

Article

Event-Centered Maze Generation Method for Mobile Virtual Reality Applications

Kisung Jeong and Jinmo Kim *

Department of Software, Catholic University of Pusan, Busan 46252, Korea; coco4612@naver.com

* Correspondence: jmkim11@cup.ac.kr; Tel.: +82-51-510-0645

Academic Editor: Angel Garrido

Received: 7 September 2016; Accepted: 31 October 2016; Published: 4 November 2016

Abstract: This study proposes a method of effectively creating mobile virtual reality scenes centered at events for the purpose of providing new experiences in virtual reality environment to users. For this purpose, this paper uses Prim's maze generation algorithm to automatically create maze environments that have different patterns every time and to compute mazes with finite paths. This paper designs a scheme of creating virtual reality scenes based on event-centered mazes to maximize users' tension and immersion. Here, event components that are appropriate for the maze environment are defined and maze patterns are created centered at the event point where events that are appropriate for the maze pattern are automatically created. Finally, the paper analyzes whether the proposed virtual reality scene based on event-centered mazes is helpful in enhancing users' immersion and arousing their interest through diverse experiments.

Keywords: mobile virtual reality; creation of virtual reality scene; event-centered maze, immersion; motion sickness

1. Introduction

Virtual reality is a technique that enables users to have a reality-like experience by stimulating their visual, tactile, and auditory senses in a virtual environment that is created by the computer. Following the recent advancement of computer graphics techniques and virtual reality hardware devices, various types of virtual reality equipment including HMD (Head Mounted Display), 360° camera, and leap motion began to disseminate to general users, which consequently preceded active progress of techniques and research in the relevant fields. Such interest in virtual reality is accelerated due to its application in diverse fields, such as games, movies, animation, education, and tourism. Due to the recent advancement of smart devices and low-priced mobile HMD, anyone can now learn and experience simple virtual reality content.

Studies related to virtual reality from the research on the HMD system in the 1960s, which is responsible for transferring visual information that has stereoscopic perception to users in virtual space. They were further developed into studies such as input systems that deliver physical responses generated in the interaction in the virtual world [1]. Afterwards, various studies followed such as the study that provides life-like experience in virtual reality by overcoming spatial restriction and the one that enhances realism of auditory experience. In particular, there are studies that attempt to effectively generate 3D scenes which constitute virtual worlds in a more intuitive structure. The background in virtual space provides a space where users in the virtual space experience motion as if in reality, and it plays a role in leading the situation. Moreover, lively terrain, allocation of props and performance of events can also increase immersion. For this purpose, a number of schemes have been studied and developed that compose virtual scenes in intuitive structures by using languages such as VRML (virtual reality modeling language) and X3D [2] or create realistic virtual space directly by using a city creation engine such as CityEngine [3]. However, virtual reality content on mobile platforms

have the potential problem of motion sickness due to delay that is caused by complicated rendering computations. Hence, the number of polygons is sometimes limited to under 500k. As such, mobile platforms have relative limitations in expressible space and scenes compared to PC platforms. A maze is one of the environments that can provide users with new and diverse experience by solving such systematic limitations. While having simpler structure than cities or landscapes, mazes have distinctive characteristics that can enhance users' concentration and immersion. However, if mazes are created in an identical form every time, users will be bored even though other events have a decent structure.

This study proposes a scheme of automatically creating virtual reality scenes every time in mobile platform environments that can enhance users' immersion by using maze generation algorithms. This study defines virtual reality events that are appropriate for the maze environment and designs a scheme for automatic updating of maze scenes centered at the events. The proposed event-centered maze generation scheme has the following two contributions.

- It proposes maze scene generation scheme for the first time that is appropriate for virtual reality environments through 3D prop and maze generation algorithm.
- It designs events that are appropriate for maze environments in virtual reality and provides a novel environment where users can focus on the maze centered at the events.

Chapter 2 describes research related to the creation of 3D virtual scenes and maze generation algorithms based on virtual reality techniques. Chapter 3 systematically explains the implementation principles of the maze generation algorithm proposed in this study. Chapter 4 describes a scheme of creating event-centered maze scenes that enhances immersion of the virtual reality environment based on maze generation algorithms. Chapter 5 analyzes the impact of the proposed event-centered maze-based virtual scene creation on users' immersion through diverse settings of the immersion environment. Chapter 6 presents conclusions and future research.

2. Related Works

Studies related to virtual reality stem from the research on HMD systems that Sutherland [1] suggested for the purpose of transferring stereoscopic visual information that is appropriate for the virtual reality environment. Afterwards, studies involving tactile sense system, motion platform, and flight simulation followed aiming at the delivery of physical responses in virtual worlds to users [4,5]. For the purpose of overcoming the restrictions of limited space depending on HMD, CAVE (Cave automatic virtual environment) has been studied, which enhances immersion by manufacturing a cube space of a certain size and projecting image information that corresponds to the previously computed users' sight on the wall by a projector [6]. Through this, users' behavior range broadened, consequently enhancing their immersion. There are also studies that enhance realism by satisfying users' auditory experience in virtual space through an application of stereo or surround audio [7].

As for the studies regarding users' tactile sense or physical responses, there were studies that let users directly touch and control virtual objects through data gloves. Some studies implemented liberal movement without spatial restriction by analyzing users' joint information or recognizing movement by using motion capture devices [8,9]. Laycock et al. [10] conducted research on the improvement of the realism of interaction in the virtual reality through virtual object manipulation and motion simulation. Danieau et al. [11] suggested HapSeat for the simulation of actuating users' head and hand at lower cost. As the spatial restriction is resolved, physical props became an important factor for the recreation of virtual space. From actions such as grabbing a door knob to opening the door or pushing the walls to various physical objects including stairs, switches, wind, and ladders, studies involving interaction with objects are recently under active progress [9].

