

On the Exploration of Game-Based Learning in STEM Education with the Development and Application of Virtual Reality Course Modules

Dr. Lei Zhang, University of Maryland, Eastern Shore

Dr. Lei Zhang received his Ph.D. Degree in Electrical Engineering on 2011 from the University of Nevada, Las Vegas. Since 2012 he is working in the Department of Engineering and Aviation Sciences, University of Maryland Eastern Shore. His main research interests include image processing, autonomous system, optical SoC/NoC architecture, and on-chip optoelectronic device design.

Mr. Benjamin Earl Whiteley

Dr. Dinesh K. Sharma, University of Maryland, Eastern Shore

Prof. Ibibia K. Dabipi, University of Maryland, Eastern Shore

I. K. Dabipi has been in academia for approximately 26 years during which he worked for Bellcore, AT&T Bell Labs and Southern University. He was the Chair of the Electrical Engineering Department at Southern University from 1997 to 2001. He was Professor and Chair of the Engineering and Aviation Science Department at University of Maryland Eastern Shore from 2001 - 2006. He is currently a Professor in the Department of Engineering and Aviation Sciences at University of Maryland Eastern Shore. His research interests are in the areas of Computer Security and Network Management, Parallel Computing and Algorithms Development, Performance Evaluation of Computer Networks, Optimization of Transportation Networks, and Economic Analysis of Transportation Facilities and Human factors in Aviation Security. He is a member of ASEE, HKN, ACM and a senior member of IEEE

Dr. Willie L. Brown Jr, University of Maryland, Eastern Shore

Dr. Willie Brown is an Assistant Professor in the Department of Engineering and Aviation Sciences at the University of Maryland Eastern Shore in Princess Anne, Maryland. He earned his Ph.D. in Business Administration with a specialization in Homeland Security Leadership and Policy (Aviation Safety and Security) with a joint doctoral program from Northcentral University and Embry-Riddle Aeronautical Science; and a Master's degree in Software Engineering from Embry-Riddle Aeronautical University. He also received dual degrees with a Bachelor of Science, in Aviation Science, and a Bachelor of Science, in Computer Science. In addition, he is a Federal Aviation Administration (FAA) licensed private pilot and earned a Management Development Graduate Certification from Harvard University.

He has worked in various capacities for the Office of Naval Research (ONR), National Aeronautics and Space Administration (NASA) and the North Carolina Department of Transportation in the Division of Aviation. He serves as the Vice President of Administration of the Eastern North Carolina IEEE for Geoscience and Remote Sensing Society. He has authored and co-authored publications for both national and international peer-reviewed journals with conference proceedings involving aviation and engineering education.

Dr. Weiwei Zhu, University of Maryland, Eastern Shore

Biographical Sketch Department of Math & CS University of Maryland Eastern Shore Weiwei Zhu, PhD Princess Anne, MD 21853 Phone: (410) 621 – 2376 Email: wzhu@umes.edu

(a) Professional Preparation

Ph.D. in Applied Mathematics University of Missouri-St. Louis July 2009 M.S. in Computer Science University of Missouri-St. Louis May 2002 Ph.D. Course Work in Statistics Northern Illinois University August 1999 - May 2000 M.A. in Applied Mathematics Truman State University May 1999 B.A. in Electronics Engineering Harbin Uni. of Science & Technology, July 1989

(b) Appointments

Assistant Professor of Mathematics, University of Maryland Eastern Shore, 2011- present Visiting Assis Professor of Mathematics, University of Maryland Eastern Shore, 2010 – 2011 Visiting Assis Professor

of Mathematics, University of Missouri-St. Louis, 2009 – 2010 Graduate Teaching Assistant, Dept. of Math & CS, Uni. of Missouri St. Louis, 2000 – 2009 Graduate Teaching Assistant, Dept of Math, Northern Illinois University, 1999 – 2000 Graduate Research Assistant, Dept. of Math & CS, Truman State University, 1997 – 1999 Director of CAD Laboratory, Institute of Technology, PANDA Electronics Co., Ltd, 1995 – 1997 Electronic Engineer, Institute of Technology, PANDA Electronics Co., Ltd, 1995 – 1997 Assistant Electronic Engineer, Institute of Technology, PANDA Electronics Co., Ltd, 1989 – 1996

