a 3D terrain based on user's input. As previously stated, our system generate the terrain based on 2D image created and modified by user(s). However, creating a 2D image that able to represent an aesthetic 3D terrain can be a complicated task. A terrain may consist of many details such as random height changes. Drawing those details may be time consuming especially using a traditional computer input such as mouse.

To solve this problem, we decided to support the content creation by using an implementation of procedural generated terrain. There are many researches focusing in creating a new and more effective method on procedural generation content. [15], [16], and [17] are examples of generating a specific content, such as canyons and mountains, using procedural method. [18] introduced Charack, a real time procedurally generated world able to create a pseudo infinite virtual worlds. [19] and [20] introduced a new way to generate content based on gameplay and current story which in results may offer a new map based on the game progress and enrich the user's experience [21] proposed an interesting concept to create a procedurally generated terrain based on a simple 2D image. The algorithm proposed on this paper allows user to use a 2D

images as a base reference for the algorithm to create a terrain. The algorithm will then generate noises to make the terrain more aesthetic. Another interesting work regarding procedural generated terrain has been done by Patel et. al. [22]. Similar to the previous work, this work can also generate a terrain based on an 2D image as a reference and has been used in one of his web based game.

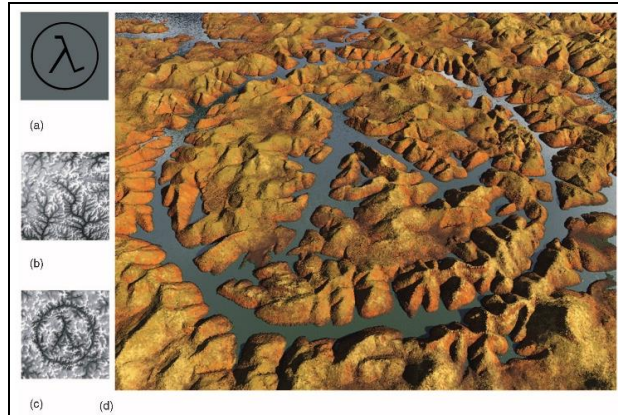


Fig. 3: 3D Terrain based on a sketched half-life symbol (a) user sketch used as a base reference, (b) Mount Vernon, (c) Synthesis result, (d) Rendered Terrain. Image courtesy of Zhou et. al. [21]

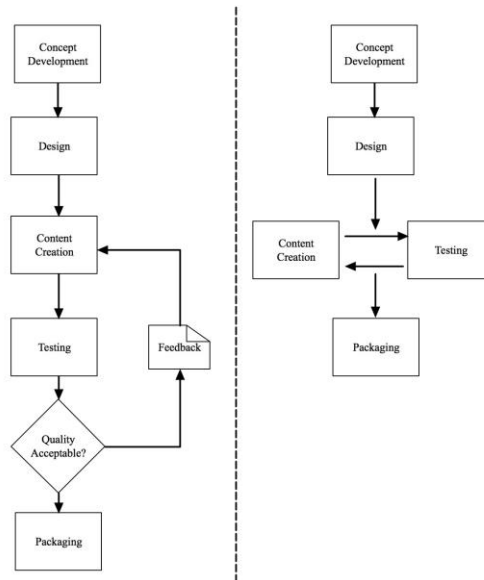


Fig. 4: The Current Prototyping Development Process (Left) compared to the proposed Development Process (Right)

Figure 4 above illustrates the comparison between the conventional workflow to our proposed workflow. While the difference is minor, the affect to the development process can be significant. Reducing communication process between designer and tester may increase the speed of the development process. Moreover, involving game designer in early stage of the creation process may increase the gameplay quality of the content.

4. Collaborative Workflow

Our primary objective is to change the current workflow of content generation, especially in creating 3D terrain development. By allowing multiple devices to work concurrently, we are able to redefine the creation process. Content creator (such as artist) are now able to work in parallel with designers and testers.

The most common software development paradigm used in modern game development is prototyping [23]. Essentially, this paradigm requires an iterative process of creating and refining the product. Creators repetitively build a smaller but working version of the product and tester evaluate the result. Based on the feedback from the tester, creators refine the product and expand the prototype. This concept is commonly used in software development since it allows developers to visualize the end product at earlier stage.

However, this paradigm has an issue regarding the time consumption. The iteration between creators and testers requires additional time before both party can agree on a result. The form of communication and the flow of iteration affect the time significantly. Our system offers a new way of performing the iteration which minimizes the communication between two parties. By allowing both party to work concurrently, tester/designer may evaluate the product during the creation process. Hence, it might reduce the disagreement at the end of the process.

5. Conclusion and Future Work

The objective of this paper is to present the architecture of our collaborative terrain generator. We decided to create a web based application to allow various devices to run the application. The system itself will use the currently available web technologies such as WebGL as a 3D rendering library and WebSocket as an inter device communication. We have made a design regarding the communication protocol and the terrain rendering and collect related works as future references.

The next step of this research is to implement the system and create the application based on the current design. While the current design only specifies the ability to create a terrain, we would also like to expand the functionality so the system can be used to create a full 3D worlds and the objects on it (such as vegetation, rocks, and water). To measure how our system may affect the game development process, we are planning to have the end product to be used as a tool in game industry and observe the impact on game development lifecycle.

