https://doi.org/10.2352/ISSN.2470-1173.2019.2.ERVR-182

An Approach to Game Prototyping with a Tangible Interface

Daniel S. Buckstein; Division of Information Technology & Sciences; Champlain College; Burlington, VT/USA: dbuckstein@champlain.edu;
Michael Gharbharan, Andrew Hogue; Faculty of Business & IT; University of Ontario Institute of Technology; Oshawa, ON/CAN: andrew.hogue@uoit.ca;

Abstract

In this paper, we explore the use of tangible user interfaces in the context of prototyping digital game design. Traditional approaches to game design use a combination of digital and non-digital prototyping techniques to identify core game mechanics and object placements when designing levels followed by a laborious process where the non-digital prototypes are translated into digital counterparts in a game engine. The presented system aims at reducing the need for non-digital prototypes by providing an easy to use interface that doesn't require technical expertise and results in a playable digital prototype in a modern game engine. The presented system uses Augmented Reality markers as the tangible interface to facilitate specific functionality in a modern game engine. A preliminary user study is presented to understand the current strengths/weaknesses of this approach. Our hypothesis is that our system would improve the game design experience for users with respect to usability, performance, creativity support and enjoyment as we firmly believe that the process of designing games should also be an enjoyable process.

Introduction

Traditional approaches to game design can be a very time consuming iterative process since both digital and non-digital prototyping techniques are frequently revisited before final design decisions are made. Non-digital prototyping techniques are used by game designers to determine placement of objects and characters, to identify weaknesses in level design, and to create physical playable board-game mockups to facilitate iteration of game mechanics parameters. Moreover, these non-digital prototypes must be translated into playable digital counterparts into the final game engine (a non-trivial task requiring technical expertise). Non-digital prototyping (sometimes called paper or physical prototyping) tends to be a more playful and hands-on process where designers can modify rules and item placements to quickly understand how these changes affect gameplay. Digital prototyping requires the use of a game engine (e.g. Unity3D or Unreal) and the use of a mouse and keyboard interface. For users without expertise in these engines, this process can be cumbersome and require a significant amount of training to create even the simplest of digital prototypes. Can we create interfaces that individuals without significant expertise in game engines could use? Could we enhance the process of game design prototyping by making it an enjoyable and playful process? If so, designing digital games would become even more accessible to those without programming expertise and could result in interesting gameplay. We hypothesize that bringing playfulness to digital game design will facilitate a more efficient and enjoyable development process.

To address these issues, we have developed ***PlayTIME****: a* ***T****angible* ***I****nteractive* ***M****edia* ***E****nvironment for Game****Play***. The conceptual target of this system is to ultimately create a collaborative environment where designers focus on specific aspects of game design and asset creation without the need to learn complex software development systems (e.g. 3DS Max, Maya, Unity3D, Unreal). The system presented in this paper is focused on the use of tangible interfaces in the form of Augmented Reality (AR) markers in a level design context. These tangible user interfaces (TUI) are used to perform specific development tasks in a popular game engine (Unity3D). To identify whether these interfaces are in fact more enjoyable to use than traditional approaches, our proof-of-concept prototype enables developers to focus on basic scene layout and creation tasks such as object placement/movement, object property manipulation, and virtual camera control. The system identifies which of these tasks are to be performed, records their movements and applies them to virtual assets to create a playable digital prototype of the game.

In this paper, we also present a preliminary user study that evaluates the current implementation of PlayTIME to gain an understanding of how game designers interpret and work with the system. The study aimed to explore the usability of the interface, how it fosters creativity in a game design project, and how the users enjoyed the system versus solely using Unity3D.

