International Journal of Technology and Engineering Studies
volume 4 issue 3 pp. 86-94 doi: https://dx.doi.org/10.20469/ijtes.4.10002-3

A Somatosensory Edutainment System Based on Plant Growth Simulation of L-system

Lieu-Hen Chen *
National Chi Nan University, Nantou, Taiwan

Ting-Hui Lee
National Chi Nan University, Nantou, Taiwan

Cheng-Yu Sun
National Chi Nan University, Nantou, Taiwan

Hao-Ming Hung
National Chi Nan University, Nantou, Taiwan

Abstract: In recent years, the somatosensory technology has been used widely in the game industry. Moreover, it is also used widely for the purpose of rehabilitation, motion capture, input/output interfaces, and so on. Meanwhile the development and application of VR and AR have also become a hot trend. In many scenes of virtual reality environments or movies, a large number of plants often appear to present a natural landscape. In the field of Computer Graphics, Lindenmayer system or L-system for short is the mainstream method of simulating the growth of plants. The L-system is a parallel rewriting system which consists of axiom, the number of iteration, angle and production rules. In this paper, we integrate the motion-capture technique of Kinect 2, the plant growth simulation of L-system, and Unity to implement a somatosensory Edutainment System. This system provides users interesting and interactive experience of growing plants in real-time. The user can swing his/her body to control the virtual tree growth in the beginning. After that, the virtual tree will swing its branches as user did, just like the Ents in the fantasy world. Finally, users can wear HTC VIVE HMD to observe their own generated trees. We look forward to increasing users' awareness and care about the environment through our system.

Keywords: *L-system, somatosensory, human-computer interaction, plant growth simulation*

Received: 20 February 2018; ***Accepted:*** 17 April 2018; ***Published:*** 25 June 2018

I. INTRODUCTION

In the very popular movie, "My Neighbor Totoro" [1], two sisters and Totoro danced in the garden under the moon. The seed planted in the soil gradually grew into a giant tree as they swung up and down.

Being inspired by this beloved scene, in this paper, we developed a somatosensory edutainment system which associated users action with plants growth simulation. Users can interact with plants. To achieve this idea, at first, the user's motion is detected and analyzed by the Kinect v2. Then the results are introduced into the grammar of L-system to control the growth rates of the plant. Finally, by repeated iterations of the formal language, the growth of the plant is simulated in a botanical manner.

Early plant images, due to hardware limitations, coupled with L-system grammars, were defined by successive iterations of plants, resulting in the dwarfing of the hard disk imaging. And now the hardware equipment can already calculate the image information in large amounts. With the continuous improvement of Rendering, the fidelity of the image is getting closer to the real image. Many people had proposed different approaches to 3D modeling of plant by L-system. Zhongke Wu et al. utilized Ball B-Spline Curves (BBSCs) to 3D modeling [2]. [3] scanned real-world tree trunk and used these trunk fragment models to combine 3D trees. For the issue of modeling plants, because our system mainly focuses on improving the user's experience of interactions, it only adopts the most basic approach to build up multiple components of a tree.

*Correspondence concerning this article should be addressed to Lieu-Hen Chen, National Chi Nan University, Nantou, Taiwan. E-mail: lhchen@csie.ncnu.edu.tw

We introduce Kinect, L-system, and Unity in related works. On system design and implementation, we detail how the system works. We show implementation in the current research.

II. RELATED WORKS

A. *Kinect*

Kinect is a somatosensory camera introduced by Microsoft. Kinect allows players to interact with system directly by using sound or body motions. Kinect is divided into two generations. The first generation of Kinect hardware is divided into "Kinect for Xbox 360" and "Kinect for Windows". The second generation of Kinect hardware is divided into "Kinect for Xbox One" and "Kinect for Windows". The former can be connected to a dedicated adapter cable "Kinect Adapter for Windows" to computers. Our system used Kinect v2. The main components of the Kinect include an RGB camera, the depth sensor, and a multi-array microphone. The color camera used to identify the player. The depth sensor includes an Infrared (IR) emitter and an IR depth sensor. The emitter emits infrared light, and the depth sensor receives the reflected infrared, by this method to capture a depth image. Kinect relies on the depth sensor to detect player's movements. Kinect captures sound by four microphones. Kinect can record audio and find the sound source and the direction of the audio wave due to the multi-array microphone [4]. Kinect is also equipped with tracking technology, the base motor will follow the movement of the focus object with the rotation [5, 6].The difference between Kinect v1 and Kinect v2 is shown in Table 1.