(c) Publications Five publications closely related to the project

[1] W. Zhu & W. He, "Wavelet Tight Frames for linear NURBs: Intervals", Essays on Mathematics and Statistics, Volume 4, Athens Institute for Education and Research, November 2013. [2] W. Zhu, "The multilevel structures of NURBs and NURBlets on intervals: Monograph on NUMBlets modeling," ISBN: 978-3-659-19291-3, Lambert Academic Publishing, Germany, November 2012. [3] W. Zhu, "Natural Cubic B-Splines Structure at the boundaries," International Journal of Science & Information, 2(3), 2012, pp. 34 – 40. [4] W. Zhu, "Existence condition for tight frame wavelets on NURBs: bounded," International Journal of Science & Technology, 1(1), 2011. [5] B. Li, W. Zhu & Q. Jiang, "Analysis and Design of Quadrilateral/Triangular Subdivision Schemes", Computer Aided Geometric Design, 26 (2009), pp. 904-922

Five Additional Selected Publications [1] F. Lin, K. Sookhanaphibarn & W. Zhu, "Observation of tsunami radiation at Tohoku by remote sensing", Science of Tsunami Hazards, 30(4), 2011, pp. 223 -232. [2] W. Zhu, D. Ukaivbe, "Simulation data with Bezier Curve with Subdivision Scheme", International Conference of Emerging Trends & Developments in Science Management & Technology, Raj Kumar Goel Institute of Technology, Ghaziabad, U.P., India, March, 2013 [3] F. Lin, K. Sookhanaphibarn & W. Zhu, "Overview tsunami detection by remote sensing," IEEE International Geoscience and Remote Sensing Symposium, IGARSS, Munich, Germany, July 2012. [4] W. Zhu & W. He, "Wavelet tight frames for linear NURBs," 5th Annual International Conference on Mathematics, Athens, Greece, June 2011. [5] W. He & W. Zhu, "Hierarchy structure of non-uniform rational B-Splines in multiresolution analysis: bounded intervals," Journal of Global Information Technology, 5(1), 2010, pp. 34 – 44.

(d) GRANT

"STEM Game and Project Based Learning (SGPBL) Program," CPI, \$428,098, Funded by US Department of Education, October 2016

(e) Synergistic Activities

(i) Students - Undergraduate research advisor: (1) Dafe Ukaivbe and Salamahn Mills, Dept. of Math & CS, Fall 2012 ~ present Research: Simulation in Modeling Data with Subdivisions in Matlab. (2) BreAsia Deal, LSAMP-URF Fellowship, Fall 2013 Research: Tsunami Early Warning System (3) Kenneth Nuemeta & Djomo Celine, UMES Honors Program, Fall 2013 Research: Tutorial software development in Calculus in applets (ii) Outreach/Public Lectures (1) 4th & 5th Innovative STEM Conference Judge/Reviewer committee, 2012 & 2013 (2) Advisor of Mathematics & CS lecture at UMES, 2011 ~ present

(iii) Professional Service - Membership: American Mathematical Society, Society for Industrial and Applied Mathematics, Tsunami Society International - Reviewer of International Journal of Science & Informatics (IJSI) and Precision Instrument and Mechanology (PIM).

(iv) Conferences organized or related subjects None

(e) Collaborators and other affiliations

W. He (University of Missouri-St. Louis), F. Lin (University of Maryland Eastern Shore), A. Chi (University of Maryland Eastern Shore), B. Li (Institute of Mathematics, Chinese Academy of Science), H. Wang (Beijing University of Aeronautics and Astronautics).