As for the means of effectively creating immersive virtual space, methods that use web-based languages such as VRML and WebGL were mainly studied in the 1990s. Clay and Wilhelms [12] suggested a scheme of interactively creating virtual objects by using graphic domains of the system

and language. Zeng and Tan [13] devised language interface for 3D scene creation and further developed it in a visually semantic way [14]. Afterwards, engines such as Unity 3D [15] and Unreal [16] emerged following the advancement of computer graphics technique and hardware, enabling users to realistically and efficiently create virtual scenes in intuitive structures. As they support multi-platforms, they provide environments where mobile virtual reality content can also be effectively generated. However, the structure of mazes have a certain pattern different from cities or landscapes. Hence, there are limitations when users attempt to manually create them.

Maze generation algorithms first set sections by cell unit where mazes will be drawn and calculate paths and walls in a recursive structure. Depth first search is one of the most fundamental types of research regarding the creation of mazes. Afterwards, Prim's algorithm, Kruskal's algorithm, and the growing tree algorithm followed. Prim's algorithm [17] creates walls by determining which cell will be generated as mazes and which cells are the neighboring cells based on minimum spanning tree. Kruskal's algorithm generates mazes by listing each path or wall among randomly selected maze cells and allocating storage space in proportion with the maze size to circulate the ordered path [18]. Growing tree algorithm [19] is capable of generating mazes that have different textures. However, there is no research that applies maze generation algorithms to immersive virtual reality content. Hence, this study proposes a scheme of creating event-centered maze-based virtual reality scenes by using maze generation algorithms that can enhance users' immersion.

3. Maze Generation Algorithm

The most important factor in the creation of virtual reality scenes in maze environments is to eliminate a situation where users fall into infinite paths. When users fall into infinite paths, motion sickness or dizziness can occur. Hence, it is necessary to come up with an algorithm for generating finite mazes. Finite mazes require consideration of the following two key factors. The first is to generate mazes in non-circular structures where only one path can exist from the entrance to the exit. The second is to make sure that blocked walls where users cannot enter are not generated during the process of maze wall generation.

This study uses Prim's algorithm as a means of generating the event-centered finite maze, which is one of the minimum spanning tree algorithms. Here, the minimum spanning tree means a spanning tree that has the lowest cost out of all spanning trees that were created from a random weight graph. When given with a graph as in Figure 1, trees are calculated from the starting nodes by selecting one of the neighboring nodes that has paths with the lowest costs. When this process is repeated until the tree has $n - 1$ number of paths, the minimum spanning tree is completed. Here, yellow nodes refer to initial status, red nodes mean starting nodes, and green nodes and green paths are the minimum cost values that are selected in the current step.

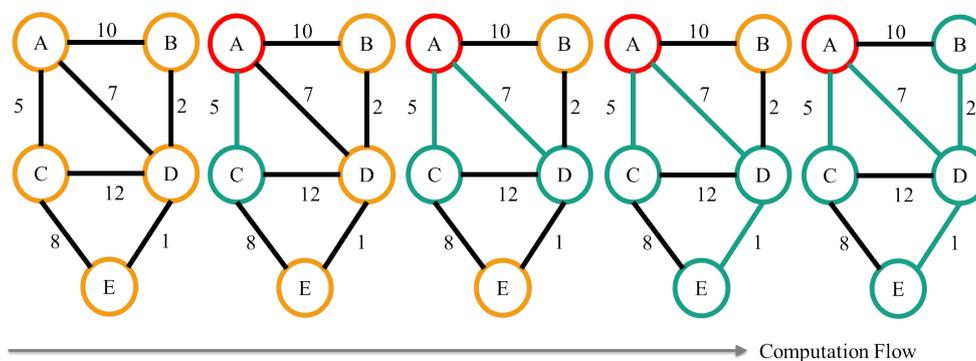


Figure 1. Computation process of minimum spanning tree.

Applying Prim's algorithm to mazes based on the computation of minimum spanning trees, we can generate finite mazes that have the size that users want. Figure 2 shows the procedure where paths and walls that compose the maze are generated step by step. First, mazes with users' desired size is set by cell unit. Under the assumption that every cell can be a wall, coordinates are stored in cell arrays. Next, starting cells are selected where paths will be generated as in Figure 2a. Mazes are being generated in a way where paths are procedurally generated from the starting cell and walls are built along the path. As shown in Figure 2b, two directions out of four directions surrounding the starting cell are randomly selected to determine the next path. Once the path is determined, walls are created in four directions around the path as in Figure 2c. Here, cells that are already determined as paths or those out of maze areas cannot become walls. As in Figure 2d, yellow cells means walls and green cells means paths that have been generated so far. New paths in the next step is generated by selecting one cell out of the generated walls and penetrating this wall. Hence, walls are generated by first selecting one cell (purple) out of wall cells as in Figure 2e and then selecting the next cell in a forward direction (Figure 2f). Repeating this process can generate maze patterns as shown in Figure 2g.