Fig. 1. Kinect v1

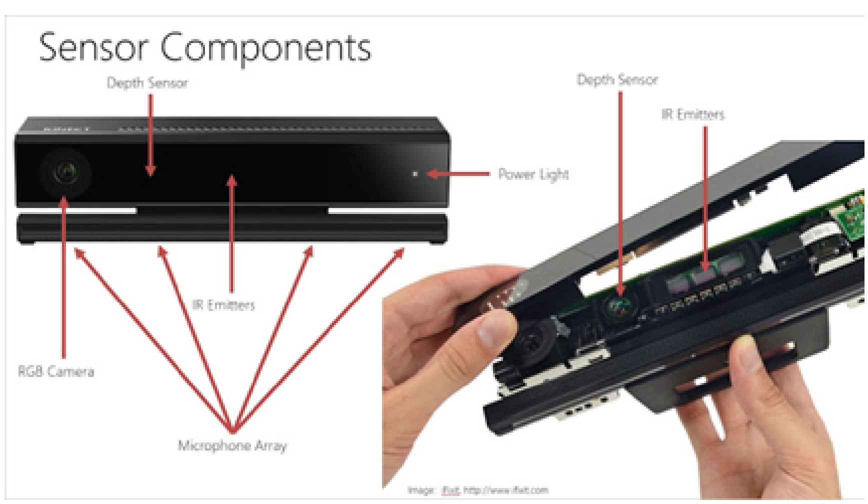

Fig. 2. Kinect v2

TABLE 1
DIFFERENCE COMPARISON WITH KINECT V1 AND V2

Kinect v1	Kinect v2
RGB camera	
640x480 30fps	1920x1080 30fps
Depth sensor	
320x240 30fps	512x424 30fps
1.2∼3.5 m	0.5∼4.0 m
Skeletal tracking	
Max: 2 Skeletons	Max: 6 Skeletons
Joints: 20	Joints: 25

The IR capabilities of Kinect v2 produce a lighting-independent view that allows Kinect2 use in the dark. Users can use RGB camera and depth sensor at the same time. Kinect v2 has more joints than Kinect v1: Neck, HandTipLeft, HandTipRight, ThumbLeft, and ThumbRight joints. Because there are thumb joints, Kinect v2 judges the hand opening or closing.