Co-editors: None

Graduate Students (co)-advised: 2

Postdoctoral researchers mentored in the past five years: None

Thesis Advisor: Wenjie He (University of Missouri-St. Louis)

Dr. Xianfang Tan, Delaware County Community College

Dr. Xianfang Tan is an assistant professor of Engineering Technology in the STEM division at Delaware County Community College. Her research interests include computer architecture, network-on-chip (ONoC), photonic/optical NoC etc.

On the Exploration of Game Based Learning in STEM Education with the Development and Application of Virtual Reality Course Modules

Abstract

Game-Based-Learning is a powerful pedagogy that can effectively motivate and engage students in studying. With specifically designed computer games, Game-Based Learning (GBL) can create an enjoyable and engage learning experience for students. The emerging Virtual Reality (VR) technology in recent years can significantly improve such learning experience. In this paper, we present our GBL practice in STEM education through the development and integration of VR-based GBL modules into the engineering curriculum. The feedback from students, the GBL module can effectively improve the learning experience for users. This work is part of an on-going project sponsored by the DoED. More advanced GBL modules with fascinating features will be developed in future.

Keywords

Virtual Reality (VR), Game-Based-Learning (GBL), STEM Education

1. Introduction

Researches have shown that the key to promoting the education in STEM stands on how to motivate and engage students in studying and improving learning performance [1]. Digital electronic technology has been widely used for communication, entertainment, and education. Computer games may provide powerful ‘hands-on’ tools for teaching practical and technical skills. Game-Based Learning (GBL), with specifically designed computer games, can provide an enhanced learning experience compared to traditional didactic methods and create an enjoyable and engaging learning experience for students [2]. In recent years great efforts have been put into realizing game-based learning for students [3-5], among which Virtual Reality, as one of emerging technologies, has been rapidly developed and widely adopted in our daily life [6], and also find its role in education [7]. By adopting the latest VR hardware/software, a more fascinating education environment can be built to promote students’ learning experience.

In this paper, we present our exploration and achievement in improving engineering education through GBL pedagogy, which more specifically is the development and integration of Virtual Reality (VR) games into different aspects of engineering education. This research is part of an on-going project supported by the DoED. In this project, several core engineering courses were selected and in each course, several major topics were chosen to develop course modules for this GBL application. For example, lab safety training was chosen and a VR game module has been developed for it. The whole game is completely developed by undergraduate students in electrical and computer engineering guided by engineering faculty. One of the essential goals of the project is to explore the most effective method to improve STEM education through GBL approach with innovative VR technology. Different patterns of embedding learning materials into VR games have been developed and evaluated for this goal.

The paper is organized as follows. Sect. 2 is the overview of the GBL module, Sect. 3 presents

the develop process, Sect. 4 shows the operation of the module, and Sect.5 concludes the whole work and discusses the future plan.

2. GBL Module Overview

2.1. Memory Game GBL Module

The first VR GBL course module developed is a memory game with object matching for lab safety. The memory game is in a traditional game style, in which a player is required to reveal hidden objects and remove them by matching. There are many memory games available online for playing as well as the development for different purposes, such as provided by appypie.com [8], as shown in Fig. 1 (a).



Fig. 1. An example of memory game available online.

In lab safety training, students will be introduced to different items related to the safety, such as fire extinguisher, goggle, slippers, etc. And the goal of the memory game GBL module is to strengthen player's understanding and awareness to those objects by interacting with them in the VR environment. In the GBL module, these objects are hidden in uniformed cubes. When playing the game, if a cube is picked up and raised to a required height, an object hidden inside will be revealed for matching. A player can cancel two matching objects can be canceled, until all cubes are cleared. The game process is shown in Fig. 2.