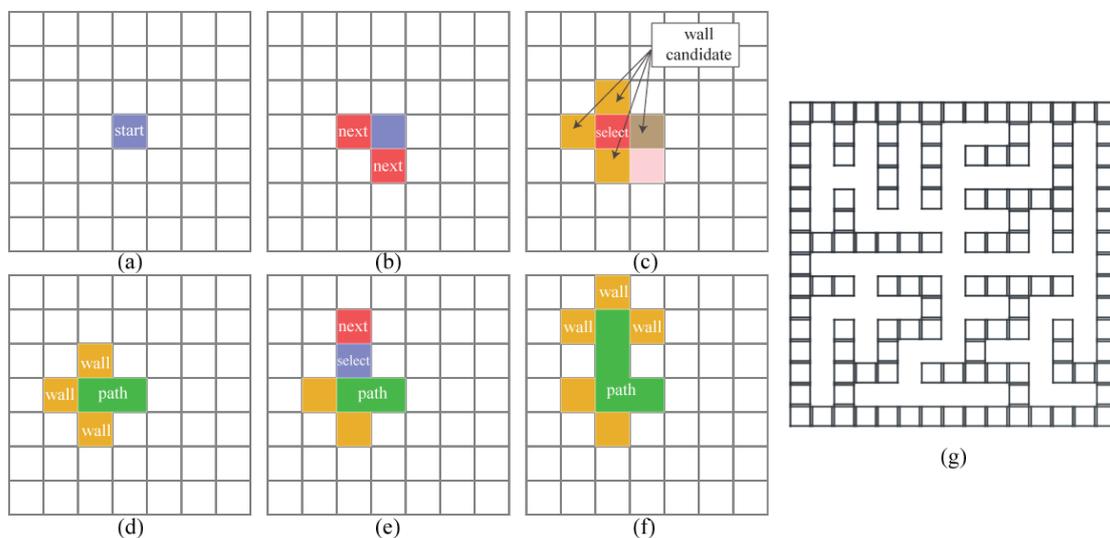


Figure 2. Process of maze generation based on Prim's algorithm: (a) selection of starting cell; (b) determination of the next path from the starting cell; (c) creation of walls around the paths; (d) separation of paths and walls; (e) generation of a new path through selection of one cell with walls; (f) expansion of paths and walls; and (g) generation of maze patterns.

4. Virtual Reality Scene Based on an Event-Centered Maze

Instead of simply generating maze maps only, this study attempts to create virtual reality scenes that can enhance users' immersion centered at events that are appropriate for maze environments. Hence, we design events that can arouse users' tension and responses in the maze environments and generate mazes centered at these events. We implement input process techniques using game pad controllers so that users can freely move around the maze in the mobile environment. Figure 3 summarizes the creation of virtual reality scenes based on event-centered mazes. This study uses the Unity 3D engine to effectively produce virtual reality content in maze environments under intuitive structures. When using the Unity 3D engine, building integrative environments with the Google VR development tool can be efficient. Hence, it is effective for achieving the purpose of this study, which is to create virtual reality scenes on mobile platforms.

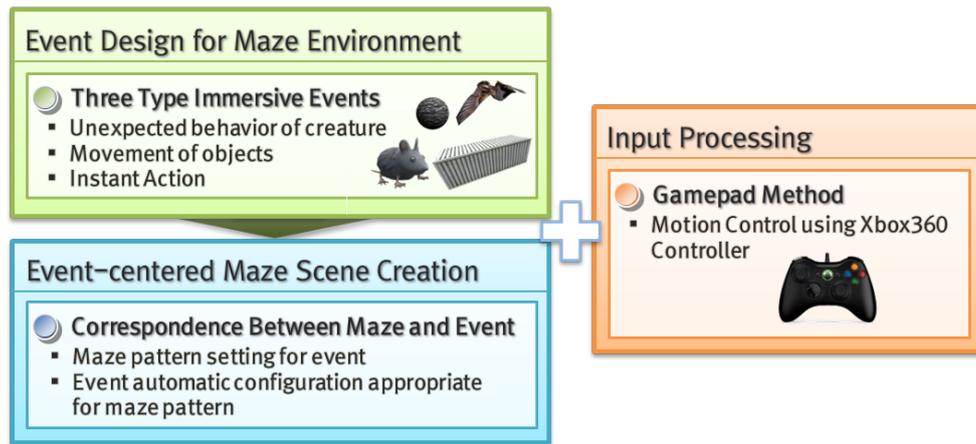


Figure 3. Overview of virtual reality systems based on an event-centered maze.

4.1. Design of Events Appropriate for Maze Environment

Before designing an event-centered maze, event factors that can enhance users' immersion and tension as well as are appropriate for the maze environment were formed by dividing into the following three types as shown in Figure 4: unexpected behavior of living creatures such as animals, instant action of containers, etc., and movement of objects such as rocks.

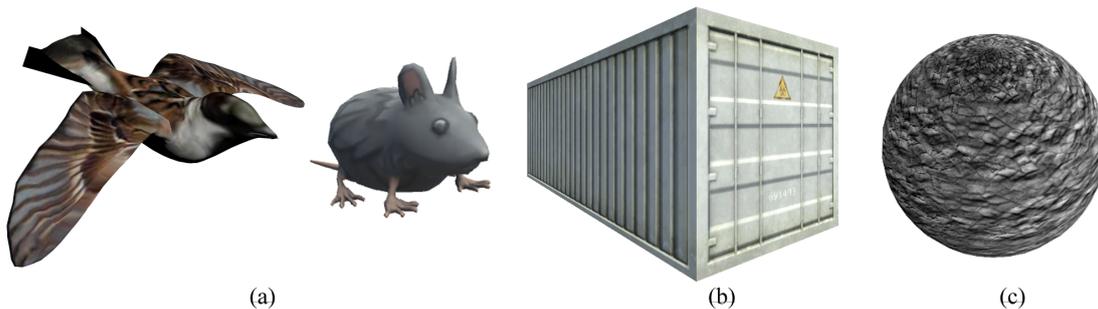


Figure 4. The proposed event components: (a) living creatures; (b) containers; and (c) moving objects.

When users encounter objects such as rocks, special effects are added as if glass is broken in front of the users' eyes to reinforce the immersion. Moreover, letting the users hear the corresponding sounds for the event can maximize tension in the maze environment.