Related Work

Augmented Reality (AR) technology has seen an increase in sophistication that can enable the creation of new interaction techniques using physical objects. While AR has been shown to be usable in many digital game applications, developers mostly use AR as a way for the end-user to interact with the game thus creating new types of gameplay [1]. The MagicBook–introduced by Billinghurst in [2, 3]– demonstrated the use of AR as a storytelling mechanism where each page of a book had an augmented animatiom component that could be visualized. These animations needed to be pre-defined and were not interactive. In [9], students prototyped tangible interfaces for a variety of applications by attaching fiducial markers to movable parts; markers were used as physical widgets that allowed the end-user to interact with the physical prototype. In [10], the authors discuss designing tangible interfaces using AR, and identify special actions that users can perform with AR markers, such as shaking. The authors found that the tangible layout allowed players to use spatial reasoning and communication. Commercial tangible interfaces such as *Osmo* [19] provide children with hands-on, playful, tabletop-based exercises in spatial awareness and problem solving. While tangible interfaces are certainly effective as the end-user interaction mechanism, questions remain regarding how to effectively augment existing game prototyping methods to use tangible user interfaces (TUIs).

Can TUIs be as effective for use with modern game engines during the design phase? Can TUIs augment and make the game prototyping experience more enjoyable? Can TUIs engage non-technical individuals in the creation of non-digital games?

ToyVision–introduced in [15]–is a digital tabletop tangible prototyping kit that uses fiduciary markers to track objects that, when interacting with each other, establish game play. ToyVision is used to prototype a tabletop game in [16] however it hasn't been shown to be effective at prototyping 3D environments nor has a study been performed to determine how these types of tangible interfaces compare with the existing state-of-the-art game design and development engines. In this paper, we explore the use of AR paddles in the context of game development with a modern game engine (specifically Unity3D) to determine whether the process of designing digital prototypes can be made more effective and more enjoyable by using simple tangible interfaces.

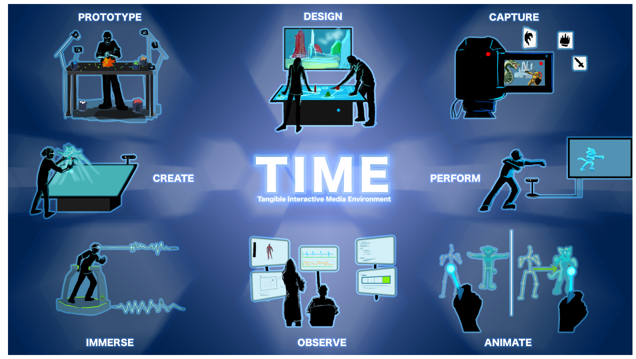

Figure 1: The concept of TIME: a Tangible Interactive Media Environment for prototyping interactive digital media. The top-middle "Design Station" is the focus of this paper we call PlayTIME.

Basic Concept
Traditional Design Methods

Building a non-digital or physical prototype involves the use of everyday materials. Paper, pens, toys and other readily accessible supplies are used to map out an interactive scenario, for example a level in a game, and allow designers to pre-visualize their ideas. Also called paper prototyping, within and outside of the context of game design, this process is extremely flexible because the objects can be handled and replaced at will to suit the needs of designers. Furthermore, we as humans understand spatial tasks. Non-digital prototyping yields the fundamental rules and mechanics, which are then forwarded to developers as guidelines for creating the digital product.

Game engines such as Unity3D and Unreal are popular, free, and can be used to create 3D environments. These engines come with a vast set of features and the ability to preview the game instantly. However, these engines have a steep learning curve and require the user to have a good grasp of programming concepts to create even the simplest of interactive environments. While tutorials are certainly helpful to get started, sometimes you just want to dive in and get a prototype completed quickly so you can iterate on interaction design. Moreover, those without technical backgrounds either have to rely on others with these technical skills or take the time to learn these skills themselves. Even with these modern tools, the process of developing a digital game is still very time consuming. In contrast, physical prototyping is meant to be agile and easy to iterate upon.

Tangible Interactive Media Environment

Our system is part of a suite of independent interfaces that utilize the same underlying framework for communication. We define each of these systems as a *Tangible Interactive Media Environment*, or ***TIME***.