6. Acknowledgment

The authors wish to thank the Head of Research Department in University of Sumatera Utara, Prof. Erman Munir, Ph.D and team for the support during the creation of this research. The authors would also like to thank the Rector of University of Sumatera Utara, Prof. Dr. Runtung Sitepu, SH., M.Hum for the opportunity and support during the process of this research.

7. References

- [1] M. Labshutz, K. Krosi, "Content Creation for a 3D Game with Maya and Unity 3D", in Proceedings of The 15th Central European Seminar on Computer Graphics, 2011
- [2] Khronos Group, WebGL Specification, 2014 (available online: <https://www.khronos.org/registry/webgl/specs/1.0/>)
- [3] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, J. Blat, "3D Graphics on the Web: A Survey" Computer and Graphics, vol. 41, pp. 43-61, 2014.
- [4] A. Atminanto, A. Mardasatria, R.R. Hariadi, N. Suciati, "3D ITS Campus on the Web: A WebGL Implementation", in Proceedings of 2015 International Conference on Information, Communication Technology and System, 2015, pp. 141-144.
- [5] J. Geelhaar, G. Rausch, "3D Web Applications in E-Commerce - A Secondary Study on the Impact of 3D Product Presentations Created with HTML5 and WebGL", in Proceedings of 14th International Conference on Computer and Information Science, 2015, pp. 379-382.
- [6] D. Gulobovic, G. Miljkovic, S. Miucin, Z. Kaprocki, V. Velisavljev, "WebGL implementation in WebKit based web browser on Android Platform" in Proceeding of 19th Telecommunications Forum (TELFOR), 2011, pp. 1139-1142
- [7] K. Kim, S. Kang, K. Lee, "Mobile 3D Fusion Application Based on HTML5 WebGL" in Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2013, pp. 1700-1702
- [8] The WebSocket Protocol, Request for Comments: 6455-2011
- [9] T. Anusas-armornkul, C. Silawong, "The Study of Compression Algorithms for WebSocket Protocol" in Proceedings of 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON), 2014, pp. 1-6.
- [10] B. Chen, Z. Xu, "A Framework for Browser-based Multiplayer Online Games using WebGL and WebSocket", in Proceedings of International Conference on Multimedia Technology, 2011, pp. 471-474

- [11] H. Li, K. Leung, T. Nakane, M. Wong, "iview: an interactive WebGL visualizer for protein-ligand complex", *BMC Bioinformatics* [Online], 2014, Available: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-56>
- [12] M. Hennig, D. Gaspers, B. Mertsching, "Interactive WebGL-based 3D Visualizations for Situated Mathematics Teaching" in *Proceedings of 12th International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2013, pp. 1-6.
- [13] G. S. P. Miller, "The Definition and Rendering of Terrain Maps", in *Proceedings of the 13th annual conference on Computer Graphics and Interactive Techniques*, 1986, pp. 39-48.
- [14] C. Sun, D. Chen, "Consistency Maintenance in Real-Time Collaborative Graphics Editing System", *ACM Transactions on Computer-Human Interaction*, vol. 9, no. 1, pp. 1-41, 2002
- [15] N. Mikulicic, Z. Mihajlovic, "Procedural Generation of Mediterranean Environments", in *Proceedings of 39th International Convention on Information and Communication Technology, Electronics, and Microelectronics (MIPRO)*, 2016, pp. 261-266
- [16] D. M. D. Carli, C. T. Pozzer, V. Schetinger, F. Bevilacqua, "Procedural Generation of 3D Canyons" in *Proceedings of 27th SIBGRAPI Conference on Graphics, Patterns, and Images*, 2014, pp. 103-110
- [17] K. R. Kamal, M. Kaykobad, "Generation of Mountain Ranges by Modifying a Controlled Terrain Generation Approach", in *Proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008)*, 2008, pp. 527-532.
- [18] F. Bevilacqua, C. T. Pozzer, M. C. D. Ornellas, "Charack: Tool for Real-Time Generation of Pseudo-Infinite Virtual Worlds for 3D Games", in *Proceedings of VIII Brazilian Symposium on Games and Digital Entertainment*, 2009, pp. 111-120.
- [19] Q. Yu, R. Crawfis, "Gameplay-driven Terrain Generation in Scorched Earth", in *Proceedings of 20th International Conference on Computer Games*, 2015, pp. 126-130.
- [20] J. Togelius, M. Preuss, N. Beume, S. Wessing, J. Hagelback, G. N. Yannakakis, "Multiobjective Exploration of the StarCraft Map Space", in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, 2010, pp. 265-272
- [21] H. Zhou, J. Sun, G. Turk, J. M. Rehg, "Terrain Synthesis from Digital Elevation Model", *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, issue. 4, pp. 834-848, 2007
- [22] A. Patel, "Polygonal Map Generation for Games" [Online], 2010, Available: <http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/>
- [23] J. Gibson, "Introduction to Game Design, Prototyping, and Development", Chapter 1: Thinking Like a Designer, pp. 3-17, Addison Wesley, 2015.