4.2. Creation of Event-Centered Maze Space

The purpose of creating virtual reality scenes in maze environments by using maze generation algorithms is to provide users with new maze scenes every time. Mazes in identical patterns can rapidly decrease the interest of content as the escape paths are exposed during the repeated use of the contents. Similarly, event location should be set according to the newly generated maze patterns. Users should be able to pass the path where events are set and events that are appropriate for the maze pattern around the event should be given. Hence, we design a scheme of creating event-centered maze space by considering the three points above.

The first is a scheme of dynamically creating event locations that fit the maze pattern. When mazes are generated in sizes that users determine, we are unaware of which one of the walls or paths will be created at a point that we attempt to develop events since maze patterns differ every time. Hence, cell points where events will occur should be excluded from the wall creation condition so that they will be unconditionally become paths instead of candidates for walls that were set in

Figure 2c. When event points are randomly determined in maze cell space that has a designated size as in Figure 5a, walls do not develop in the appointed locations, as shown in Figure 5b.

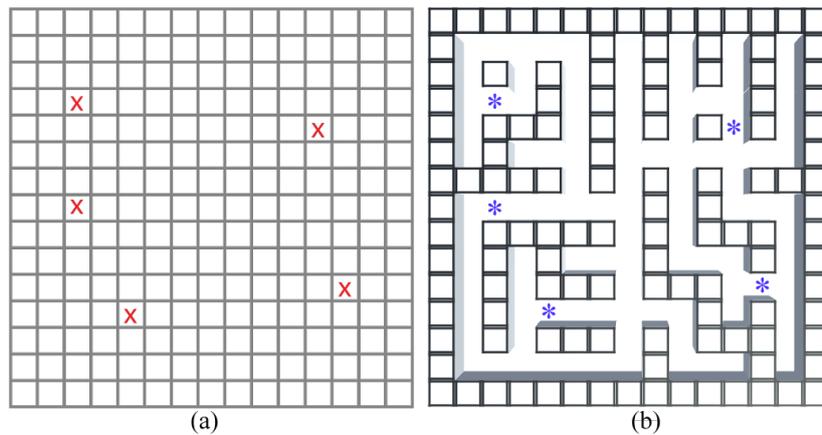


Figure 5. Maze pattern generation that considers events: (a) automatic determination of event points in maze maps; and (b) generation of event-centered paths.

Here, one problem is that walls can be possibly generated in four directions around the event cell, although walls do not occur in event cells. These kinds of results can lose meaning as users cannot reach the event cell. Hence, once an event-centered maze is created as shown in Figure 5, paths should be created in up-and-down or left-and-right directions through four-direction tests around the event cell. This is presented in Figure 6. Furthermore, Algorithm 1 outlines the automatic creation method of the designed event-centered maze space considering these problems.

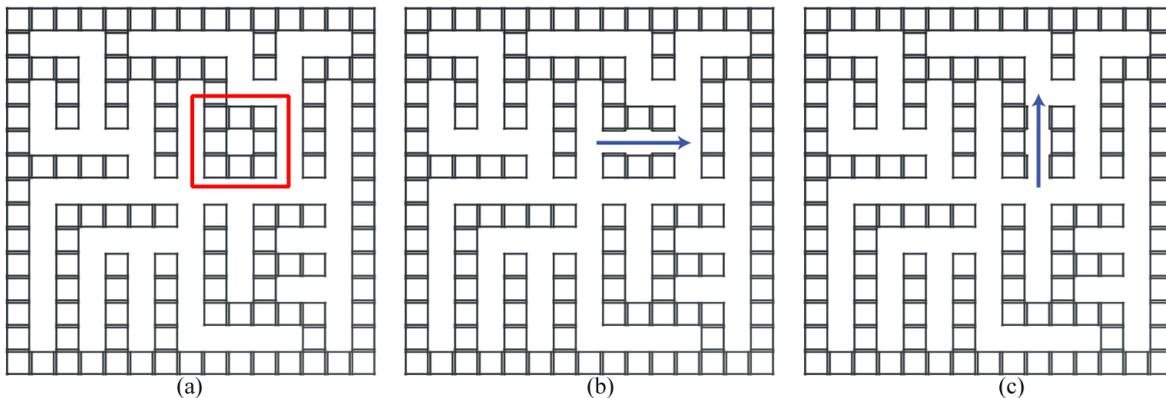


Figure 6. Path generation scheme for solving the problems of the event-centered maze pattern: (a) problem of maze patterns; (b) generation of left-and-right direction paths; and (c) generation of up-and-down direction paths.

Algorithm 1 Creation of event-centered maze space.

```

size[2] ← size of the maze pattern (width by height).
Array_EventsPt ← array for storing the event cell position.
procedure AUTOMATIC GENERATION OF EVENT-CENTERED MAZE PATTERN(size, Array_EventsPt)
   $n_{events} \leftarrow$  specified number of events (Array_EventsPt.size).
  for  $i = 0, n_{events} - 1$  do
    the event cell position (Array_EventsPt[i].x, Array_EventsPt[i].y) is specified as the path.
  end for
  generate a maze map of size[0] by size[1] using Prim's algorithm.
  for  $i = 0, n_{events} - 1$  do
    test cells in four directions around the event cell position (Array_EventsPt[i].x,
Array_EventsPt[i].y).
     $n_{walls} \leftarrow$  store the number of walls after checking whether the cells in the surrounding four
directions are walls or paths.
    if  $n_{walls} = 4$  then
      if  $n_{random} \text{ (random number)} \% 2 = 0$  then
        create paths in left-and-right directions from the event cell position.
      else
        create paths in up-and-down directions from the event cell position.
      end if
    end if
  end for
end procedure

```

Once the event-centered maze pattern is automatically generated, one event out of the three defined event factors is automatically given to the event cell. Here, event setting that fits the maze pattern is necessary because of the difference of characteristics of each event component. Living creatures such as animals and insects should be able to unpredictably dash toward users from anywhere. In the case of objects such as rocks, a road over a certain distance in a single path should be prepared so that users can actually feel as if the rock is rolling toward them, which enhances tension. In the case of an action where a container suddenly appears, a single path is necessary. Only then can users encounter the event without missing. Here, short distance do not matter. Events that fit the maze pattern should be set by considering all of these processes. Algorithm 2 describes the setting.

