Enabling Participatory Design of 3D Virtual Scenes on Mobile Devices

Xiaowen Sun[1], Yafang Wang[1]*, Gerard de Melo[2], Wei Gai[1], Yuliang Shi[1]*
Lu Zhao[1], Yulong Bian[1], Juan Liu[1], Chenglei Yang[1], Xiangxu Meng[1]
1: Shandong University, Jinan, China
2: Rutgers University, New Brunswick, NJ, USA
yafang.wang@sdu.edu.cn, shiyuliang@sdu.edu.cn

ABSTRACT

Designing 3D virtual scenes is essential for computer animation, computer game design, and virtual reality (VR) applications such as virtual museums. In this paper, we present a novel paradigm for participatory design of 3D virtual scenes on mobile devices. Designers and users are actively engaged in the design process so as to ensure that the results meet their needs with high standards of usability. Our new system allows the designers to construct an initial virtual scene via two tablet devices, one of which supports the scene assembly in a 2D window, while the other displays the corresponding 3D scene synchronously. Subsequently, the designers adapt the 3D scene based on real-time feedback of users exploring the virtual scene via a VR device, consisting of a smartphone with cardboard 3D glasses. The participants can then discuss how to further fine-tune the virtual scene by replaying the recorded footage of the user's experience process, interacting on their own respective tablets or smartphones. The system has been applied to the design of VR environments for virtual museums, residential decoration, firefighter training, and 3D games. Our user study suggests that this system not only increases the efficiency of the design process, but also gives rise to better designs.

Keywords

Virtual Scene Modeling; User Experience; Multi-Touch Interfaces; Participatory Design

1. INTRODUCTION

3D virtual scenes are widely applied in virtual reality (VR) applications, computer animation, gaming, and so on. With the rapid advances of 3D modeling technology (such as 3D scanning) and the emergence of large model libraries, it has become possible to assemble virtual scenes based on existing object models. For example, [37] introduced a personalized virtual museum modeling software based on Web3D. It is based on 2D mouse and keyboard interaction.

Recently, multi-touch interfaces have been an important area of investigation within the field of Human-Computer Interaction because of the high degree of freedom and the direct unmediated touch experience that they provide [5]. Kin et al. [20] described a multi-touch set construction system enabling artists to construct virtual organic environments by selecting and positioning geometric models of objects and of vegetation. There has been some work on multi-touch for 3D scene construction, but for the most part such work has been unable to overcome the drawbacks of direct 3D manipulation on touch screens. In particular, when interacting with a vertically positioned multi-touch screen, users need to hold their arm in the air for extended periods of time [10].

Some studies have proposed combining a horizontal table-top device with separate vertical displays to implement interactions [13, 3, 14]. A 3D scene is displayed on the vertical display, whereas its 2D view is presented on the table-top device.

In most existing VR applications, 3D scenes are modeled in advance via some modeling system, and are later separately used in the applications. However, with the development of VR technology, new requirements on applications of virtual scene modeling are emerging. Often, the end users are also expected to participate in the scene modeling process. The layout of virtual scenes should be adapted according to their VR experience and feedback. For example, a home design system should not only support the designers in easily, flexibly, and rapidly prototyping the layout of a 3D indoor scene, but should also enable an end user to experience this virtual scene and give the designer the opportunity to alter the layout of furniture in real-time in accordance with the user's feedback from the VR interaction. This approach may lead to improved design results for end users. This is evidenced by some of our previous work on this challenge [16, 17], which presented a new maze game genre combining user-controlled game design in physical space with game play in the virtual space on a mobile device. The system has two interaction modes: "Design" and "Play". In the "Design" mode, the maze is arranged with chalk, ribbons, and other drawing tools according to the user's imagination in the physical world, and then a corresponding 3D virtual maze is auto-generated after the maze on the floor has been captured by a smartphone. In the "Play" mode, a portable VR device and position tracker are used to track users' orientations and positions with respect to the world coordinate frame in order to control viewing parameters and allow for natural interaction.