The underlying conceptual principle of TIME is that all aspects of digital media development can be performed more easily with tangible interfaces in a collaborative manner. This can be achieved by enforcing the use of a common interaction language where users do not require extensive training but rather they can explore the creative space iteratively and naturally using their bodies and physical objects (e.g. toys, lego, etc.). TIME can be demonstrated more clearly when thinking of the Herculean task of designing a video game. When designing a game, developers must create prototypes and final assets (for characters, levels, items, etc), place them in the scene at appropriate locations, designate properties of these assets, determine trigger events/areas and cause-effect relationships, keyframe camera and lighting motion for cut-scenes, rig character skeletons, and produce a series of keyframe animations tied to each asset that can be blended together at appropriate times. Our concept of TIME separates these tasks into separate independent "stations" where developers can focus on these specific tasks and move the information to other stations by utilizing cloud data storage and synchronization. Content creators working on different parts of a game with would interact with these independent systems in different ways: a 3D modeler would use Lego bricks as a tangible method of constructing rough, static model resources for the game (1 top left). Meanwhile, rough 3D models would be accessed by a sculptor who would use "virtual clay" and hand gestures, tracked with a depth sensor, to smooth and refine the model (1 middle left). For example, the object could be placed in the scene by a level designer (1 middle top). As the data is created and refined, objects in a TIME-managed scene automatically synchronize using cloud-based data synchronization (current implementation uses Dropbox) between stations. This allows any display station to have an complete "snapshot" of the current version of the game at all times. We ensure this to enable a "play-at-any-moment" policy to encourage iteration and identify problem areas quickly. For the purposes of this paper, we focus solely on the "Design Station" we have dubbed PlayTIME.

PlayTIME: A Tangible Interface for Game Play Design

PlayTIME aims to be a tangible "level design station" created specifically for game design and prototyping. The current preliminary implementation works within Unity3D as a plugin to augment traditional level design approaches and uses a set of fiduciary markers Figure 2. Each marker is used to perform some task in the editor: there are object placement markers to add the player character and enemies to the game world, behaviour markers so enemies can respond appropriately to the player in-game, and a camera pan marker to shift the view within the editor.

We have defined a set of 11 markers (See Figure 2) that augment specific functionality of the implemented scene. These markers can be placed in the scene at any time and can be used for object selection (Selection Tool), moving objects and re-orienting selected objects (Manipulate Tool), placing and orienting the vir-

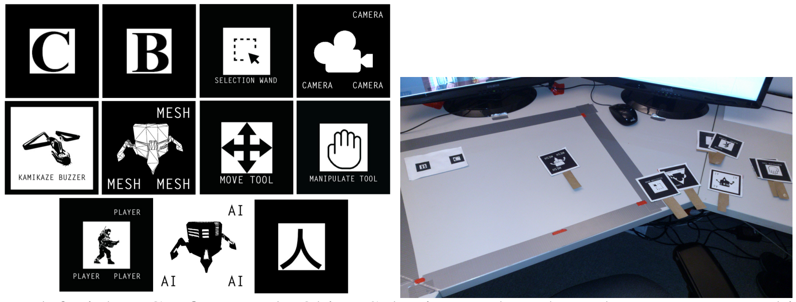

Figure 2: Left: AR Tools from left-right—Confirm, Back, Object Selection Tool used to select one or more objects to manipulate, Camera Manipulation Tool used to move the virtual camera viewpoint, Kamikaze Buzzer Tool used to add gameplay specific element to the scene, Mesh placement tool used to add a new object to the scene, Move or Panning tool used to pan the view, Object Manipulation tool used to move and orient objects in the scene, Player tool used to place and orient the player in the scene, AI behaviour tool used to attach AI behaviours to objects, Calibration tool used to calibrate the camera and AR parameters. Right: Interactive workspace area of prototype system.

