Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 02, February-2017

Procedural Content Generation for Emotional Virutal Characters using Behavior Tree

Jinseok Seo
Division of Digital Contents Technology
Dong-eui University
Busan, Korea

***Abstract*— This paper introduces the procedural generation technique for the behavior of a virtual character in virtual reality contents. We implemented the AI of a virtual character with parameterized behavior tree, and which is controlled by an emotion engine. As a result of implementation, our virtual character became emotional and had its personality. It behaves differently according to users' reactions and its current emotional state, and the personality also changes continuously.**

Keywords—Virtual Character; Procedural Content Generatoin; Emotion Engine; Behavior Tree

I. INTRODUCTION

Recently, various examples of VR (Virtual Reality) applications are emerging due to advancement of hardware and software for virtual reality systems including HMDs (Head Mounted Display), and it is predicted that VR will be used in many fields of daily life in a near future. However, many experts point out that rich and high-quality VR content is far more important than hardware in order for VR technology to become more popular. For this reason, VR contents including virtual characters, various virtual objects, and interaction mechanisms will play a key role in the popularization of VR technology.

It is natural that it takes a lot of time and effort to produce rich and high-quality VR contents that meet the needs of various users. In addition, as the VR market grows, we will be faced with the need for massive scalability with quantitative demand for content. In the field of computer games, which has been growing on the basis of popularization and commercialization, procedural content generation (PCG) technique has been tried in various fields in order to solve this problem.

In this study, we apply the PCG technique to virtual characters for a VR content. The content we are producing is a VR tourism that guides users around mausoleum of the First Qin Emperor [Fig. 1]. When users visit the virtual mausoleum, a virtual terracotta warrior appears to guide tourists. The main purpose of this study is to apply the PCG to the virtual character to react dynamically according to the responses of users.

The composition of this paper is as follows. In Chapter 2, we describe of various case studies on PCG and examples of commercial games. Chapter 3 explains example scenarios with the system developed to apply PCG to the behavior of our virtual character. Finally, Chapter 4 concludes.

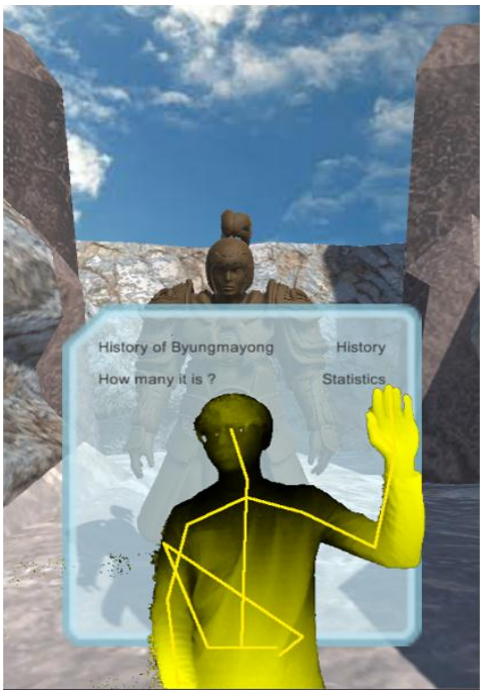

Fig. 1. Our system under demonstration

II. RELATED WORKS

A. Researches on PCG

It is quite natural that it takes a lot of time and effort to produce rich and high-quality game contents that meet the needs of various users. In addition, the larger the game market, the more demand for content will increase and the greater the scalability of content. Recently, procedural content generation (PCG) techniques using various kinds of algorithms have been studied to solve this problem.

The primary purpose of PCG is automating or helping to create various content within a game. However, there are many practical difficulties to apply PCG to a game. An example of a typical difficulty is that the content generated using PCG should be artistically complete, satisfy the requirements of the artist, and be interesting to users at the same time.