The ultimate goal of the present study is to construct a design system for 3D virtual scenes on mobile devices, such that it actively involves both designers and laypeople or end users in the design process to help ensure the results meet their needs and that usability aspects have been considered. To address this challenge, this paper proposes a novel genre of participatory design for 3D virtual scenes on mobile devices. As a first exploration in this direction, we developed a 3D virtual scene design system, a user experience-driven participatory design system that has ventured to embed real-time VR experience in the process of designing virtual scenes by capital-

---

* Corresponding authors.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04
http://dx.doi.org/10.1145/3041021.3054173

izing on mobile computing, the web as well as wireless local area networking, motion-tracking, and stereoscopic display technologies.

An overview of the system is provided in Figure 1. It can be run in three ways: (a) Constructing 3D virtual scenes: It supports the designers in constructing an initial 3D virtual scene with two tablets, one of which is used to support the scene assembly in a 2D window, whereas the other synchronously displays the corresponding 3D scene; Certain 3D manipulations are retained for the 3D scenes, while the rest are replaced by 2D interactions in the 2D view in order to increase the design efficiency and user comfort. (b) Optimizing 3D scenes: A user wearing a mobile VR device moves in the physical space to experience the virtual scene and a designer adjusts the layout of objects according to the user's real-time VR experience feedback. The system's dual touch screens provide functions for conveniently monitoring the user's VR experience in the virtual scene. (c) Fine-tuning and optimizing 3D scenes: the designer and user can discuss how to fine-tune and optimize the virtual scene by collaboratively reviewing the surveillance footage recorded by the system from the user's experience process, each interacting on their own respective tablets or smartphones.

The system is portable and easy to learn and operate. It is network-based, and thus can not only be used locally, but also remotely via the network. Our user evaluation shows that our system not only provides a good user experience and improves the design efficiency but also the outcome.

2. RELATED WORK

2.1 Display and Manipulation on Multi-Screens

A number of systems have relied on setups involving multiple screens. Since mobile devices such as watches and smartphones tend to have small screens, exacerbating the "fat finger" problem, multiple connected screens are used jointly as a larger interface [23, 7]. Similarly, in order to better display pictures, videos, or other contents, some studies have used multi-display composition to form a larger logical display. Examples include Paddle [27], JuxtaPinch [25], Lyons' system [24], MobileVideoTiles [21], and so on [32, 31, 35]. However, these approaches all suffer from the disruptive visual appearance at the edges of the devices, which can be hard to ignore.

Some systems need to display different kinds of contents on different screens. BUILD-IT [13] and other systems [14, 2] display one view of the tasks on the horizontal screen and another view on the vertical screen or wall. To explore the importance of a vertical screen orientation correlating with a table-top device, Wigdor et al. built a multi-display collaborative environment with large vertical displays surrounded by numerous desktop computers [38, 39]. Their research results imply that users consider factors other than the performance when establishing their preferences and that physical comfort may be more significant than their preference. Their results emphasize the importance of the vertical screen orientation with respect to the horizontal screen. Forlines et al. [14] combine a planar view on a table-top device with 3D views on several vertical screens for a geo-spatial application. Unlike our system, the planar view suffers from perspective issues and it does not provide holographic displays. All of these systems that combine horizontal and vertical screens mainly place the manipulations on the horizontal screens.

2.2 Participatory Design with VR Experience

Good et al. [18] remarked that VR can facilitate the use of participatory design in architecture. A simple step that they indicated is to have multi-participant walkthroughs, where a user and architect explore a VR space together, and the comments of the user are recorded for future use. Thus, the VR system becomes a design tool, not just a visualization tool, because users are able to propose changes to the designs within the system.

Chu et al. [9] developed a VR-based system for users who lack expertise in using industrial CAD systems to design conceptual shapes. The authors focused on determining the requirements for multi-sensory user interfaces and assessing the applications of different input and output mechanisms in virtual environments, but they did not rely on users' VR experiences to improve the design. Co-Star [28] is an immersive stereoscopic system for cable harness design and enables cable harnesses to be designed by using direct 3D user interactions with a product model. The system obtains a user's VR experience via a questionnaire and interview. Unlike our system, this sort of VR feedback is not real-time.

Thalen et al. [36] pointed out that VR techniques can provide an intuitive, integral, and interactive representation of future application situations. They also presented several case studies with user involvement, such as a driver assistance system and an operating theatre design. They suggest that a system's design should involve users throughout the design process, i.e., design for, with, and by end users.

Fukuda et al. [15] presented a participatory design approach using VR and a blog as design media in a design process.