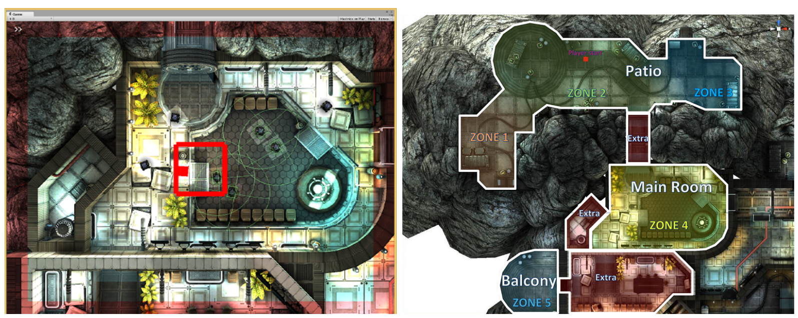

Figure 3: An overview of the Unity3D implementation. Left: top-down view of the "main-room" zone with an overlay of AR input markers; Right: top-down view of the entire map users can manipulate.

tual camera (Camera tool), panning the top-down game level view to move to a different area in the scene (Move tool), placing a mesh object (Mesh tool), placing the player (Player Tool) or a buzzer (Kamikaze Buzzer tool) or attach an AI behaviour (AI tool). We also define a Confirm, Back and AR Calibration tool to use if necessary. With this simple set of AR tools a wide variety of gameplay levels can be iterated upon quickly simply by placing the AI objects in the scene, setting the player's initial position and orientation, and placing meshes with pre-defined behaviours. We employed image-based markers rather than using barcode-type markers to clearly communicate the marker functionality to the user. Our current system prototype in Unity3D can be seen in Figure 3 which provides users with a pre-built game world without enemies and behaviours. Users must place enemies and attach behaviours to the assets to make a functional digital game prototype.

Evaluation

Study Overview

We conducted a preliminary user study to evaluate *usability*, *creative support* and *enjoyment* of our system. Our hypotheses were as follows:

1. ***Usability***: The use of tangibles will have a significant positive effect on the *performance* of users.

2. ***Creativity***: The use of tangibles will have a significant positive effect and on the users' *creative output*.

3. ***Enjoyment***: The use of tangibles will have a significant positive effect on the users' *enjoyment* of the activity.

Each participant was asked to complete a pre-determined design task to create a level with specific constraints in a given Unity3D project in two randomly ordered conditions (a) using the tangible system as an extension of Unity's editor (without working within the editor itself), and b) using Unity's editor as-is, with the underlying tangible system disabled. The Unity editor condition was controlled using a mouse and keyboard.

Participants

20 university and college students in digital media participated in the study. All participants had some degree of experience in creative domains: game and application development, film production, 2D and 3D art are a few of the specializations included in the demographic sample. All participants reported some familiarity with game engines such as Unity, so the experience of running the study was not entirely new. None of the participants had prior knowledge of, or experience with, tangible user interfaces.

Method

Session Overview

After informed consent, a demographics questionnaire was administered to understand participants' experience in multimedia design, programming, art, and related disciplines. Each participant completed two activity conditions, (1) using the tangible interface, and (2) using Unity3D. The condition order was alternated between participants to minimize learning effects. After being briefed on the activity (discussed below), participants were provided a short orientation and step-by-step instructions on how to use the system. Each activity condition was limited to 20 minutes, but they could stop when they felt they had completed the task to the best of their abilities. A post-activity questionnaire was administered to collect data after each condition. After both activity conditions were completed, participants completed a questionnaire to directly compare the two systems used and indicate system strengths/weaknesses and their preference.

Experimental Setup

The experimental setup consisted of an open workspace on a desk, seen in Figure 2 in the Unity3D condition, a mouse and keyboard was used and in the TUI condition, AR marker paddles were solely used to interact with the system. The game level editor was displayed on two monitors positioned behind the workspace one showcasing the current editing view and the other showing a gameplay preview that can be viewed at any time. A camera above the workspace was used to detect and capture the AR marker locations and orientations. Users were also provided with a printed map of the level.