Algorithm 2 Automatic setting of events appropriate for the maze pattern.

```

1:  $p_{event} \leftarrow$  event cell location.
2: procedure EVENT SETTING BASED ON MAZE ANALYSIS( $p_{event}$ )
3:    $b_{sp} \leftarrow$  determination of a single path by testing four directions around  $p_{event}$ .
4:   if  $b_{sp} = \text{false}$  then
5:     event settings of unexpected actions by living creatures at  $p_{event}$ .
6:   else
7:      $l_{path} \leftarrow$  computation of paths by testing five cells in up-and-down or left-and-right directions
from  $p_{event}$ .
8:     if  $l_{path} \geq 5$  then
9:       event settings of object movement at  $p_{event}$ .
10:    else
11:      event settings of instant action at  $p_{event}$ .
12:    end if
13:  end if
14: end procedure

```

Through this process, users will not miss events during their escape from the maze and they can experience events that match the maze pattern, which ultimately increases their immersion.

4.3. Input Processing Technique

In this study, mobile HMD was used for transferring visual information that has stereoscopic perception when creating maze-based virtual reality scenes on mobile platforms. In the case of mobile platforms, the mobile device is attached to an HMD for operation, and, as a result, there are restrictions in the input process. Hence, we used a game pad controller, which is generally used for gaze or console games. This study uses simple but accurate and convenient game pad controllers as an input process technique for the provision of liberal movement of users in maze environments by relieving the spatial restriction. When using the input process technique, it is important to match the controller keys with motions for maze escape. Hence, the input process was implemented by matching the input property of the Unity 3D engine with the controller (Figure 7).



Figure 7. Key setting process of the game pad controller.

5. Experimental Results and Analysis

We used Unity 3D (Unity 5.3.4f1, Unity Technologies, San Francisco, CA, USA, 2016), and the Google VR development tool (gvr-unity-sdk, Google, Mountain View, CA, USA, 2016) for the creation of maze-based virtual reality scenes and implementation of virtual reality techniques. The PC environment used in the experiment has built-in Intel® core™ i5-4690, 8 GB RAM, Geforce GTX 960 GPU. Here, an Xbox360 Controller (Microsoft, Redmond, WA, USA, 2010) was used for the input process.

The experiments consist of a process that checks whether the virtual reality scene is effectively created centered at events by using the proposed scheme and analyzes whether the event-centered maze environment helps enhance users' immersion. First, we check the results of the proposed virtual reality scene based on an event-centered maze. In Figure 8, a maze scene is created by using a wall prop centered at the event creation point in the virtual reality environment on a mobile platform. Users starts at the center of the maze and automatically escape it if they arrive at one of the four end points

in the upper left, upper right, lower left, and lower right corner. Users experience events that are set to happen during the escape process from the maze through game pad controllers and head tracking.