2.3 Player-Centered Design

Player-centered design is an approach taken in game development to improve the game from the perspective of individual players [34]. Taking a such a design approach may enhance the game-play experience regardless of gender, age, or experience [6].

There are several traditional approaches to implement a player-centered design. Ermi [11] provides players with a number of different game-play scenarios in a comic strip format. The users' opinions are analysed and used to establish the design requirements. However, Charles et al. [6] argue that this approach fails to guarantee the quality of the collected information and may lead to an overly narrow focus on the desires of specific players, neglecting other kinds of players. Schaefer et al. [30] take the approach of conducting play testing in pre-release versions of the game for a player-centered design. With this approach, secrecy can be an issue, as game concepts leak to the public too early [34].

In addition, there are some newer approaches to implementing a player-centered design that are taking root in the game development industry [34]. These include modeling player types and using adaptive game systems. Charles et al. [6] enumerate approaches to account for different player types as follows. The first is to collect significant amounts of information on players during game development and develop specifically for game-play styles based on this information. Thus, designers obtain feedback from users in the early design stages, which assists in modeling player types. The second approach is referred to as "adaptability by emergence": A designer constructs a game, in which players can choose their own style of achieving the objective of game. The RPG game *Heretic Kingdoms: Reluctant Hero* [4], for instance, has adopted this sort of adaptability by emergence as a design philosophy. Finally, with the approach of using adaptive game systems, a game adjusts itself over time based on input and measurements from individual players, as is the case for *Max Payne* [1].

2.4 Manipulation based on Multi-Touch Interaction on Portable Devices

Multi-touch devices provide a higher degree of freedom than mouse pointing devices, and in some instances mobile device-based multi-touch interfaces have enabled direct manipulation of 3D objects [12].