User Activity

The users' assigned task was focused on object placement and configuration within a game scene. They were provided with a pre-built Unity3D project with a complex level and no characters, and given access to a single folder of assets. Guided by a map of a portion of the level and list of object placements required to make the game winnable, players were asked to place the player token at a location in the start zone and enemies throughout the labeled areas in the game world. Creativity was encouraged as long as enough enemies were placed for the game to be playable. Users were instructed that the player must be able to clear all areas of enemies in five minutes of gameplay or less. Users were encouraged to play the game at any moment to see the effect of their designs.

Data Collection

The post-condition questionnaire was administered after each condition and consisted of three standard Likert-scale based surveys. For usability and satisfaction using the systems, we used the *Computer System Usability Questionnaire* (CSUQ), also known as the *Post-Study System Usability Questionnaire* (PSSUQ) [12][13]. Usability is one of the most important and frequently sought-after attributes of systems designed in both the domains of HCI and game design. It was critical to select a qualitative measurement that would quickly and simply provide an overview of PlayTIME's usability. We used the Creativity Support Index (CSI) [5][6] questionnaire to determine how participants felt about the systems' abilities to support creativity and expression, immersion, enjoyment, exploration, producing desired results and collaboration. Since our system is designed to support creative expression, and enjoyability, it was important to find a qualitative measurement that could explain how well the interface supported these areas. We used the Positive and Negative Affect Schedule (PANAS) questionnaire [20] as a quantitative measure of enjoyment. The questionnaire gives two separate scores: *positive affect*, measuring strong *positive* emotions (e.g. excited), and *negative affect*, measuring strong *negative* emotions (e.g. distressed). PANAS was selected due to its popularity across various scientific fields and was validated in [20] and [7].

Data Analysis

For performance analysis, we measured and annotated the time spent performing a wide variety of tasks using each system (e.g. placed object, removed object, moved object etc), and the distribution of time spent using each feature in the experiment. The resulting scenes created were analyzed to tell how far they deviated from the instructions they were given. A weighted Manhattan distance [4] of each object was used to measure the deviation of each scene from the exercise that the participants were assigned. For significance testing on all survey responses, performance times recorded from the screen captures, and scene Manhattan scores, three methods for non-parametric data were used: the *Kruskal-Wallis one-way analysis of variance by ranks* [11] (KW) was used to analyse the effects of both the condition and the activity order; the *Wilcoxon Signed-Rank test* [21] ("WSR") to analyse the effect of the conditions only (cond. P data against cond. U data); and the *Mann-Whitney-Wilcoxon rank-sum test* [14][21] ("MWW"), also called the *Mann-Whitney U test*, to analyse the effect of the activity order (P-first data against U-first data). A p-value was collected for each metric (p_K, p_W respectively), denoting the significance of the effects, telling us how much the data from each set differed from the other sets. For all p-values, significance was considered if $p < 0.05$.

Results

The PANAS questionnaire, administered before the study and after each condition, describes the conditions' effects on the participants' emotions. Figure 4 shows the progression of PANAS scores over the activity. Changing the condition had a significant effect on the *positive affect change* through the conditions ($p_K = 0.02532$, $p_W = 0.04362$). The same metric was significantly affected by the activity order, for both the PlayTIME results ($p_K = 0.01238$, $p_M = 0.01377$) and the Unity results ($p_K = 0.004003$, $p_M = 0.00451$). The CSUQ allowed participants to report the usability of PlayTIME and Unity. Figure 5 shows the average CSUQ scores. We found that changing the condition had a significant effect on the *system usability* score metric ($p_K = 0.03021$, $p_W = 0.009305$). The average system usability score with PlayTIME was 5.069 (SD=1.166) out of 7. With Unity, the average was 5.688 (SD=1.228). The CSI (Creativity Support Index) scores did not show any significant effect related to the activities themselves or the activity order. The average PlayTIME CSI score for all participants was 70.125 (SD=17.819) and the average Unity CSI score was 74.925 (SD=14.471).