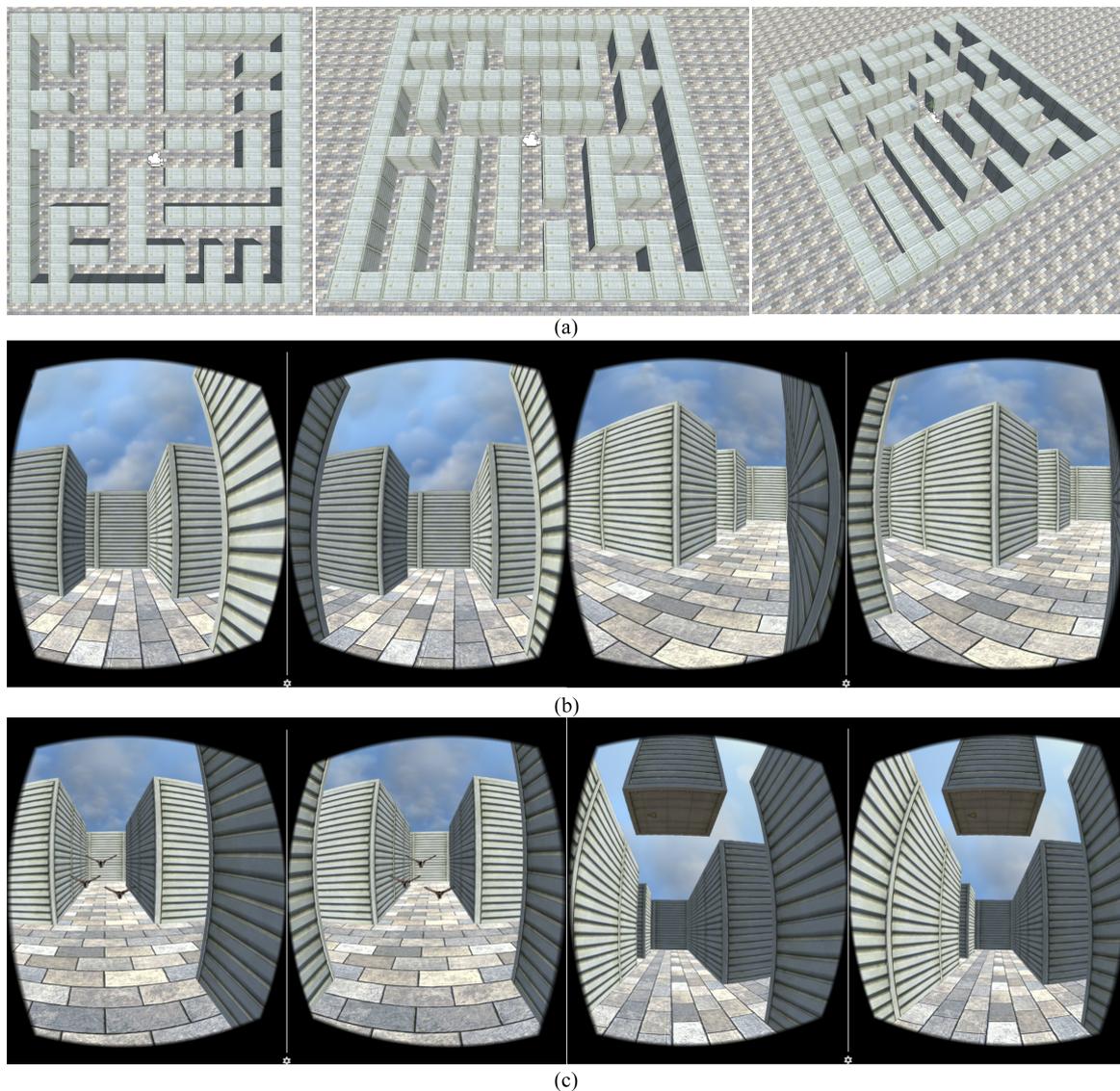


Figure 8. Virtual reality scene created by using the proposed event-centered maze: (a) automatic generation of maze map; (b) virtual reality scene of maze; and (c) automatically created events on the maze map.

Here, it is particularly important to carefully consider the refresh rate and fps in the case of virtual reality content on a mobile platform. The proposed virtual reality scene has a total of 14k size and fps does not exceed 75 during the contents process. This makes sure that users will not experience delay due to hardware problems.

Figure 9 describes specific progress of the maze scene that was created centered at events. As users explore the maze and proceed toward the exit, the event takes place when they arrive at a certain distance from the event occurrence point. This study implemented events by dividing into three types in a structure appropriate for a maze environment.

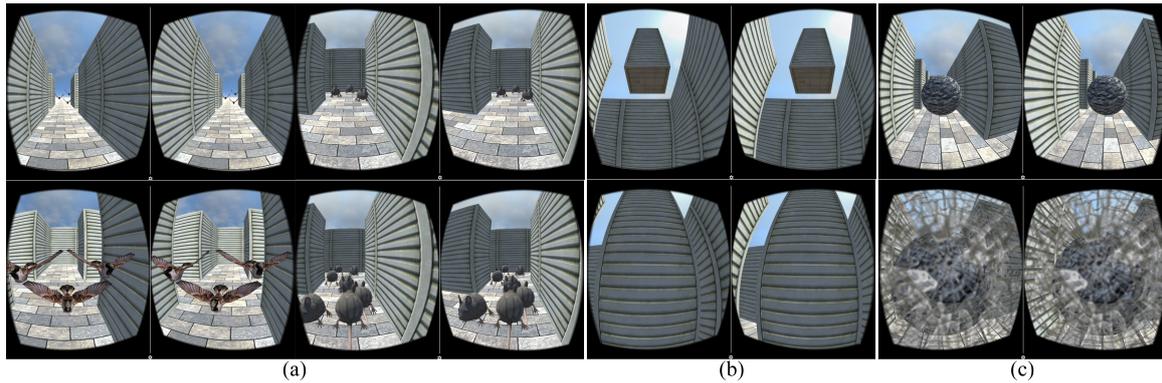


Figure 9. Three types of proposed events and their implementation results: (a) unpredictable action by living creatures; (b) instant actions; and (c) moving objects.

One of the most important purposes of the proposed scheme of creating maze-based virtual reality scenes is to automatically generate mazes with various patterns every time. Once the users define desired maze size and the number of events, event location is randomly determined where the pattern is automatically generated according to the maze size. In Figure 10, we can see that various maze patterns are automatically generated according to the size defined by the users. In addition, the time for automatically generating a maze pattern using the proposed algorithm according to the maze size was measured and verified. The results are outlined in Table 1. The average maze map generation time was 74.6 ms for the 15×15 maze size in Figure 10a and 136 ms for the 23×23 maze size in Figure 10b. These generation times were reasonable for using the proposed algorithm for virtual reality content of mobile platforms, which requires consideration of capacity and operation time.