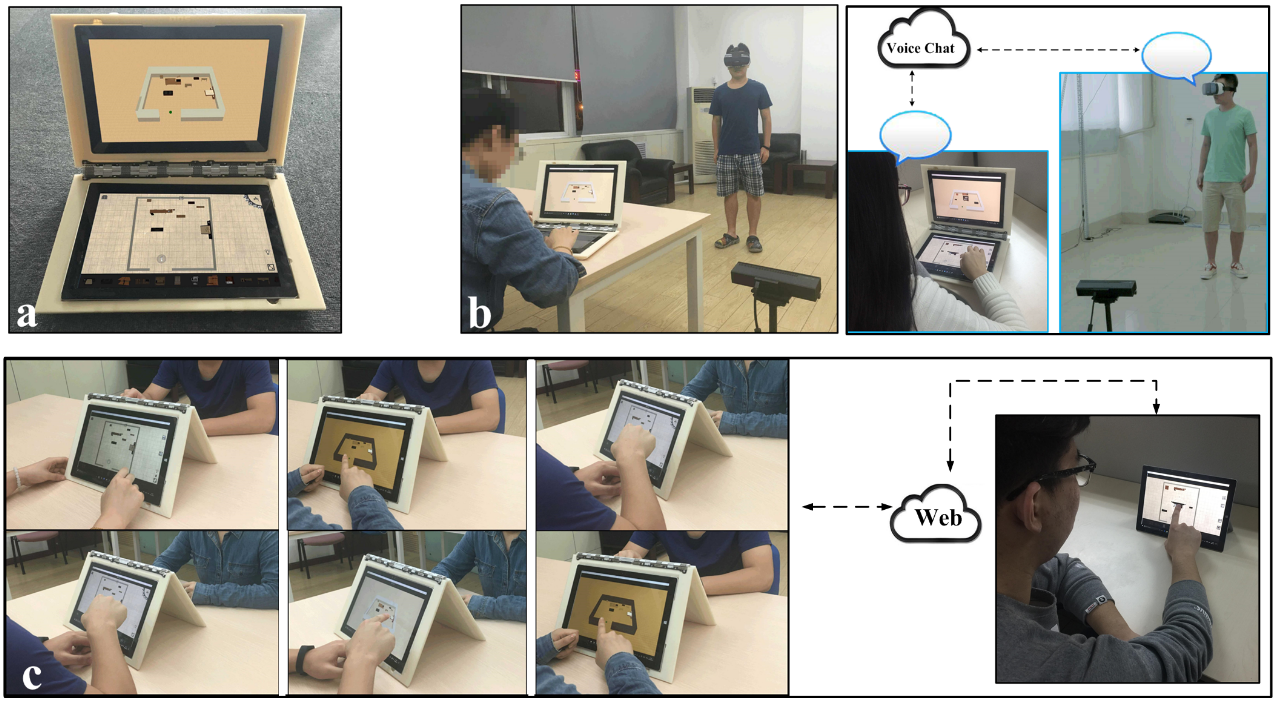


Figure 1: Our system as a laptop with two touch screens: (a) Constructing 3D Scenes: The scene is assembled in 2D on the horizontal screen and synchronously displayed in 3D on the vertical screen; (b) Optimizing 3D Scenes: The scene is adjusted by the designer based on the user's VR experience, which supports both LAN (left side) and remote operation (right side); (c) Fine-tuning 3D Scenes: The screens can be rotated by 270 degrees so that the designer and the user can collaboratively optimize and fine-tune the 3D scene in 2D/3D interfaces face-to-face by replaying the surveillance footage of the user's experience; others who have a tablet can also join in via the web.

Qiu et al. [26] propose a simple 3D modeling method that relies on portable pressure-sensing multi-touch devices. Such prior works, however, have limitations with regard to their ability for 3D content to be manipulated efficiently on a mobile device. Sankar and Seitz [29] proposed a smartphone application designed to capture, visualize, and reconstruct homes, offices, and other indoor scenes. It relies on data from smartphone sensors such as camera, accelerometer, gyroscope, and magnetometer to help model indoor scenes, thereby enabling users to create immersive tours on a smartphone. Another study [33] facilitated 3D interior design based on mobile true-3D displays. Still, the above systems remain too complicated and inconvenient to build non-trivial 3D virtual scenes.

3. SYSTEM ARCHITECTURE

Figure 2 provides an overview of the system architecture. The architecture of our system is network-based with a web-based server for data exchange. It can be used locally via a wireless router, or remotely to enable communication with users that are not in the same place as the designer.

For the user's setup, the requirements include a Head-Mounted Display (HMD), a Kinect device, and a computer that can connect with the Kinect via USB. Previous work [19, 8] shows the wide applicability of Kinect for location tracking. The Kinect device monitors the data of a user's real-time position, and the computer sends this data to the server via the web. The HMD receives its data via the web to display relative changes in the virtual environment. Furthermore, the data includes the user's position and the information of objects changed by the designer. At the same time, the HMD sends a user's rotation Euler angles to the web server, while a voice chat software integrated with the HMD supports communication between the user and designer for optimizing the scene.

The designer's setup involves two smart tablets or a phone that can run our design software. Considering the limited screen size of smartphones, it is typically preferable to rely on dual tablet-sized touch screens. While designing the scene, one screen displays a 2D view of the 3D scene, receives the user's position, and sends the manipulation data for the current object to the web server. Correspondingly, the other screen, which displays a 3D view of the scene, receives the user's position and information pertaining to the object manipulation.

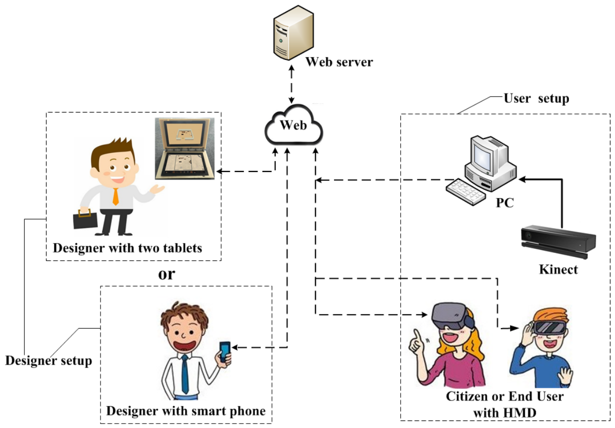


Figure 2: System architecture.

4. SYSTEM DESIGN AND IMPLEMENTATION

We aim to develop a simple, portable, and participatory design system for 3D virtual scenes to support the designer in flexibly and rapidly constructing 3D virtual scenes. Therefore, a sort of laptop setup, consisting of dual touch screens, is used as a platform for operating the system. A vertically oriented screen is used to display the scene in 3D and the other horizontal one provides a 2D view of the scene (Figure 1a). The system provides a novel means of constructing 3D virtual scenes.