Discussion

The PlayTIME design activity caught participants' interest the moment they figured out what it would be used for, sometimes before they were briefed, and overall participants reported that their positive emotions had increased significantly during the activity. Furthermore, their negative emotions, which were low to begin with, had generally decreased. In contrast, the Unity design activity generally had a much smaller or opposite effect on the positive affect scores. Although this measure does not contribute to system usability, the consistently positive emotions tell us something about PlayTIME that is important for users: *it is fun*. Perhaps seeing and using this novel technique for scenario design was exciting and enjoyable for users because it felt like play. Some of the participants commented on this in the feedback portion of the

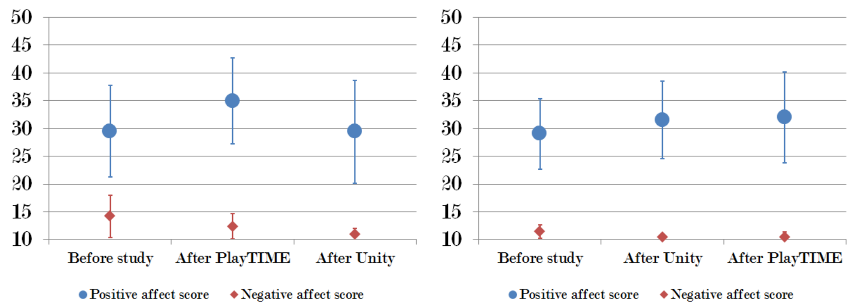

Figure 4: The average PANAS scores, ranging from 10 to 50, as a progression over the duration of the study for the PlayTIME-first and Unity-first participants. The error bars represent one standard deviation from the mean.

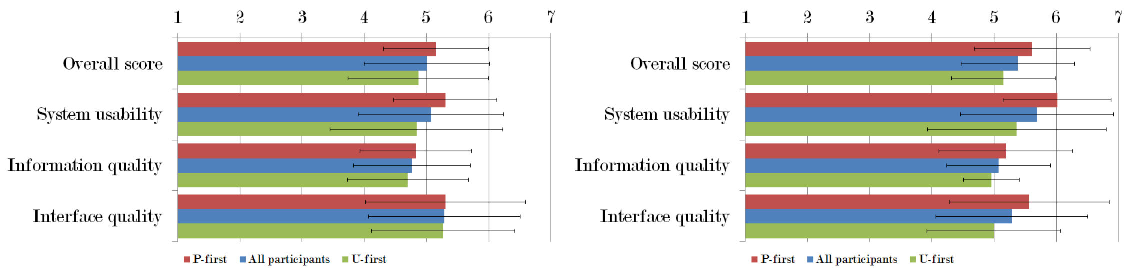

Figure 5: The average CSUQ/PSSUQ scores for all participants between both groups. The error bars represent one standard deviation from the mean.

post-study questionnaire:

> *"I enjoyed PlayTIME more... It also made it seem more like playing a game to create a game; I was almost more interested in making [my game] than playing it."*

> *"It was far more interesting to be using technologies that I'd never used before, whereas I am almost always using just a mouse and keyboard, or as in this case, a mouse alone."*

> *"PlayTIME was definitely more enjoyable because it felt like I was playing a game as opposed to working."*

This is great for scenario designers, since one of PlayTIME's purposes is to have game development feel more interactive by treating it like play. To see people agreeing with this purpose, that PlayTIME was indeed playful and fun, means it is doing its job at making game design more conducive. Despite the positive usability scores, the lowest of all CSUQ ratings were consistently from Q9, regarding error messages. For PlayTIME the overall scores for Q9 averaged 2.7 (SD=1.382) and for Unity the scores averaged 2.85 (SD=1.74). Errors occurred often, and were frequently repeated, because neither system provided any indication that something had gone wrong in the first place. This was an oversight in the system's design; specific errors were discovered that were not anticipated. In some cases, users would commit an error and not realize until they visited that location in the map several minutes later. For example: with PlayTIME, objects were frequently left selected while working on a completely different area of the level, and an accidental or intentional occlusion of the 'cancel' marker would cause the off-screen objects to be deleted. Users were allowed to ask for help when things got too frustrating, in which case the observer would describe the necessary steps to correct the error. This goes back to two of Nielsen's usability heuristics: error prevention, and error recognition, diagnosis and recovery [17][18]. Based on the user responses and performance, both of the systems require additional user error identification and feedback.