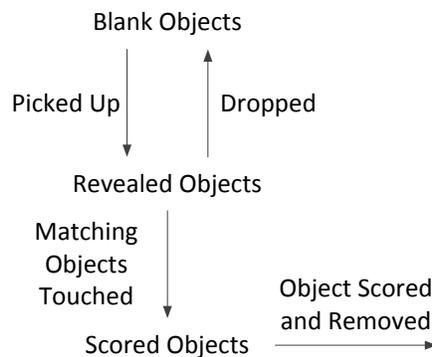


Fig. 2. Object matching game diagram.

The VR GBL module is developed in *Unity* [9]. *Unity* is a dominating industry game platform that can be used to build high-quality 3D and 2D games across mobile, desktop, VR/AR, consoles, etc. *Unity* has many amazing features including the embedded 2D and 3D physics engines. *Unity* is object-oriented, in which objects are known as game objects. The game objects in *Unity* are developed in C# and JavaScript. *Unity* provides excellent basic tutorials for learning the interfaces and workflow of the software. *Unity* is popular among small and independent game developers because of its simplicity and low cost (free for non-commercial use). Because of its popularity, many coders have developed packages/modules/libraries and posted on *Unity*'s online market for sale or even free to use, which greatly enriched its supportive resources and capacity. VR kits involved in the VR module developments are *Oculus Rift & Touch* [10] and *HTC VIVE* [11], both of which include headset, controllers and base station. An interactive GBL studio is specifically developed in University of Maryland Eastern Shore to support various GBL development activities and to accommodate related GBL teaching schemes. The studio is equipped with all required resources and facilities.

2.2. GBL Module Design Diagram

The diagram of the module is shown in Fig. 3. In the basic game, the gameplay is centered on the matching cubes with objects hiding in. The hidden object will be revealed when a cube is being held above a special height. When an object is revealed then it can be matched with another revealed object. If the player reveals two cubes that do not match, they have to put at least one down for the next try. When a cube is put down, it will decay after some time and revert back to a uniformed cube. When two matching objects are revealed touch each other, both of them will disappear. One game level will end when all cubes in the level are cleared and then the player can move forward to the next level, which has more cubes and more categories of hidden objects.

In developing the VR GBL module, to keep tracking the process during one game level, there are two tracked parameters, *boxCount* and *boxReq*. *boxReq* is set at the beginning of a level which is the total number of cubes to match. *boxCount* counts the number of cubes that have been correctly matched and removed from the level. When these two values are equal then the level is completed. The next level then will be initialized for a player to get started. When a level is started, there are a set amount of cubes to be placed and an amount of spawn locations to be filled. The level spawning script then assigns its cubes to locations and time stamps the level start. A global controller, which is a black game object, is defined to keep tracking the time and number of cubes left. When all cubes are cleared, the controller time stamps and calculates the time passed. It outputs and displays the time value to the player's view, and also notified the level spawner to spawn the next level.

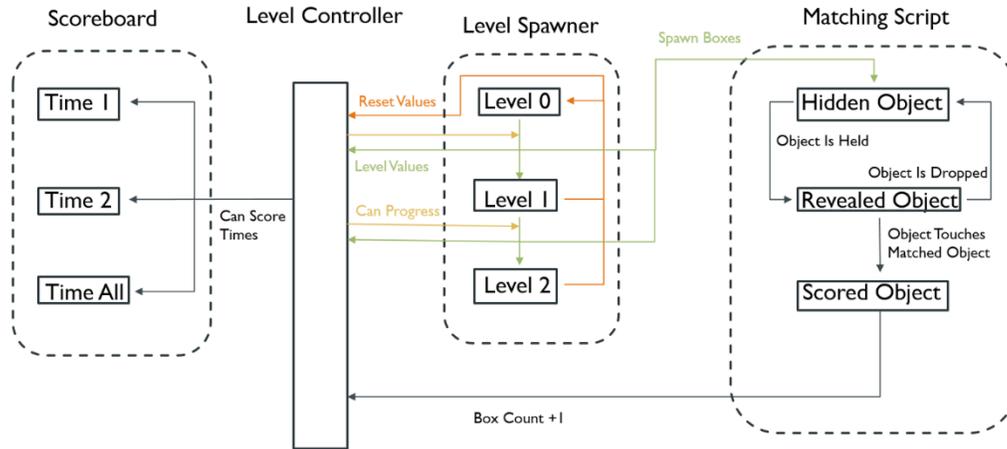


Fig. 3. Diagram of the GBL memory module.