It can be run in three ways. In Section 4.1, we discuss the process of constructing 3D virtual scenes, which involves the designer interacting with a 2D window on the horizontal screen, while a 3D window synchronously displays the corresponding 3D scene (Figure 1a). In Section 4.2, we discuss the user-based optimization of a 3D scene, in which a user wearing a mobile VR device moves in the physical space to experience the virtual environment, while the designer adjusts the layout of objects according to the user's real-time VR experience feedback (on the left side of Figure 1b). The system also supports the case of a user communicating with the designer via voice chat when they are not in the same place (on the right side of Figure 1b). During this phase, the system records the user's experience with virtual cameras to allow for replaying the footage in the next stage. In Section 4.3, we describe the fine-tuning process. The designer and user, assuming they are in the same place, can sit face to face to replay the footage and discuss how it can be fine-tuned, each interacting on their own screen. For this, the angle between two screens can be adjusted to 270 degrees to facilitate more natural face-to-face communication and cooperative work, while also providing for a comfortable professional distance between the two participants. In addition, another designer or user can also join the discussion remotely via web (Figure 1c).

4.1 Constructing 3D Virtual Scenes

The basic process of 3D scene construction consists of: (1) Setting the terrain according to the script of the game or animation; (2) Determining the boundaries of the set, e.g. by establishing the walls for an indoor scene; (3) Placing virtual objects at available locations within the virtual scene and manipulating objects via rotations, scaling, and translations; (4) Roaming in accordance with a specified route or observing the scene from arbitrary angles through the scene camera; (5) Repeating the above steps until the scene complies with all relevant requirements.

4.1.1 Set Terrain

Virtual scenes can be divided into indoor scenes and outdoor scenes. For most indoor scenes and many outdoor scenes, the terrain is usually flat and without fluctuations. In our prototype, we therefore set the terrain by providing the texture of the ground. There are some common built-in textures in our system, such as wooden floors, stone floors, and grass. If uneven terrain is needed, it can be imported to the system as an object and one can set this terrain by manipulating it as an object.

4.1.2 Drawing Walls

For general scenes, the height of the wall is usually fixed, and the walls are vertically adjoined to the ground. As a result, drawing walls only requires data on the X and Z axis, which can be accomplished via the canvas in the 2D window and is inspired by the method in [22]. Based on this, the corresponding full 3D wall is automatically rendered in the 3D window. All available operations, such as adding, deleting, modifying, and so on, are shown in a box at the top of the 2D interface. Figures 3 through 5 depict the operations of modifying a wall, adjusting a curve and deleting a wall. The functions in Figure 3 and Figure 4 are implemented via drag-and-drop operations. Figure 5a and b show how to delete a wall, while Figure 5c and d show how to delete several adjacent walls in one fell swoop. Both the functions in Figure 5 are implemented by a simple touch operation.

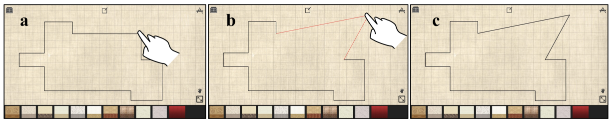


Figure 3: (a) Touching the points that need to be modified without lifting the finger. (b) Dragging the point to the desired target location. (c) Lifting the finger to leave the screen and the two adjacent segments connected to the point will be updated.

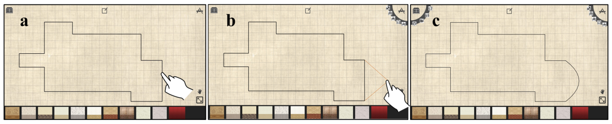


Figure 4: (a) Touching the line segment that needs to become curved and not lifting the finger. (b) Dragging the control point to a desired location. (c) Lifting the finger such that the original line is replaced by a curve.