Limitations

With the exception of previewing time and total user error time, which remained basically the same for all groups, the PlayTIME-first group had consistently high significance (lower p-values) compared to the Unity-first group. This is likely because the activity times for the people who ran PlayTIME first were quite far apart, whereas the people who ran Unity first had less of a difference between their times. The participants' first activity always took longer which indicates that order had an effect on the total activity time. The KW test returned 0.0001571 for PlayTIME and 0.005159 for Unity; these indicate high significant differences between the conditions. The feedback leads us to a very logical explanation for this. The final question in the post-study survey was "*Did doing the first activity help you complete the second activity in any way?*" All participants reported that the order had a definite impact on their results. Those who cited reasons generally agreed that the reason was simple: since they had already done the activity once, they were able to skip the learning process for the task and just complete the activity, which they had memorized; the learning curve for the system itself may have still been an issue. Furthermore, the performance capture revealed that both systems were susceptible to Fitts's law [8]: for PlayTIME the detrimental factor was the general unfamiliarity with the AR paddles, slowing down selection time, and in Unity the targets for selection were generally small requiring the user to zoom in and out.

Conclusions & Future Work

Here we introduced a conceptual framework for a Tangible Interactive Media Environment and a preliminary implementation of a design station we call "PlayTIME". Our goal was to create an interface that could augment existing game design engine functionality to make the process easier to perform tasks and be enjoyable at the same time. We performed an initial user-study to find evidence supporting or contradicting hypotheses relating to usability, creativity and enjoyment of this new system with respect to the existing method for game design in Unity3D. Based on the results of this preliminary study, we can *accept* or *reject* our initial hypotheses as follows:

1. ***Usability***: This hypothesis is ultimately rejected since, although the use of PlayTIME had statistically significant effects on the CSUQ results and performance, the results are not in PlayTIME's favour.

2. ***Creativity***: This hypothesis is rejected since changing the system used showed no significant effects on the CSI results and scenes.

3. ***Enjoyment***: This hypothesis is accepted since the PANAS yielded favourable positive affect results for PlayTIME. Additionally, the feedback was generally positive towards PlayTIME, with citations of its novelty and fun factor.

The feedback presents us with clear indications that PlayTIME is a step on the right track towards fun and collaborative digitalization of physical game prototyping; we now know that tangible AR has the benefit of being fun as a game design tool, which could lead to a more collaborative, conducive and less iterative development experience for designers. However there are several limitations: The performance analysis showed that one of the main hindrances in the PlayTIME experience was the use of primitive fiduciary AR technology. Future implementations of the system should take into consideration newer and more stable AR tracking technologies. Marker-less tracking technologies may also be used to increase awareness of the tangible objects' functionalities.

Future studies evaluating the efficacy of PlayTIME include, but are not limited to: determining how familiarity with PlayTIME affects the performance and perceived usability; using biometrics and physiological measures to evaluate emotional responses elicited by PlayTIME; and comparing the creative output of children versus industry professionals using PlayTIME

Acknowledgements

This section has been removed for review.

References

[1] Mark Billinghurst, Hirokazu Kato, Kiyoshi Kiyokawa, Daniel Belcher, and Ivan Poupyrev. 2002. Experiments with Face-To-Face Collaborative AR Interfaces. *Virtual Reality* 6, 3 (2002), 107–121. DOI:http://dx.doi.org/10.1007/s100550200012

[2] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2001a. The MagicBook: a transitional AR interface. *Computers & Graphics* 25 (2001), 745–753.