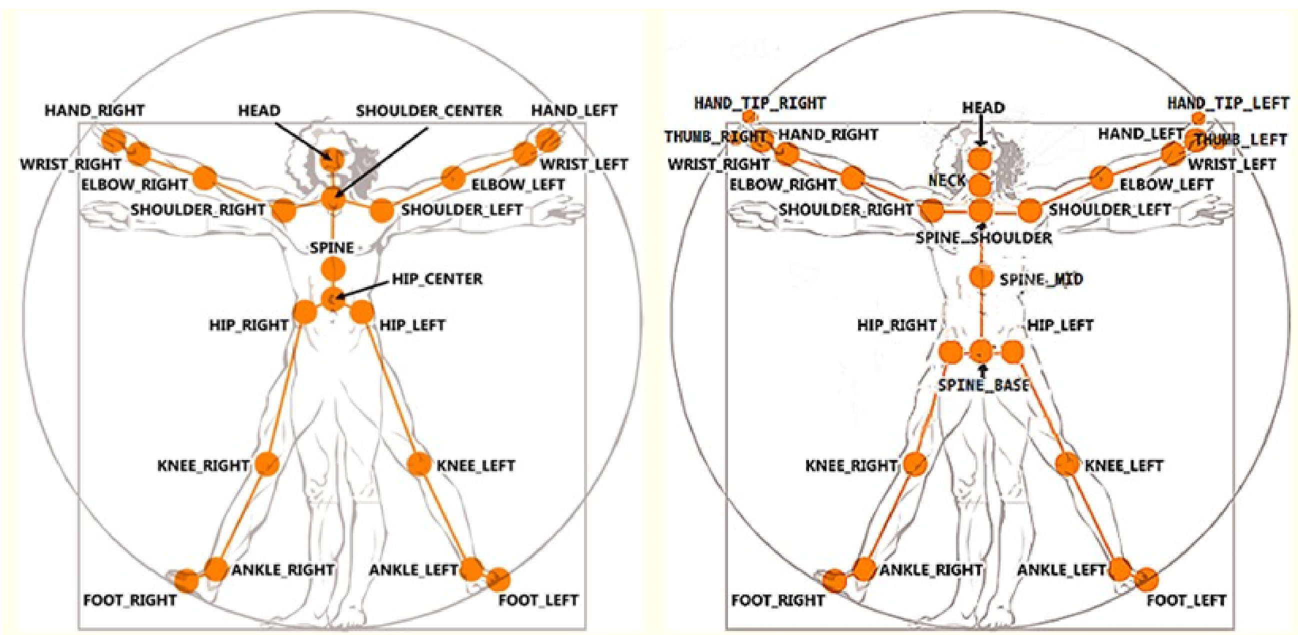

Fig. 3. Figure 3. Joint comparison picture between Kinect v1(left) and Kinect v2(right)

Kinects depth acquisition was enabled by "light coding" technology. The process coded the scene with near-IR light, the sensor read the encoded light, and then produced a depth of the image after the operation of the chip decoding. The key of light coding is Laser Speckle. As the laser light to rough objects, or through the glass, the reflective spots, called Laser Speckle, are highly random and change the pattern with distance. Speckle in any two places in space are different patterns, which means that the whole space is marked. Then, determining whether the object is moving is done by recording speckles position [5].

When Microsoft introduced Kinect initially, it did not consider launching the Windows development kit. However, due to Kinect's powerful features and relatively low prices, many users had expressed the hope that the Kinect can be used with a computer. Some manufacturers have developed a driver kit, such as CL NUI Platform, OpenKinect/libfreenect, and OpenNI [7]. Microsoft later introduced the Kinect for Windows SDK Beta for non-commercial developers. The Kinect for Windows SDK makes it easy for developers using C^{++}, C# or Visual Basic to work with Microsoft Visual Studio.

There have been many studies of somatosensory [8, 9, 10]. In recent studies, many researchers have chosen Kinect as a somatosensory camera.

1) Lindenmayer System: In 1968 a biologist, Aristid Lindenmayer, introduced a new type of string-rewriting mechanism, subsequently termed as L-system. Originally L-system was devised to provide a formal description of the development of such simple multicellular organisms, and to illustrate the neighborhood relationships between plant cells. Along with the change in times and technological flourishing, this system was extended to describe higher plants and complex branching structures [11].

L-system, by writing the geometry of a plant into a grammar, can produce slow-growing effects by increasing the number of layers returned, and plants can gradually become more complex. L-system is defined as a triplet:

$$G = (V, \omega, P)$$

V is the alphabet of the system. ω (axiom) is an initial string. P is a set of production rules which define the symbols that can be replaced with other symbols [11].

The simplest L-system is the Deterministic and Context-free L-system (DOL-system for short). That is, for any symbol in the L-system alphabet, there is only one rewrite rule. It is identical for all the plants generated by the same DOL-system. An attempt to combine them in the same picture would produce a striking effect, they seem rather unnatural. Therefore, the L-system is further developed into a Stochastic L-system.

There are two cases being context-sensitive: 1L-system and 2L-system. Productions in 1L-system have one-sided context only, shown in the form $a_l < a \rightarrow \chi$ or $a > a_r \rightarrow \chi$. Letters a_l and a_r mean the left and the right context of a in this production. 2L-system uses productions of the form $a_l < a > a_r \rightarrow \chi$, where the letter a (called the strict predecessor) can produce word if and only if a is preceded by letter al and followed by ar. OL-system, 1L-system and 2L-system belong to a wider class of IL-system, also called (k, l)-system. In a (k, l)-system, the left context is a word of length k and the right context is a word of length l [11]. The difference between L-system and general formal language is that formal language can only apply one grammar rule in each iteration, while L-system can apply many different rules at the same time. For example, Lindenmayer proposed the earliest L-system (A→AB), (B→A). A and B can be replaced by other states. This is why the plants produced by the L-system are more authentic and complex than the formal language. The concept of Turtle Graphics is used when drawing plant by L-system. Imagine a turtle crawling on the beach, climbing the route will leave traces, and consider the turtles can only move forward, this is the Turtle Graphics. The state of the turtle is defined as (x, y, α). The turtle is in position (x, y). The turtle's head is facing the direction of the angle α.

In L-system, the more commonly used symbols are "F", "+", "-", "[", "]".

F Move forward a step of a branch length and draw a branch between points (x, y) and (x',y'). The state of the turtle changes to (x', y', α), where $x' = x + d cos\alpha, y' = y + d sin\alpha$.