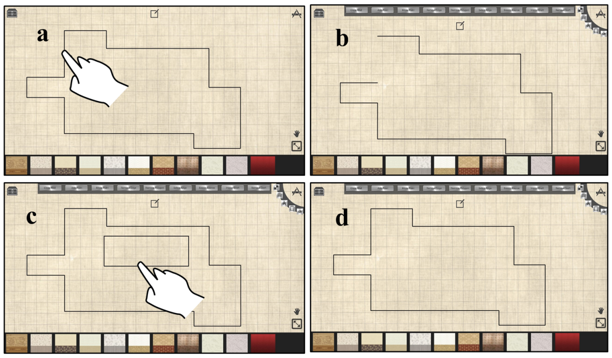


Figure 5: (a) Touching the line segment that needs to be deleted. (b) A line segment has been deleted and the next drawing will start from the deleted segment's starting point. (c) Touching any segment in the polygon that needs to be deleted. (d) Several walls of the polygon have been deleted.

4.1.3 Object Manipulation

This system is mainly used for the layout of scenes in which objects are placed on the ground in a vertical position. In this case, objects do not need to be rotated around any axis except the Z-axis. Thus, the object manipulation operations and gestures consist of: translation along the X, Y, Z axis and rotation around the Z axis. As the plane of the 2D canvas is equivalent to the level ground of the 3D space, and object manipulation is very frequent, the object manipulations are assigned to the 2D screen. But in this case, users cannot move objects along the Z axis of the 3D space directly on the 2D canvas. In order to solve this problem, we use a vertical panel with a scroll bar to achieve a translation of the Z axis, and a transparent circular horizontal panel providing a reading of the angle to control the rotation of the object around the Z axis. As shown in

Figure 6, this object manipulation menu appears for long-presses of an object on the 2D canvas with one finger. If users select "adjust height", the Z coordinate can be modified by moving the slider on the vertical panel. Similarly, the object rotates around the Y axis when users move the slider on the horizontal panel. With regard to the object translation operation in the X-Z plane of the 3D space, users only need to select objects on the 2D design canvas and drag them to a given location (Figure 7). For the translation and rotation along the Y axis, we use the panel to assist the operation. The rationale for this is multi-fold. Firstly, it avoids the need for the user to memorize complex multi-touch gestures. Secondly, manipulating objects in the 2D window and previewing the effect on the 3D screen avoids the blocking and misoperation caused by big fingers. Thirdly, it enables the system to not only be used for organic scenes [20], but also to improve the accuracy of object manipulation in artificial scenes, such as hanging a picture on the wall.

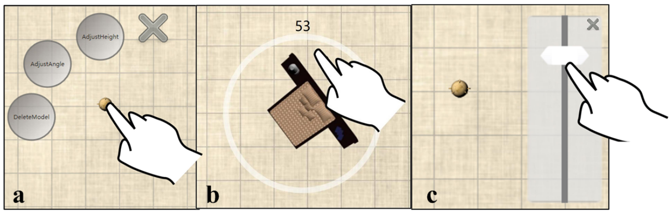


Figure 6: (a) Long-press an object and the object manipulation menu appears. (b) The transparent circular panel controls the angle of an object. (c) The vertical panel controls the height of an object.

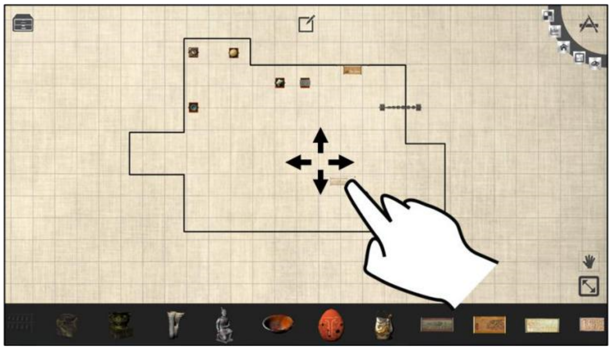


Figure 7: Selecting an object and dragging it to translate.

4.1.4 Camera Control

In our system, we bind the gestures for camera control to the 3D scene. Thus, simple gestures can be reused, avoiding the complexity and potential misoperation caused by switching modes and preventing the gesture inventory from becoming too overwhelming to learn. Taking into account the fact that the walls have a certain height, the camera's complete parallelism to the ground direction would lead to obstructed views, so the initial direction of the camera is instead from an inclination at the top of the scene. The system offers camera control operations, including movement along the X axis and Y axis, moving the camera forward and back, as well as rotations around the Z axis. As shown in Figure 8a, the camera is controlled on the X axis and Y axis by means of dragging any area of the 3D window with a single finger. We use the pinch gesture to steer the camera (dolly) forwards and backwards. This involves placing a finger on any area of the 3D window and rotate another finger around the first touching point (Figure 8b). As shown in Figure 8c, the angle of rotation is the angle of the camera rotation. As both the camera dolly and the rotation around the Z axis are two-finger operations, there is an issue about how to distinguish these two operations. To solve this issue, we set a rotation angle threshold as the cut-off point. If the rotation angle of the multi-touch gesture is less than the threshold, the operation will be considered as a dolly movement, otherwise as a camera rotation.