This work was supported by the ICT R&D program of MSIP/IITP (I5501-16-1016, Instant 3D based Join & Joy content technology).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 02, February-2017

At Delft University in the Netherlands, for the first time, the PCG field was investigated and summarized in depth [1]. They divided the game content into 6 layers depending on the depth of the object to which an algorithm applies. These layers have low-level contents such as textures and sounds as you go down the content layer pyramid, and more abstract contents such as maps, environments, stories, and leader boards is placed on top. Based on these layers, they investigate layers the existing PCG techniques are applied to, and provide guidelines for using the PCG technique. In addition, this research concludes that the PCG technique applied to one layer can be applied to other layers as well. In this research, we tried to apply the genetic algorithm used in various fields to the movement pattern of enemy aircrafts in a scrolling-shooter game.

Shaker et al. [2][3] conducted researches to automatically generate various types of platform for the optimal experience of game players in the "Infinite Mario Bros" game released as open source. They defined the elements representing the game level and the features of a player's gameplay as variables and predicted the user 's game experience using sequence mining and neural network. An evolutionary algorithm was used to find level designs that maximize a desired effect among challenge, engagements, and frustration.

ITU's Liapis et al. [4] automatically created the mesh models for space crafts by applying the FI-2Pop (Feasible-Infeasible Two-Population) genetic algorithm to the CPPN-NEAT (Compositional Pattern Producing Network – Neuro Evolution of Augmenting Topologies) algorithm. In order to define the function for evaluating the generated mesh, symmetry, weight in the bottom half, weight in the middle third along the x-axis, weight in the middle third along the y-axis, containment within a forward-point triangles, simplicity, and jaggedness are quantified.

Togelius et al. proposed a method for automatically evolving tracks in racing games [5] and a method for automatically generating maps in a real-time strategy game, Starcraft [6]. In our previous research [7], a sort of PCG technique was proposed to diversify the patterns of enemy aircrafts' movements in scrolling-shooter games. We presented a genetic based AI algorithm that collects data about a player's gameplay results and enemies' movements, analyzes the data, and provides new movement patterns of enemies when the next level starts reflecting the analyzed results.

In addition to above researches, we can find many PCG algorithms applied to various games [8][9][10][11]. As can be seen from most of the researches, the commonly used methods for automatically generating content in games are based on various artificial intelligence techniques. In the future, as the game industry develops and virtual reality contents become popular, more PCG algorithms based on artificial intelligence techniques are expected to be applied.

B. PCG in commercial computer games

Moss explained examples of PCG that developers need to know in Gamasutra's article entitled "7 users procedural generation that all developers should study." In this section, we analyze five games among the games introduced in the article.

Crusader King II, a strategy simulation game with a medieval background, applied PCG to the personalities of characters in the game. In this type of game, the progress of the game depends on the detailed nature of each character, so that the personality that is generated controls the central system of the game. The personality determined by the various traits combines with another personality to form a very complex family tree.

In Shadow of Mordor, PCG is applied to the creation of an orc, which is an enemy character. All orcs appearing in this game are unique. The name, appearance, way of talking, and relationship with other orc are all generated by PCG algorithms. In other words, this game can create a new orc constantly and infinitely. An orc is promoted to higher rank by killing a player character. Then the orc will remember the player the next time it encounters him. And, if an orc is killed or injured, the algorithm works to evolve. Eventually, users will always encounter its own unique enemy character.

In a platformer game like Spelunky, it is very important to make sure that users can not predict the structure and details of a new level each time. However, levels include both complex tactics as well as various types of tiles, and it is very difficult for game designers to properly design complete levels each time. Moreover, in recent years, the consumption rate of game content has become very fast, and the life cycle of the game has become very short. Therefore, more and more development costs are needed. This game has randomly places characters, shops, items, caves as well as the tiles and mechanics that make up the level, and since the original version was released in 2008, no player in the world has found the same level exists. Even more surprising is that the formulation of the algorithm to randomly generate levels at runtime is very simple,

RymdResa is a game in which a player operates his spaceship and survives in outer space where no one seems to be. In this type of game, the game system has to randomly place items such as planets and space crafts in outer space every time you play. In addition, the structure should be aesthetically well-completed and at the same time provide the player with the appropriate level of difficulty and fun. This game is famous for offering a very sophisticated and beautiful background as if it were designed by very skilled artists over a long period of time.