[3] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2001b. MagicBook: Transitioning Between Reality and Virtuality. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 25–26. DOI:http://dx.doi.org/10.1145/634067.634087

[4] Paul E. Black. 2006. Manhattan distance. [Online], (2006). nist.gov/dads/HTML/manhattanDistance.html

[5] Erin A. Carroll and Celine Latulipe. 2009. The Creativity Support Index. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 4009–4014. DOI:http://dx.doi.org/10.1145/1520340.1520609

[6] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (June 2014), 25 pages. DOI:http://dx.doi.org/10.1145/2617588

[7] John R. Crawford and Julie D. Henry. 2004. The Positive and Negative Affect Schedule (PANAS): Construct validity, measurement properties and normative data in a large non-clinical sample. *British Journal of Clinical Psychology* 43, 3 (2004), 245–265.

[8] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.

[9] Eva Hornecker and Thomas Psik. 2005. Using ARToolKit Markers to Build Tangible Prototypes and Simulate Other Technologies. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'05)*. Springer-Verlag, Berlin, Heidelberg, 30–42. DOI:http://dx.doi.org/10.1007/11555261_6

[10] Hirokazu Kato, Mark Billinghurst, Ivan Poupyrev, Kenji Imamoto, and Keihachiro Tachibana. 2000. Virtual object manipulation on a table-top AR environment. In *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on*. 111–119. DOI:http://dx.doi.org/10.1109/ISAR.2000.880934

[11] William H. Kruskal and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* 47, 260 (1952), 583–621.

[12] James R. Lewis. 1992. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 36, 16 (1992), 1259–1260.

[13] James R. Lewis. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7, 1 (1995), 57–78.

[14] Henry B. Mann and Donald R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* (1947), 50–60.

[15] Javier Marco, Eva Cerezo, and Sandra Baldassarri. 2012. ToyVision: A Toolkit for Prototyping Tabletop Tangible Games. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '12)*. ACM, New York, NY, USA, 71–80. DOI:http://dx.doi.org/10.1145/2305484.2305498

[16] Javier Marco, Ian Oakley, Eva Cerezo, and Sandra Baldassarri. 2013. Designing and Making a Tangible Tabletop Game with ToyVision. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 423–426. DOI:http://dx.doi.org/10.1145/2460625.2460719

[17] Rolf Molich and Jakob Nielsen. 1990. Improving a Human-computer Dialogue. *Commun. ACM* 33, 3 (March 1990), 338–348. DOI:http://dx.doi.org/10.1145/77481.77486

[18] Jakob Nielsen. 1994. Heuristic Evaluation. In *Usability Inspection Methods*, Jakob Nielsen and Robert L. Mack (Eds.). John Wiley & Sons, Inc., New York, NY, USA, 25–62. dl.acm.org/citation.cfm?id=189200.189209
[19] Tangible Play, Inc. 2014. Osmo. (2014). playosmo.com
[20] David Watson, Lee A. Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology* 54, 6 (1988), 1063.
[21] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* (1945), 80–83.

Author Biography

Daniel Buckstein is a graduate of UOIT in Ontario, Canada. Dan holds a Master's degree in Computer Science and a Bachelor's degree in Information Technology, majoring in Game Development & Entrepreneurship. His programming and development interests include game engines, tools, mixed reality, graphics, physics and animation. He has a keen interest various areas of mathematics, including calculus and linear algebra, both of which carry over to computer animation and graphics techniques.

Michael Gharbharan is a graduate of UOIT as well and holds a Master's degree in Computer Science and a Bachelor's degree in Information Technology, majoring in Game Development & Entrepreneurship. Michael's research revolves around the use of tangible interfaces in game design scenarios.

Andrew Hogue is an Associate Professor at the University of Ontario Institute of Technology in the Faculty of Business & IT and a founding member of the GAMERlab at UOIT and received his PhD from York University in 2007. His expertise lies in developing new interfaces for Game Design, Virtual Reality, and investigating issues in stereoscopic 3D as it pertains to creating content.