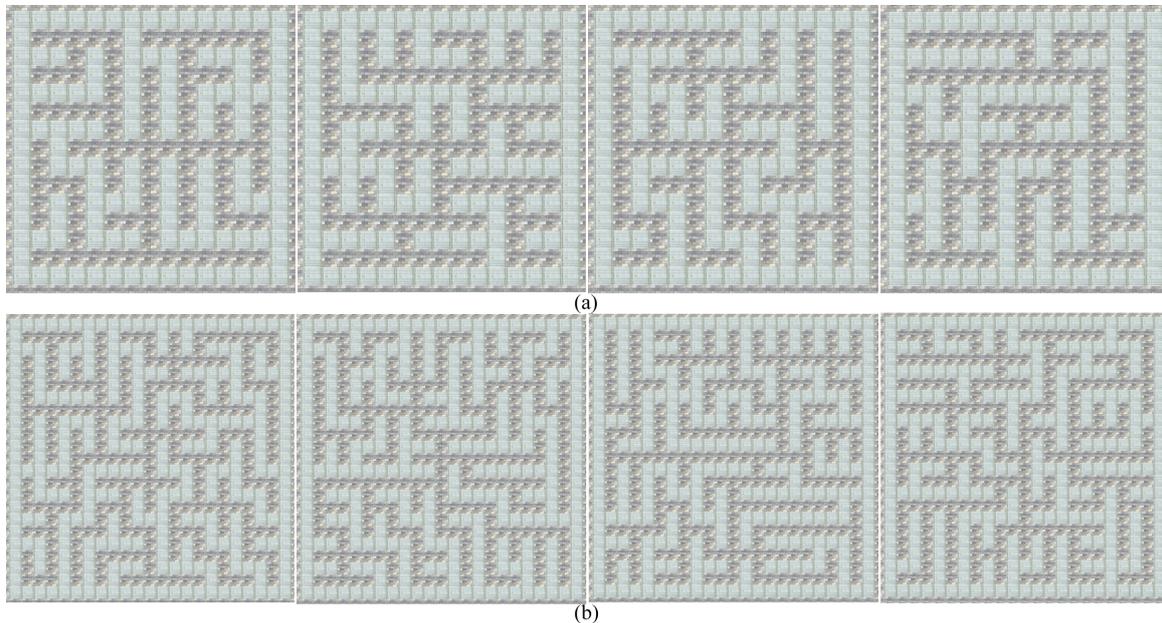


Figure 10. Results of generating various maze patterns according to the maze size: (a) 15×15 maze; and (b) 23×23 maze.

Table 1. Comparison of maze map generation time by maze size.

Maze Size	Average Generation Time
15 × 15 (Figure 10a)	74.6 ms
23 × 23 (Figure 10b)	136.0 ms

Next, we conduct an experiment on the impact of maze-based virtual reality scenes on the actual immersion of users and analyze the results. The experiment progressed with a survey by dividing positive factors of the maze-type virtual space from the aspect of immersion and negative factors from the aspect of motion sickness due to dizziness. A total of ten participants between the ages of 21 and 30 were randomly selected for the survey. Figure 11 shows the experience environment that was composed for the experiment of immersion into the maze in the implemented content. In particular, setting dark light as in Figure 11c is helpful to maximize the tension in the maze environment and immersion into the proposed events. Hence, we composed scenes of the creation results and conducted experiments by inducing users' sight with spot light only in the dark background overall as shown in Figure 8.



Figure 11. Experience environment of the contents using the proposed maze space: (a) mobile HMD (Head Mounted Display) used in the experiment; (b) operation outcome of the mobile virtual reality contents; (c) actual maze scene used in the experiment; and (d) users' experience.

Participants in the experiment responded to the survey in the following two question domains: the appropriateness of the maze environment for the immersive virtual reality content and the impact of the proposed event factors on immersion. Users responded to all questions with a score between 1 and 5. Figure 12 shows the experiment results of the appropriateness of the proposed maze-type virtual reality scene as immersive virtual reality content. All the participants replied positively, where the satisfaction score regarding immersion was as high as 4.4 on average. Hence, we could confirm that virtual reality content that uses mazes can provide new directions for enhancing immersion.

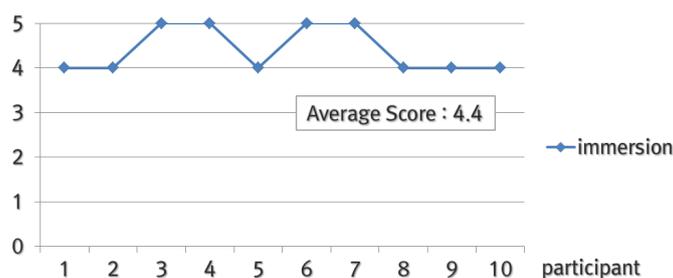


Figure 12. Analysis results of immersion of the maze-type virtual reality scene.

The final one is a comparison experiment of immersion of the events proposed in this study. As this study creates mazes centered at events rather than using simple maze patterns, the existence of events is a critical factor. Figure 13 presents the results of the comparison experiment. Comparing the immersion with or without events, the immersion score was 2.4 without events and 4.2 with event application. This implies that applying events is very helpful in increasing immersion. Here, the score is a relative value that compares the immersion according to the existence of events.

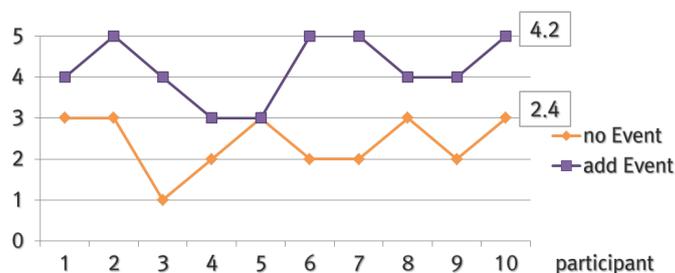


Figure 13. Experiment results of comparing the immersion of the proposed event-centered maze.

We could see that the virtual reality scene using an event-centered maze proposed in this study is appropriate for the users' experience of content with strong immersion. Here, we believe that allocating event factors that users can properly respond to will maximize the immersion of the maze environment, compared to the case of the virtual space that is composed of only mazes. If one can make diverse plans of events that match the maze environment according to the goals of the contents based on this, new types of mobile virtual reality content can be produced.