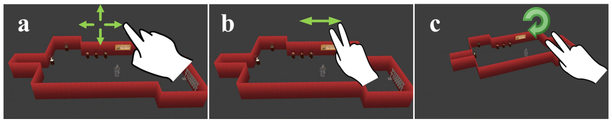


Figure 8: (a) One finger touches the 3D interface and drags to translate the camera. (b) Users control the camera's forward and backward movements (dolly) via a pinch gesture. (c) The camera can be rotated by using one finger rotating around another finger touching the screen.

4.1.5 Adding Objects and Changing Textures

The system provides three methods to add an object. First, users can select a model from the model library in the 2D interface, and place it on the 2D canvas via a drag and drop operation (Figure 9). This method is suitable for scenes that depend on fewer models. However, one often needs to place a large number of identical objects in a scene, such as when designing a grove of trees with the same plant model. Inspired by the throw-and-catch technique [20], we have incorporated a second method to add objects: One finger selects an object in the model library and holds it, and another finger taps on the screen to specify destinations (Figure 10). Such touch operations thereby enable rapid placement of the same model and save time and avoid the distance of having to move the same object repeatedly. There is another common circumstance. When constructing a large scene consisting of several small scenes, designers often need to focus on a small area adding many different models to it. For this, we provide a third method to add objects: One finger selects a closed area, and another finger taps on different objects in the model library (Figure 11). This way, users can quickly add many different models to the same scene area. Similarly, the texture of the wall, ceiling, and floor can also be customized by selecting the corresponding object in the library and dragging it to the 2D canvas. The 3D window reflects any changes of the textures instantaneously.

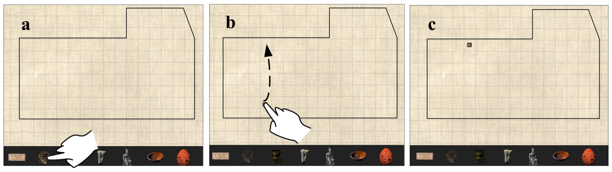


Figure 9: (a) Selecting an object in the bottom model library. (b) Dragging it to desired target location. (c) Lifting the finger.

4.1.6 3D Roaming

The system provides 3D Roaming capabilities for designers to observe and have a look around the scene from different angles to preview the scene being designed. With regard to the scene plan in the 2D window, we rely on a footprint icon to represent the camera's current position on the 2D canvas, and the position of the observer in the 3D scene is controlled by 2D touch interactions. Specifically,

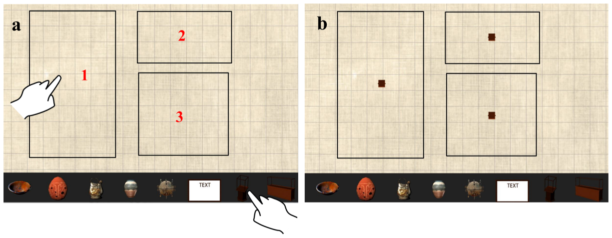


Figure 10: (a) One finger selects an object in the model library and holds it, and another finger taps on the screen to specify destinations, such as the areas represented as “1", “2", and “3". (b) The same model will appear in several different areas.

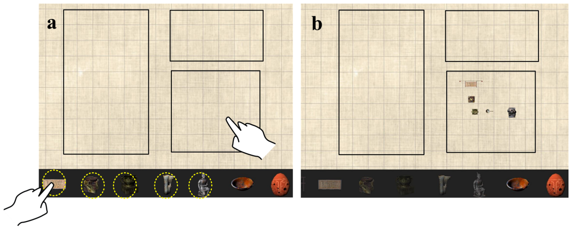


Figure 11: (a) One finger selects a closed area, and another finger is used to select several objects in the model library. (b) In rapid succession, different models can be added to the same area within the scene.

users use one of their fingers to drag the footprint icon within the 2D window to move the camera, while another finger can be used to drag on the 3D window interface to control the orientation of the camera. When the orientation of the camera is adjusted in the 3D window, the camera position in the 2D window is kept unchanged (Figure 12).