+ Turn left by angle δ. The next state of the turtle is (x, y, $\alpha + \delta$). The positive orientation of angles is counterclockwise.

 Turn right by angle δ. The next state of the turtle is (x, y, $\alpha - \delta$).

[Push the current state of the turtle onto a stack. The state includes the turtle's position and orientation.

] Pop a state from the stack and make it the current state of the turtle.

For example, $F_1[+F_2]F_3, F_2$ is the branch of F_1F_3 shown in Figure 4. Through these simple symbols, we can more easily show the structure of the entire plant and its branching status.

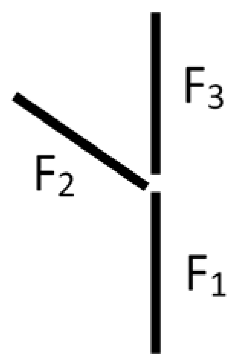

Fig. 4. Graphical result of $F_1[+F_2]F_3$

Many research result of L-system have been published [12, 13, 14, 15, 16].

B. Unity

Unity is a cross-platform game engine that can be used to create interactive content such as 2D/3D games, architectural visualizations, and 3D animations. Moreover, it enables the development of AR (Augmented Reality), VR (Virtual Reality), and MR (Mixed Reality) application with the help of development packages like Vuforia SDK, Arpa SDK, and Metaio SDK [17]

Unity has personal, plus, professional and enterprise versions available. Personal version is free for life, which created a vast community of Unity developers, as well as the resource-rich Asset Store. Many developers package their creations into packages and hit the Asset Store shelves. Packages contain models, animations, sound effects, particle systems, programs, shaders, and even complete games. Other developers can purchase these kits and change the content on their demand. It significantly reduces the game development time.

Developing iOS/MAC games requires Objective-C language. Developing Android games needs Java. Developing web games needs PHP and JavaScript. Developing PC games demands C#. Each platform requires a different IDE. Unity enables users to choose from C# or JavaScript languages to develop applications on PC, Mac, web, iOS, Android or other platforms. Because of this feature, Unity becomes one of the mainstream software for game production now. We chose Unity as the development engine because Unity has the following advantages over other game engines. It has a huge developer community. A large number of scripts, models, and packages are available for selection. We use Kinect v2 and HTC Vive in Unity via the Unity Pro suite and SteamVR SDK. Adding custom scripts to game program is quite intuitive and easy.

III. SYSTEM DESIGN AND IMPLEMENTATION

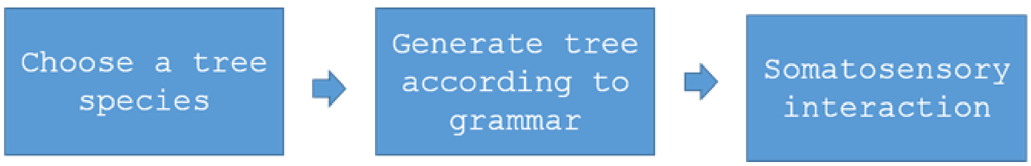

Fig. 5. System design

A. *Choose a Tree Species*

At present, only one tree species is provided for the user to choose. We wrote the grammar by referring to the characteristics of the Taiwan Golden-rain Tree. Taiwan Golden-rain Tree is up to 20 meters high, has a diameter of 20 to 45cm, trunk straight, tree bark smooth whose color is grayish black, branches erect or nearly erect, leafy to opposite or nearly opposite. Figure 6 shows the comparison between Taiwan Golden-rain Tree and our system simulated tree.