3. VR GBL Module Development

3.1. Functions of the GBL module

The following functions are needed for the GBL module:

- *Hidden Object*: A uniformed cube with a hidden object initialized at the start of a game level. A cube reveals an object when lifted by the player.
- *Revealed Object*: An object revealed when condition satisfied. It is attached to a cube and can be color, texture or a 3D model. It will revert back to a Hidden Object if dropped by the player or be touched to a matching Revealed Object to be scored.
- *Scored Objects*: When revealed objects were matched they will be converted to scored objects, get counted for the score and be removed from the scene.
- *Matching Script*: This script is attached to all playable objects to be matched. This controls the matching conditions and the Revealing/Hiding conditions.
- *Level Spawner*: This script sets the values for level states. It also contains the reactions from the button presses to change these values. *Level 0* is the state before a game is started. All values like *boxCount* and *boxReq* are set to 0 and the start input is awaited. This is also the state that the game goes to when *Reset* button is pressed. When a level is started *boxCount* is set to 0 and *boxReq* is set to the number of boxes in the level (for example, 4 for *Level 1*). A time stamp is saved as the level starting time and the boxes are spawned according to a random seed which is a random integer that selects the positions of the boxes by the spawn locations.
- *Level Controller*: This is an empty game object with a script attached to it. The script stores all global variables to be used by other scripts. This script also checks the completion of a level after started. If a level is started and *boxReq* equals to *boxCount*, the script stamps the end time to calculate the total time used and notifies the Scoreboard to update the score.


```

// Update is called once per frame
void Update()
{
    if (transform.position.y < HowHigh && revealed == true && Time.time > revealTime + 10 * waitformatch)
    {
        revealed = false;
    }

    if (transform.position.y > HowHigh && revealed == false)
    {
        revealTime = Time.time;
        revealed = true;
    }
}

void OnCollisionEnter(Collision collision)
{
    Collider collider = collision.collider;

    if (collision.gameObject.CompareTag(tag) && transform.position.y > HowHigh && Time.time > revealTime + waitformatch && revealed == true)
    {
        collision.gameObject.SetActive(false);
        this.gameObject.SetActive(false);
        print("a " + this.tag + " touched " + collision.gameObject.tag + ", Y = " + transform.position.y);
    }
}

```

Fig. 5. The script of collision detection.

3.4. Scoreboard and Menus

The Scoreboard and Menu are displayed on a canvas in front of the playing area. This canvas is a floating menu in the 3D space which has 2 buttons, *Start* and *Reset*. The *Start* button starts the game from *Level 1* and moves on to later levels when the current level is finished. The *Reset* button reverts the game back to the state *Level 0*. The level initialization, progress, and reset are implemented by the script *Level_Spawn_Test.cs*. This script uses a few values stored in the Level Controller, an empty object made to hold global variables, including:

- *boxCount*: Current number of scored objects
- *boxReq*: Required amount of objects needed to finish the level
- *curLevel*: The current level
- *levelStarted*: A flag to indicate if a level is started
- *levelFinished*: A flag to indicate if the level is finished

When started, the game is at *Level 0* with *boxCount* equals to -1 and *boxReq* equals to 0 with no flags triggered. This is also the state when *Reset* is pressed. At the start of the game the level will increment by 1, *boxCount* set to 0, *boxReq* set to the predefined value, the *levelStarted* flag gets triggered, and the game objects are spawned randomly by the seed on predetermined locations.