6. Conclusions

This study proposed an event-centered maze generation scheme for creating new virtual reality scenes in mobile environments. Among the maze generation algorithms with finite paths, we analyzed Prim's algorithm to create a 3D virtual reality maze environment. In addition to this, we designed a scheme of creating virtual reality scenes based on event-centered mazes that can maximize users' immersion when they experience the maze environment. We first designed event factors appropriate for the maze environment by dividing them into three and assigned the locations where the events would be created. Then, the maze was made to be automatically created according to the event. Here, problems that can occur involving the event setting location were resolved and wall patterns around the event cell location were tested to automatically allocate the appropriate events. Through this process, we made sure that users necessarily pass the event location during escape, increasing the immersion by responding to the event. The study conducted a survey to check whether the proposed maze-type virtual reality scene can provide users with immersive experience. In survey results, a large number of users replied that virtual reality content using mazes provided immersion. Moreover, the study conducted a survey asking whether the allocation of events appropriate for the

proposed maze environment was helpful in inducing tension and immersion of the users in their escape process. The survey results also indicated higher immersion of the event-centered mazes compared to ordinary mazes.

This study created virtual reality scenes by applying maze generation algorithm to 3D props. The outcome has limitations in that relatively simple scenes created as props were simplified due to the restrictions of the mobile platform. We plan to conduct research on the creation of realistic maze terrain landscapes in the future by expanding to PC platforms and matching the maze generation algorithms with 3D terrain. Further research is planned on the method of instant generation of a maze pattern that is appropriate for the present position in line with the situation, rather than instant generation of a perfectly complex maze, as a method of improving performance with the limited hardware resources of the mobile environment. While this study used game pads as the input process technique, which was mainly used for mobile virtual reality content, we plan to incorporate input process techniques such as leap motion in the future aiming at contents controlled with strong immersion.

Supplementary Materials: The following are available online at www.mdpi.com/link; Video S1: Event-Centered Maze Generation Method for Mobile Virtual Reality Applications.

Acknowledgments: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) and the Korea Foundation for the Advancement of Science & Creativity (KOFAC) grant funded by the Ministry of Education (No. NRF-2014R1A1A2055834).

Author Contributions: Kisung Jeong and Jinmo Kim conceived and designed the experiments; Kisung Jeong performed the experiments; Kisung Jeong and Jinmo Kim analyzed the data; Jinmo Kim wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutherland, I.E. A head-mounted three dimensional display. In Proceedings of the Fall Joint Computer Conference, Part I (AFIPS'68 (Fall, part I)), San Francisco, CA, USA, 9–11 December 1968; ACM: New York, NY, USA, 1968; pp. 757–764.
2. Pendlebury, M. 3D Virtual Reality Reconstruction on the Internet Using VRML. Master's Thesis, Manchester Metropolitan University, Manchester, UK, April 1996.
3. Esri-CityEngine. Available online: <http://www.esri.com/software/cityengine/> (accessed on 6 July 2016).
4. Stewart, D. A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **1966**, *180*, 371–386.
5. Jay, C.; Glencross, M.; Hubbard, R. Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *ACM Trans. Comput. Hum. Interact.* **2007**, *14*, doi:10.1145/1275511.1275514.
6. Cruz-Neira, C.; Sandin, D.J.; DeFanti, T.A.; Kenyon, R.V.; Hart, J.C. The CAVE: Audio visual experience automatic virtual environment. *Commun. ACM* **1992**, *35*, 64–72.
7. Cruz-Neira, C.; Sandin, D.J.; DeFanti, T.A. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93), Anaheim, CA, USA, 2–6 August 1993; ACM Press: New York, NY, USA, 1993; pp. 135–142.
8. Ortega, M.; Coquillart, S. Prop-based haptic interaction with co-location and immersion: An automotive application. In Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, ON, Canada, 1–2 October 2005; p. 6.
9. Cheng, L.P.; Roumen, T.; Rantzsch, H.; Köhler, S.; Schmidt, P.; Kovacs, R.; Jasper, J.; Kemper, J.; Baudisch, P. TurkDeck: Physical virtual reality based on people. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15), Charlotte, NC, USA, 8–11 November 2015; ACM Press: New York, NY, USA, 2015; pp. 417–426.
10. Laycock, S.D.; Day, A.M. Recent developments and applications of haptic devices. *Comput. Graph. Forum* **2003**, *22*, 117–132.
11. Danieau, F.; Fleureau, J.; Guillotel, P.; Mollet, N.; Christie, M.; Lécuyer, A. HapSeat: Producing motion sensation with multiple force-feedback embedded in a seat. In Proceedings of the 2012 ACM Symposium on VRST, Toronto, ON, Canada, 10–12 December 2012; ACM Press: New York, NY, USA, 2012.

12. Clay, S.R.; Wilhelms, J. Put: Language-based interactive manipulation of objects. *IEEE Comput. Graph. Appl.* **1996**, *16*, 31–39.
13. Zeng, X.; Tan, M. The Development of a Language Interface for 3D Scene Generation. In Proceedings of the Second IASTED International Conference on Human Computer Interaction (IASTED-HCI '07), Chamonix, France, 14–16 March 2007; ACTA Press: Anaheim, CA, USA, 2007; pp. 136–141.
14. Zeng, X. Visual semantic approach for virtual scene generation. In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI '11), Hong Kong, China, 11–12 December 2011; ACM Press: New York, NY, USA, 2011; pp. 553–556.
15. Unity3D. Available online: <http://www.unity3d.com/> (accessed on 15 March 2016).
16. Unreal. Available online: <https://www.unrealengine.com/> (accessed on 1 September 2016).
17. Prim, R.C. Shortest connection networks and some generalizations. *Bell Syst. Tech. J.* **1957**, *36*, 1389–1401.
18. Kruskal, J.B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **1956**, *7*, 48–50.
19. Growing-Tree-Algorithm. Available online: <http://www.astrolog.org/labyrnth/algrithm.htm> (accessed on 20 November 2015).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).