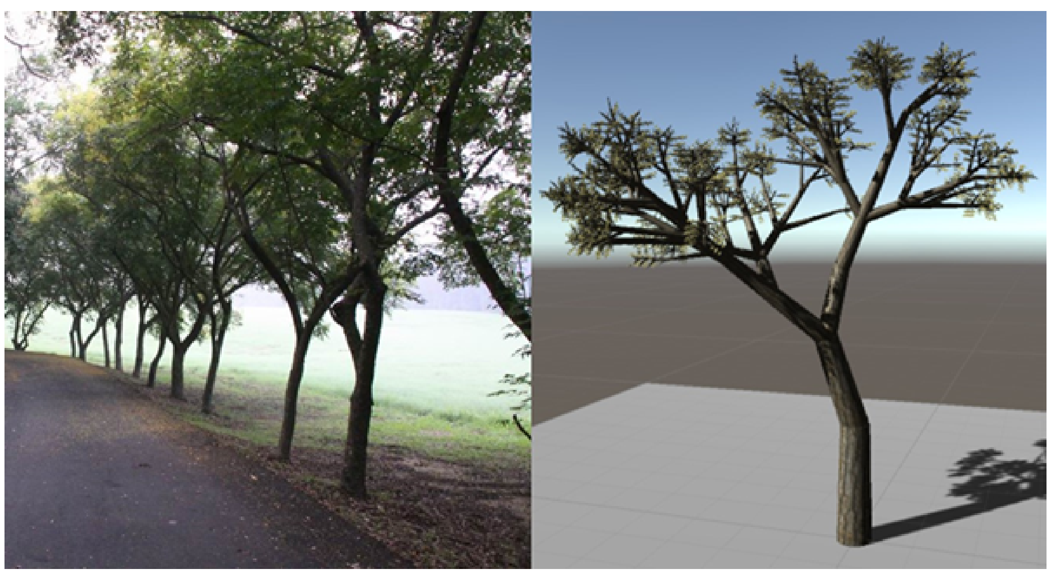

Fig. 6. Comparison between Taiwan Golden-rain Tree (left) and our system simulated tree (right)

B. *Generate Tree According to Grammar*

1) Tree height according to users height: Calculate the joints' coordinates from each other Head, Neck, SpineShoulder, SpineMid, and SpineBase. Add the distance from HipLeft, KneeLeft to AnkleLeft. The head joint coordinates are in the middle of the head, so it needed to add Head to Neck distance once again. Take multiple measurements to average. The user's height is used to determine the scale of the stem object. The higher the user is, the larger the scaling factor is.

2) Generated string by grammar: The initial string (ω) is according to the tree species which user selected. Different will make different grammar selections.

3) Build the tree and close the left/right branches rendering: Generate a tree on the screen according to the string given in the previous step. Turtle Graphics can be further developed into three-dimensional space drawing. We create two stacks of type Vector3, one is responsible for storing the turtle location, anther storage turtle orientation of head.

"[", "]" indicates that there is a branch of the trunk, the former records the current trunk state, the latter returns to the trunk state. For example, the current state of the turtle in the branch A is position (0, 2, 0), angle (30, 0, 0). If the next character is "[", this state is saved first. After that, no matter how many branches they produce, as the corresponding "]" appears, the turtle's position becomes (0, 2, 0) and orientation of head is reduced to (30, 0, 0). This means that it can go back to the main trunk after the branch is established.

"+", "-" indicate positive and negative random rotation of 30 40 degrees on Z axis. "&", "^" indicate positive and negative random rotation of 25 35 degrees on Y axis. "/", "*" indicate positive and negative random rotation of 30 40 degrees on X axis. "(Number)" means rotated by a specific angle, for example, "- (30)" is rotated by -30 degrees along the Z axis.

Level of stem represents the backbone and the branch of the class relations, shown as Figure 7.

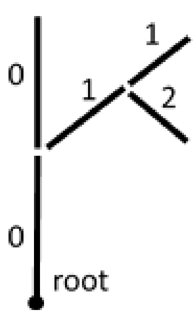

Fig. 7. Level of stem

"F" indicates growing branch, that is, a stem object is created on the screen. In general, new branches will be shorter and thinner than the older ones. The larger the level of stem is, the shorter the branch is. At the same level of stem, the farther away from the root of the branch is shorter. The stem object is a cylinder whose bottom is wider than the top. The diameter ratio is 1: 1.25. The new stem scale is the old one scale multiplied by 0.8 as shown in Figure 8. Stem object has collision detection. If branch A collides with other branches and A's level of stem is greater than 4, A would be deleted.

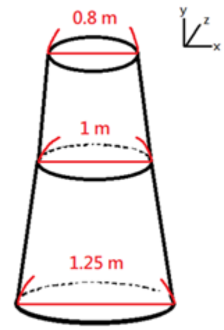

Fig. 8. Stem objects

In order to facilitate the growth of leaves, we record the location and direction of the stem objects which level of stem is greater than three.

C. Somatosensory Interaction

1) Motion detection: Flat hands: left elbow y coordinate = left wrist y coordinate = left shoulder y coordinate, and right elbow y coordinate = right wrist y coordinate = right shoulder y. The tolerance is plus and minus 3 mm.