In simulation games like Civilization, level composition is. If you had to play pre-created levels by the game designers every time, this game would not be as famous as it is now. This game is famous for giving users various and interesting worries every time, so that he will not get bored. Thanks to the highly sophisticated PCG algorithms in the game, users have to worry about their strategies every time because they can not predict the world's layout or the civilization to be encountered.

III. VIRTUAL TERRACOTTA WARRIOR

In this study, the behavior of a virtual character, the virtual terracotta warrior, is implemented by PCG based approach. The AI of the virtual character is modeled and executed by a behavior tree, which is often used for modeling objects with complex behavioral models in games, and is composed of parameterized nodes that can be dynamically controlled rather than hard-coded. This chapter describes the detailed structure

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 02, February-2017

of the system and example scenarios. Fig. 1 shows a demonstration of the developed system.

A. System Configuration

The virtual character developed in this study was implemented to recognize users' location, motion, and voice and respond to the recognized data. A Kinect of Microsoft was used as the sensor. The overall structure of the system is shown in Fig. 2.

The system is divided into two modules. One is "Sensor" module that collects and manages acquired from users by sensors. It is composed of "AudienceManager" which manages data received from sensors and "IOManager" which reads data from an XML file and writes data into a file. Another module is "AI" module, which consists of "Alert Listener" that receives data about users from "Audience Manager," "GeneralBehaviorTree" that controls AI, and "General" that controls animations of the virtual character.

B. User Events

In this system, motion and speech are used as ways for the user to communicate to the virtual character. In the case of motion, the hand-lifting motion is recognized and the motion of the right hand and the left are distinguished. The hand lifting is mainly used when selecting one of the two choices. For example, it is used to choose one of the two options presented by the virtual character.

Apart from the expression that the user intentionally conveys, the system is intended to obtain the location of users and whether or not users are detected through the sensor. Currently developed systems are designed to allow up to two people to interact simultaneously. Therefore, in the middle of one person interacting, another person may be involved, or conversely, one person may leave in the middle of two people interacting. That is, the virtual character is designed to show a different response depending on the number of recognized users. The user's location can be used to calculate the distance between the system and the user, and to determine whether it is on the left or right side of the system. When there are two users using the system, the location data is used as a criterion for selecting which of the users to ask first.

Speech recognition is used when answering the virtual character's question or asking. If the virtual character requests the answer of "Yes" or "No," it is recognized if the word is spoken by voice. Even if there are several questionnaires presented by the virtual character, they are recognized when users read the questionnaires.