3.5. Level Scoring

A second canvas Level Scoring is placed at the rear of the play area to show a player how well he or she played the game by displaying a score (time) on the board. When the *levelStarted* flag is triggered, a time stamp is taken and saved as *levelStart*. When the *levelFinished* flag is triggered a time stamp is taken again and saved as *levelFinish*. Then the difference between two stamps will be calculated to provide the elapsed time. When a level is finished an extra flag *scoreNow* is set to be true. This allows *Level_Score.cs* to update the score in the frame. If it is same or more than the original time nothing will happen, and it will change to the new one

otherwise. The time for the completed level is displayed and a total time is added up. Then the flag set back to false.

4. VR Course Module

The operation of the VR GBL lab safety module is shown in Fig. 6, in which (a) shows the initialization of the game, (b) shows the reveal of a fire extinguisher from the cube, (c) shows two revealed fire extinguishers to match, and (d) shows the scoreboard after the 1st level completed.

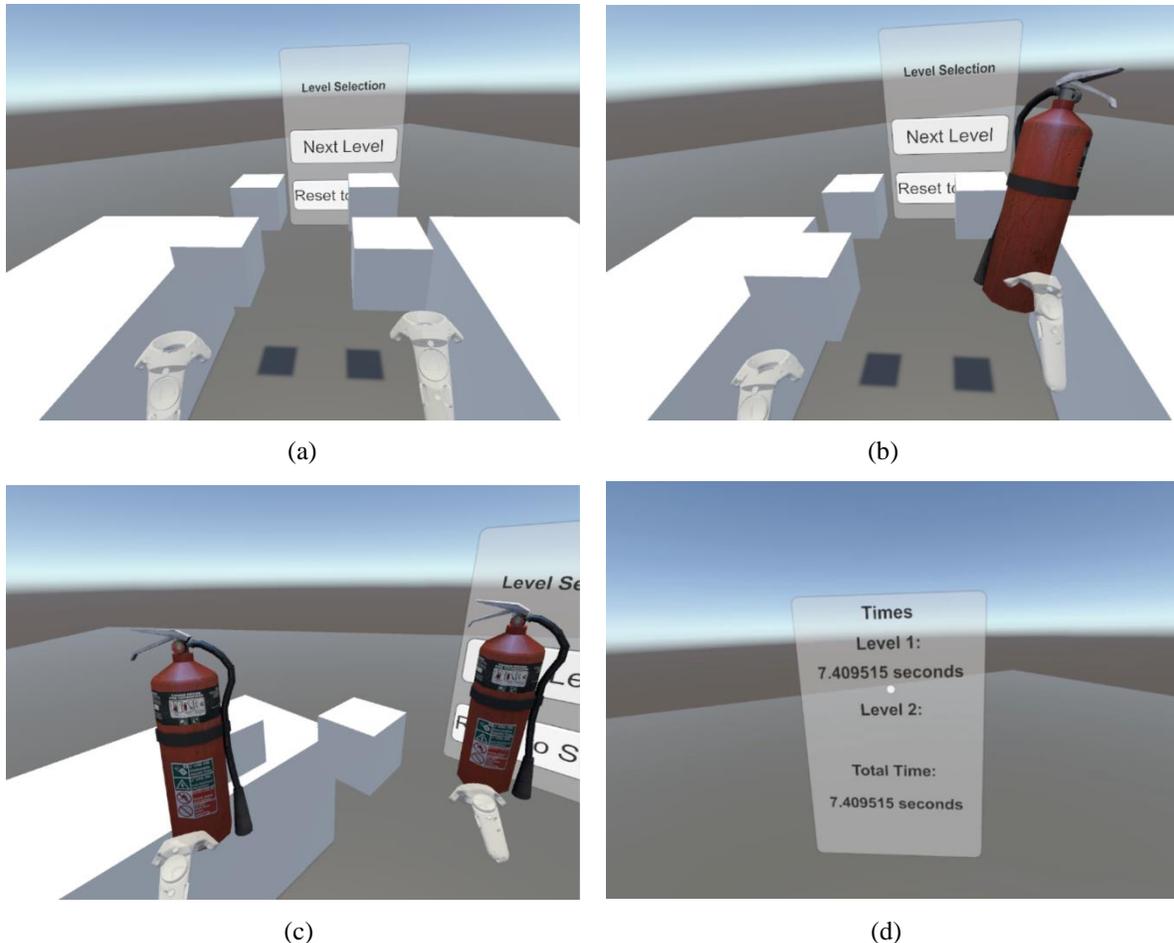


Fig. 6. (a) Start of the game, (b) reveal one object, (c) reveal a matching object, and (d) Scoreboard

The development of the lab safety module is completed recently and it is integrating into engineering lab teaching. Based on the feedbacks collected from students who have tested the module, they are extremely interested in the integration of such module into their learning process. Moreover, the development process has provided students involved a comprehensive training on programming with high-level languages and valuable experience in software development and project conducting, which can remarkable benefit their profession and career development. Meanwhile, many students have shown strong interest to participate the project to learn programming and VR software development. As a practice of our Vertical Integration

Program (VIP), a team of undergraduate ranging from freshman to senior students is organized to work on the development of more VR GBL modules.

5. Conclusion and Future Plan

In this article, we present our experience in the development of GBL module in VR environment. The first module developed is a memory game to match different objects related to lab safety training. Through the development process, our students have received productive training on different aspects. Feedback from students has shown the potential that the course module has greatly improved their learning experience.