Hold hands high: Left elbow x coordinate = left wrist x coordinate = left shoulder x coordinate, and right elbow x coordinate = right wrist x coordinate = right shoulder x. The tolerance is plus and minus 3 mm.

Waving hands: judgment based on the elbow y coordinate changes.

2) Open rendering of the branches: If it is detected that the user is waving his right hand, the branch in the right part is opened rendering and a parent-child relationship is established. The rendering of the right part branches is all turned on, indicating that the right part branches have finished growning.

3) The tree swing its branches as user did: Obtain users left/right shoulder and elbow joint coordinates, and calculate the angle of hand swing. Substitute the angle into the rotation coordinates of the left and right branches of the parent object.

D. Watching the Tree by HTC Vive (VR)

Import virtual reality technology, only using the kit - SteamVR Prefabs, to allow user to walk through the forest they create through HMD (Head-mounted display).

Growing trees in VR is our ongoing work. In addition to being able to observe the growing trees directly, it also gives users an experience of the scene ambiance.

IV. CURRENT RESULTS

a. After the user selects the tree species, hold both hands high. The system interface shown the main trunk of the tree, like Figure 9.

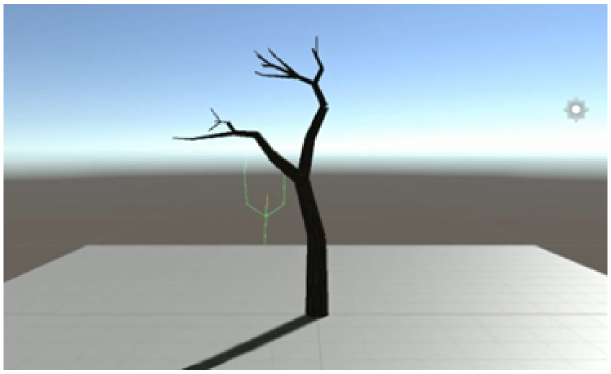

Fig. 9. System interface (grow main trunk)

b. When the user waves his left hand, the left branch starts to grow, as shown in Figure 10. The right hand is the same.

c. When waving his hands, both sides of the branches no longer grow. The branches have finished growing, as shown in Figure 11.

d. When the user lifts his hands and swings to the sides, the dead wood blossoms leaves. Figure 12 shows the appearance of the leaves.

e. If the user waves his right hand, the right part branches follow the swing, as shown in Figure 13. The left hand is the same.