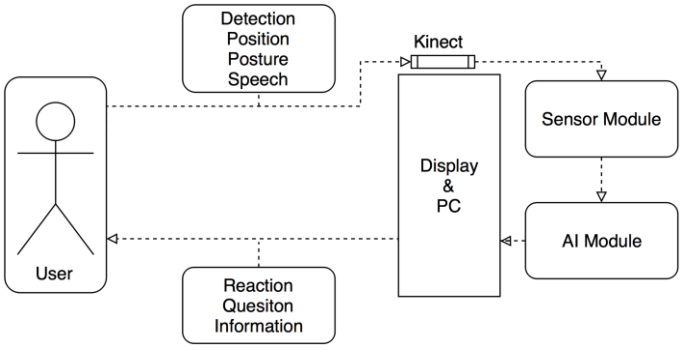

Fig. 2. The Overall Configuration of our system

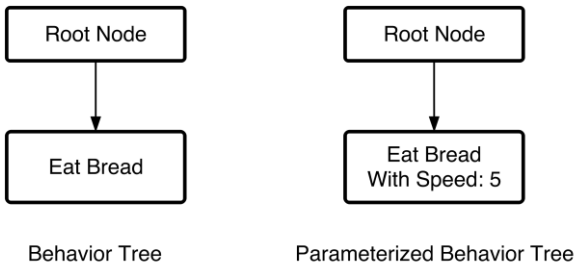

Fig. 3. Leaf node in BT and PBT

C. Parameterized Behavior Tree for Virtual Character

Events and data associated with the user obtained through sensors are managed by "AlertListener" and are used in "GeneralBehaviorTree." "GeneralBehaviorTree" is responsible for the AI of the virtual character and is controlled by a Parameterized Behavior Tree (PBT). PBT is an improved version of the Behavior Tree (BT) that is most commonly used for behavior modeling of the AI for agents in commercial games. PBT was originally proposed by Shoulson [13], but we devised a simpler version. Our PBT is able to pass a parameter when executing a specific no, so that it can perform different actions according to the parameter value. Except for the parameter, PBT we implement works very much like BT.

1) Leaf Node

Leaf node is the lowest node in Behavior Tree. It has Action node that executes its allocated function and Condition node that checks the condition. The Action node in BT simply executes the function and returns the result according to the execution result of the function. However, PBT's Action node executes a function with parameters and passes meaningful values so that different results can be obtained even if the same function is executed. Fig. 3 means to consume bread at a rate of 5. In this study, the emotion value of the virtual character is transferred to allow the virtual character to behave according to its personality. On the other hand, Condition node is not different from BT.

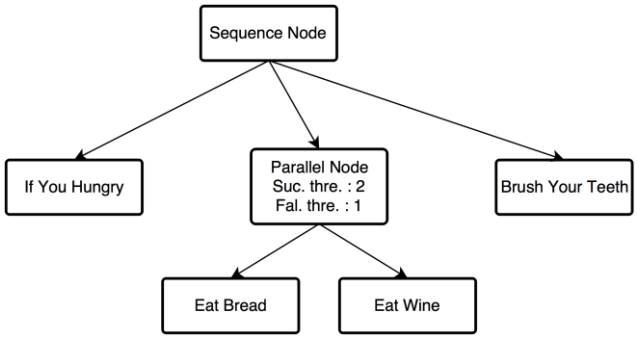

Fig. 4. Parallel node in PBT

2) Composite Node

Composite Node of PBT consists of Priority, Sequence, and Parallel node. Priority and Sequence node perform the same function as BT. Parallel node requires the threshold values for SUCCESS and FAILURE when it is created [Fig. 4]. It does not execute its descendants in order, but run them all at once and aggregate the results. After all of its descendants are executed, the result of it is determined with the aggregated results. First, it checks that if the sum of SUCCESS is higher than the threshold value for SUCCESS. If the sum is higher, it returns SUCCESS. If it is low, it

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 02, February-2017

checks that if the sum of FAILURE of its descendants is higher than the threshold value for FAILURE. Similarly, FAILURE is returned if the counted number is higher than the threshold value. If both checks are not satisfied, it returns RUNNING. Parallel node is useful when you need to run multiple functions at the same time.

3) Decorator Node

Unlike Composite node, Decorator node can have only one node as its child. Inverter, Succeeder, and Failer node is same as those of BT. However, Repeater, Repeat Until Condition, Limiter, and Max Time node are executed differently depending on the parameter values. In this paper, only the example of Repeater node is explained [Fig. 5].

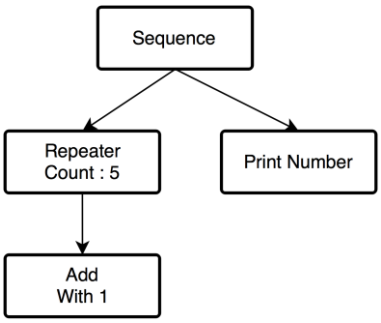

Fig. 5. Repeater nod in PBT

Repeater node receives the number of iterations at the time of creation. It repeatedly executes it child with the number of times passed and returns SUCCESS when the iteration is completed. If the result of its child is RUNNING or ERROR, it returns immediately. To illustrate the example in Fig. 5, Repeater node repeats "Add" node five times. If "Add" node does not return FAILURE, "Print Number" node will be executed.