Moreover, the object matching GBL module is an open template that can be easily customized for the learning of different subjects of all levels, for example, to match the question in one hand and the answer in the other hand. By this way the engagement and motivation of learners range from elementary school to college, may be greatly enhanced by the latest VR technology.

And we have moved into the second phase of the VR GBL development. A new archery game module is going to be created, in which the player needs to shoot the right answer of the question displayed. This can be used in testing different courses. In short, with the support of fast advancing VR technologies, the learning process of STEM students will become more and more attractive and fascinating.

References

- [1] X. Chen, "STEM Attrition: College Students' Paths into and out of STEM Fields. Statistical Analysis Report. NCES 2014-001," *National Center for Education Statistics*, 2013.
- [2] S. Tang and M. Hanneghan, "A Model-Driven Framework to Support Development of Serious Games for Game-based Learning," in *Developments in E-systems Engineering (DESE), 2010*, 2010, pp. 95-100.
- [3] S. Chatterjee, A. Mohanty, and B. Bhattacharya, "Computer Game-Based Learning and Pedagogical Contexts: Initial Findings from a Field Study," in *Technology for Education (T4E), 2011 IEEE International Conference on*, 2011, pp. 109-115.
- [4] C. Jui-Hung, T. K. Shih, and C. Jui-Yi, "To develop the ubiquitous adventure RPG (role play game) game-based learning system," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2012, pp. 2973-2978.
- [5] C. Ben, "Pilot Study of Past Decade Game-Based Learning Journal Papers Survey from the Technology Perspective," in *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, 2010, pp. 748-749.
- [6] J. M. Loomis, "Presence in virtual reality and everyday life: Immersion within a world of representation," *PRESENCE: Teleoperators and Virtual Environments*, vol. 25, no. 2, pp. 169-174, 2016.
- [7] T. Ma, X. Xiao, W. Wee, C. Y. Han, and X. Zhou, "A 3D Virtual Learning System for STEM Education," in *International Conference on Virtual, Augmented and Mixed Reality*, 2014, pp. 63-72: Springer.
- [8] Appypie. Available: <http://snappy.appypie.com/gamebuilder>
- [9] Unity. Available: <https://unity3d.com/>
- [10] Oculus. Available: <https://www.oculus.com/>
- [11] HTC VIVE. Available: <https://www.vive.com>