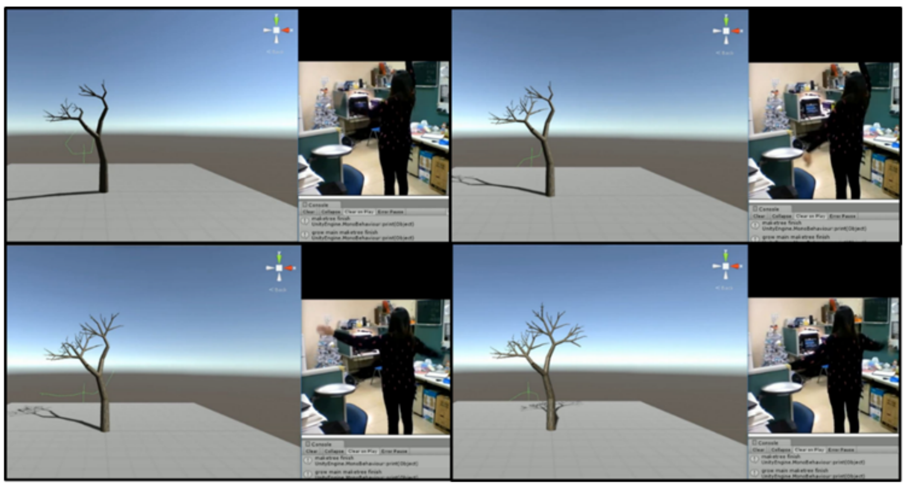

Fig. 10. System interface

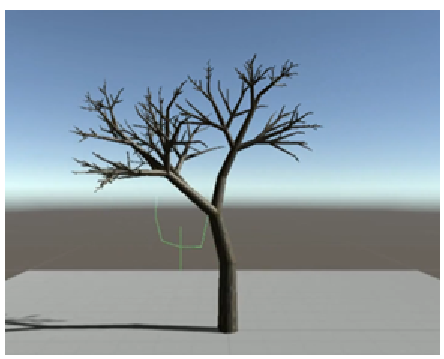

Fig. 11. System interface

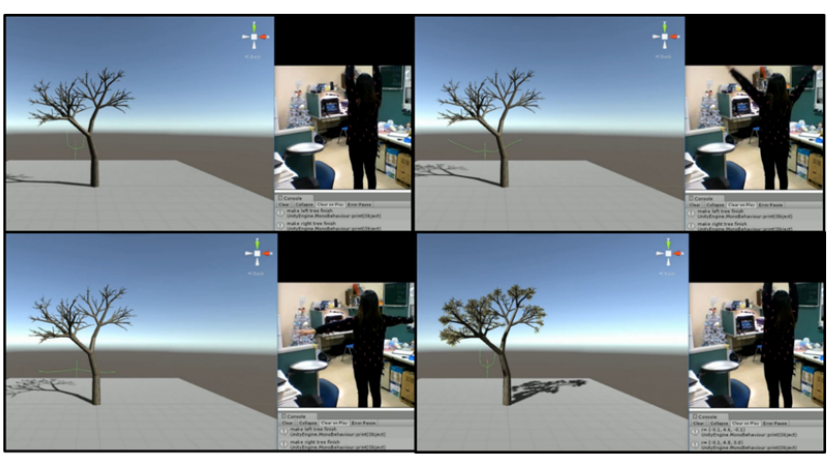

Fig. 12. System interface

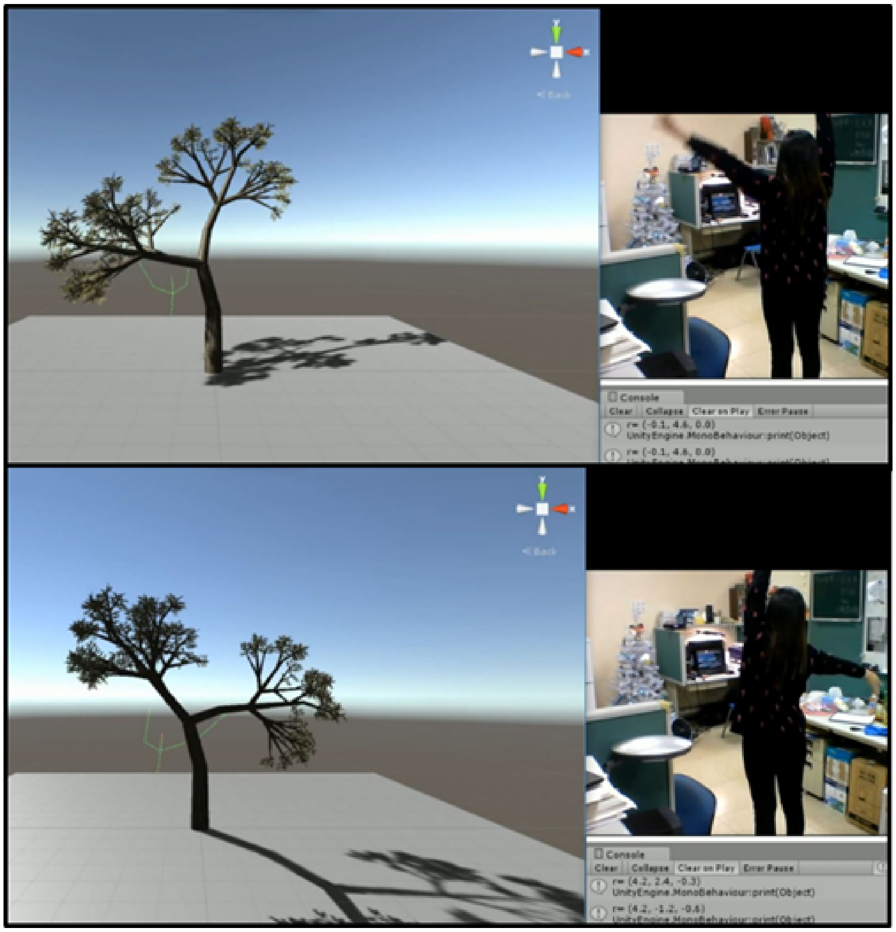

Fig. 13. System interface

V. CONCLUSION

In recent years, the somatosensory technology has been widely used in the game industry. In addition, somatosensory technology is used for the purpose of rehabilitation, too. Meanwhile the development and application of VR and AR have also become a hot trend.