4) Completed PBT for Virtual Character

The behavior model of the virtual character completed using PBT is very complex. Fig. 6 shows a partial excerpt. It is designed to greet a user when he is detected and he comes close to the system. In this PBT, Priority node checks whether the detected user is registered in the system or a new user. If a new user is detected, the virtual character greets the user. If there are no registered users and no new users are detected, the virtual character continues to run the "Idle Action" node to wait for a user.

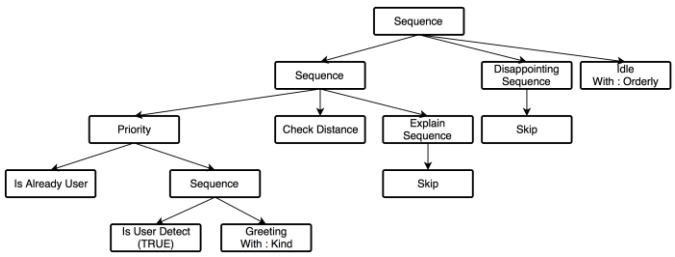

Fig. 6. Part of PBT for the virtual character (Skip node means that its descendants are omitted.)

D. Emotion Engine for Virtual Character

In this study, Extreme AI developed by Quanterm Tiger Games was used as the emotion engine. Extreme AI gives the virtual character its own personality. Personality created by Extreme AI is based on 5 factor characteristics and each factor has 6 facets. It provides 37 types of stimulus and response for easy access to the personality factors and facets. In this study, we used the calculated personality values based on Humor and Kind stimulus, and the virtual character interacts with users using response types such as Kind, Intellectual, Orderly, Affectionate, Standoffish, and Talkative. Here are just a few examples:

The virtual character's "Greeting" behavior that the system detects and greets users use the Talkative response type. If users do not come for a long time, the Talkative value is changed in a positive direction so that the virtual character shows a more active attitude when the next user comes.

The Kind response type is used for the "Explain System" behavior that explains to users how to use the system. When users listen to the end, the virtual character changes the Kind response to a positive direction.

The "Wait Question and Idle" behavior, which is waiting for users' reaction, uses the Orderly response type. Depending on the value of Orderly, the virtual character is waiting in the right posture or doing the act of boredom.

E. Interactions

The virtual character implemented in this study interacts with users using PBT and Extreme AI. There are three kinds of interaction. The first is the response to a user's request, the second is the response of the virtual character to a user when he is not responding or when a new situation comes. The third is that the avatar delivers a warning message to a user.

1) Interaction Type I

The first interaction type is when a user requests the virtual character with a motion or voice command. This occurs when a user lifts his right hand or speaks. If the virtual character does not ask the user first, the user can ask to show the list of questions by lifting his right hand. At this time, the virtual character shows a list of possible questions to the user. The question list can be edited through IO Manager with XML decoding function. The user can select a question by voice and hear the related explanation. The user can read the question as "How many it is?" or say the number like "Second One."

2) Interaction Type II

The second interaction type is when the virtual character asks a user a question first, then the user makes a desired choice, and the virtual character reacts accordingly. This also occurs when the virtual character completes the user's selected explanation or a new user enters the system.

When the virtual character completes the explanation of a selected item, it will ask the user "Do you want the next explanation? Would you like to hear it again?" The user may select one of the choices by responding to the question with his right or left hand, or he may not show any response. If the right hand is lifted, the explanation of the next item is continued. If the left hand is lifted, the item selected by the user is explained once again. If the user does not respond, it goes into the waiting state and waits for the user to select one item.

3) Interaction Type III

In addition to responding to a user's choice and request, the virtual character may also send a warning message

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 02, February-2017

depending on a user's situation. An example is a warning message related to a user's location by inducing a user to use the system properly. If a user is too close to the system, it gives a user a message, "Please step back a little." If a user is too far away, it says "Please come a little closer." If a user hears the message and does not respond, it again sends the same message a louder voice.

IV. CONCLUSION

Content production has been regarded as the exclusive property of designers and artists. However, as the virtual reality system becomes more and more popular, it can be expected that there will be a limit to manually developing everything. The PCG technique can be said to have originated from this problem. However, it is not easy for the contents created by simple PCG technique to satisfy the designers, artists and even users. In this study, to solve these problems, we tried to apply an approach to the behavior model of a virtual character by parameterizing the behavior tree, which is a game AI technique. In a near future, it will be applied not only to virtual characters but also to various content elements, so that PCG with a more extended concept will be possible.

Most experts point out that in order for virtual reality technology to become more popularized, the level of virtual reality content must be increased both quantitatively and qualitatively. However, in order to plan, design, and implement virtual reality contents, it requires a lot of time and cost in addition to the efforts of experts in various fields. In order to solve this problem, the PCG technique has a main purpose of generating contents in a procedural manner using software algorithms. Successful PCG techniques will reduce the time and effort of programmers as well as artists and planners, and will play a major role in popularizing virtual reality technology.

ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIP/IITP (I5501-16-1016, Instant 3D based Join & Joy content technology).

REFERENCES

[1] H. Hendriks, S. Meijer, J. V. D. Velden, and A. Iosup, "Procedural Content Generation for Games: A Survey," ACM Transactionson Multimedia Computing, Communications and Applications, Feb., 2011.

[2] N. Shaker, G. N. Yannakakis and, J. Togelius, "Crowd-Sourcing the Aesthetics of Platform Games," IEEE Transactions onComputational Intelligence and AI in Games, issue. 99, Dec., 2012.

[3] N. Shaker, G. N. Yannakakis, and J. Togelius, "Digging deeper into platform game level design: session size and sequential features," in Proceedings of EvoGames: Applications of Evolutionary Computation, Lecture Notes on Computer Science, 2012.

[4] A. Liapis, G. N. Yannakakis, and J.Togelius, "Adapting Models of Visual Aesthetics for Personalized Content Creation," IEEE Transactions on Computational Intelligence and AI in Games, Special Issue on Computational Aesthetics in Games Vol. 4, No. 3, pp. 213-228, Sep., 2012.

[5] J. Togelius, R. D. Nardi, and S. M. Lucas, "Towards Automatic Personalized Content Creation for Racing Games," IEEESymposium on Computational Intelligence and Games. 2007.

[6] J. Togelius, M. Preuss, N. Beume, S.Wessing, J. Ha- gelback, and G.N. Yannakakis, "Multiobjective Exploration of the StarCraft Map Space," In Proceedings of the IEEE Conference onComputational Intelligence and Games, 2010.

[7] C. Park and J. Seo, "Genetic Algorithm-Based Movement Patterns for Scrolling-Shooter Games," International Journal of Engineering and Applied Sciences, Vol. 4, No. 1, pp. 5-9, 2017.

[8] C. Grappiolo, Y. G. Cheong, J. Togelius, R.Khaled, and G. N. Yannakakis, "Towards Player Adaptivity in a Serious Game for Conflict Resolution," In the Proceedings of VS-Games 2011 Natural Interactionand Player Satisfaction in Games Workshop, Athens, Greece. May 4-6, 2011.

[9] J. Hastings, R. K. Guha, and K. O. Stanley, "Automatic content generation in the galactic arms race video game," IEEE Transactions on Computational Intelligence and AI in Games, Vol. 1, No. 4, pp. 245-263, 2010.

[10] S. Risi, J. Lehman, D. B. D'Ambrosio, R.Hall, and K. O. Stanley, "Combining Search-based Procedural Content Generationand Social Gaming in the Petalz Video Game," In: Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference, MenloPark, CA. 2012.

[11] E. McDuffee and A. Pantaleev, "Team Blockhead Wars: Generating FPS Weapons in a Multiplayer Environment," FDG PCG workshop2013.

[12] R. Moss, "7 uses of procedural generatiohn that all developers should study," Gamasutra, http://www.gamasutra.com/view/news/262869/7_uses_of_procedural_generation_that_all_developers_should_study.php, Jan. 1, 2016.

[13] A. Shoulson, F. M. Garcia, M. Jones, R. Mead, and N. I. Badler, "Parameterizing Behavior Trees," Lecture Notes on Computer Science 7760, pp. 144-155, 2011.