In this paper, we integrated the motion capture technique of Kinect v2, the plant growth simulation of L-system, and Unity to implement a somatosensory Edutainment System to create interesting 3D trees animations without using complicated modeling techniques.

There is a large number of plant scenes in computer games, animations, movies or advertisements, but the existence of these plants is not the "protagonist", such as [18, 19]. Our system is Plant-based, allowing users to focus on plant growth.

Through the edutainment human-computer interactions, users can enjoy the fun of using their body motion to control the plants growth in real time, and reproduce the impressive experience in the movie scene of "My Neighbor Totoro". Our system can also be used for the purpose of elderly exercise at home. Furthermore, through the interactions with our system, we aim to promote the children's awareness of the natural environment.

REFERENCES

[1] H. Miyazaki, *Tonari no Totoro.* Tokyo, Japan: Tokuma Shoten, 1988.

[2] Z. Wu, M. Zhou, X. Wang, X. Ao, and R. Song, "An interactive system of modeling 3d trees with ball b-spline curves," in *Second International Symposium on Plant Growth Modeling and Applications,* Beijing, China, 2006.

[3] K. Xie, F. Yan, A. Sharf, O. Deussen, H. Huang, and B. Chen, "Tree modeling with real tree-parts examples," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2608–2618, 2016. doi: 10.1109/tvcg.2015.2513409

[4] Microsoft, "Kinect for windows sensor components and specifications," n.d. [Online]. Available: https://goo.gl/ygFgaQ

[5] Techbang, "The body is the controller. microsoft kinect how to do it," 2010. [Online]. Available: https://goo.gl/h1W6iz

[6] J. Sissing, N. N. Dlamini and K. Johnston, "Using m-commerce to achieve strategic objectives in south african retail organisations," *Journal of Applied and Physical Sciences*, vol. 3, no. 1, pp. 7–16, 2017. doi: 10.20474/japs-3.1.2

[7] Zouxy, "Kinect development study notes (a) kinect introduction and application," 2012. [Online]. Available: https://goo.gl/ysespr

[8] H. Jiawei, X. Zhiguo, H. Lingeng, X. Ying, L. Nianfeng, and S. Reika, "A special edutainment system based on somatosensory game," in *6th IEEE International Conference on Software Engineering and Service Science,* Beijing, China, 2015.

[9] S.-T. Chen, Y.-G. L. Huang, and I.-T. Chiang, "Using somatosensory video games to promote quality of life for the elderly with disabilities," in *Fourth International Conference on Digital Game and Intelligent Toy Enhanced Learning,* Takamatsu, Japan, 2012.

[10] P.-F. Wu, M.-J. Huang, and N.-W. Chang, "The learning experience of fine art by somatosensory game device," in *Fifth International Conference on Service Science and Innovation,* Paris, France, 2013.

[11] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. New York, NY: Springer Science & Business Media, 2012.

[12] K. J. Mock, "Wildwood: The evolution of l-system plants for virtual environments," in *IEEE International Conference on Evolutionary Computation Intelligence,* Nagoya, Japan, 1998.

[13] F. Valladares, S. Matesanz, F. Guilhaumon, M. B. Araújo, L. Balaguer, M. Benito-Garzón, W. Cornwell, E. Gianoli, M. van Kleunen, D. E. Naya *et al.*, "The effects of phenotypic plasticity and local adaptation on forecasts of species range shifts under climate change," *Ecology Letters*, vol. 17, no. 11, pp. 1351–1364, 2014. doi: 10.1111/ele.12348

[14] Y. Manabe, S. Kawata, and H. Usami, "A pse for a plant factory using l-system," in *7th International Conference on Computing and Convergence Technology,* Seoul, Korea, 2012.

[15] M. Xueqiang and L. Hong, "Research on an generalized l-system method," in *International Symposium on Information Technology in Medicine and Education,* Guangzhou, China, 2012.

[16] Y. Sun, Y. Liang, and Q. Wu, "Visualization of plant growth based on l-system," in *World Automation Congress,* Madrid, Spain, 2010.

[17] G. Kurubacak and H. Altinpulluk, *Mobile Technologies and Augmented Reality in Open Education*. Pennsylvania, PA: IGI Global, 2017.

[18] K. Guan, "The research of the improved 3d l-system and its application in plant modeling," in *IEEE Conference on Cybernetics and Intelligent Systems,* Chengdu, China, 2008.

[19] Wikipedia, "L-system," n.d. [Online]. Available: https://goo.gl/ArG6f4