Figure 13 provides examples of a virtual museum, a home interior decoration scene, and a shooting game scene, all designed using our system. The designed scenes can not only be saved in different file formats, but, importantly, can be used for immersive VR experiences based on Unity.

4.2 Participatory Design to Optimize 3D Scenes

The system also supports designing scenes based on user VR experience feedback. While a user wearing a mobile VR device (as shown in Figure 1b) is moving in the physical space to experience a preliminary version of the scene, a designer sitting in front of the laptop with double screens can adjust the layout of objects by observing the user’s movement. In the process, the designer can see a 2D overview of the scene on the horizontal screen as well as the 3D view of the scene from the user’s viewpoint on the vertical screen. At the same time, the system records the user’s trajectory and scene adaptation information for playback and to facilitate discussions about how to fine-tune the virtual scene later. Additionally, the system places several fixed virtual cameras at key locations, as shown in Figure 14. The designer can select a camera view with a button placed in the bottom of the 2D window and then the 3D window will show the user’s movements and manipulations from this perspective. In this case, the designer can also adjust the layout of objects near the user by observing the user’s movements. All of these cameras also record the user’s movements and manipulation information.

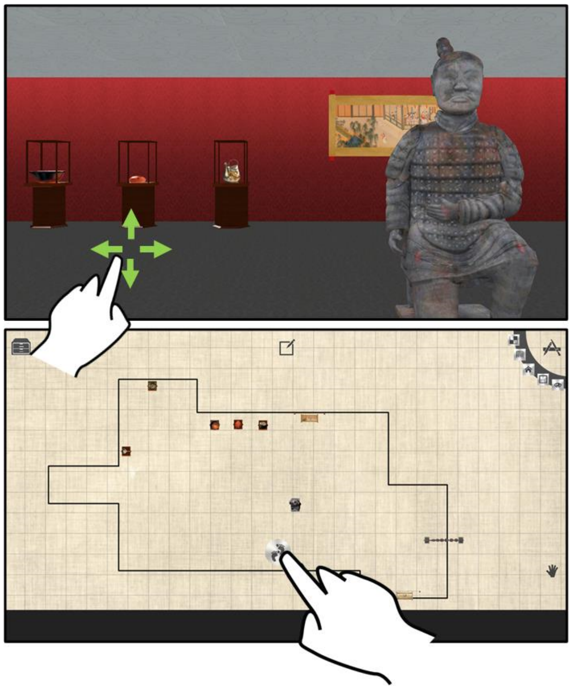


Figure 12: One finger drags the footprint icon in the 2D window to move the camera, while another finger drags within the 3D window interface to control the direction of the camera.

4.3 Fine-Tuning 3D Scenes by Replaying Recordings of the User Experience

For more convenience during the subsequent discussions between the designer and the user, we designed the rotation axis of the laptop hardware to support the screens rotating at $0 - 360$ degrees. Hence, the angle between the two screens can be rotated to 270 degrees, as shown in Figure 1c. The system adjusts the screens’ orientation according to the angles, as reported by sensors in the laptop. This setup is well-suited for enabling two users to sit face-to-face and discuss how to further optimize and fine-tune the scene by reviewing the surveillance video footage. What each participant sees can be either the 2D window or the 3D window, as shown in Figure 1c. With this setup, the system can support two people’s cooperative work. Since they do not need to sit side-by-side, face-to-face communication is enabled and the participants do not need to crowd around and operate a single small screen. This setup supports not only discussions during the design phase but also reviews of user VR experiences. In addition, other users can join this review process via the web. In virtue of this, the system thus has the potential to improve the efficiency of the design process as well as the designs resulting from it.

5. USER STUDIES

To empirically verify the effectiveness of our system, we conduct two formal user studies. Study A focuses on the usability of our design system in constructing 3D virtual scenes. It includes a usability test, evaluating the usability and ease of use of our design system. Study B focuses on the advantages of the VR experience feedback for participatory design. It includes a comparative experiment, examining whether the real-time feedback based on the VR experience can improve the system’s efficiency and user experience.

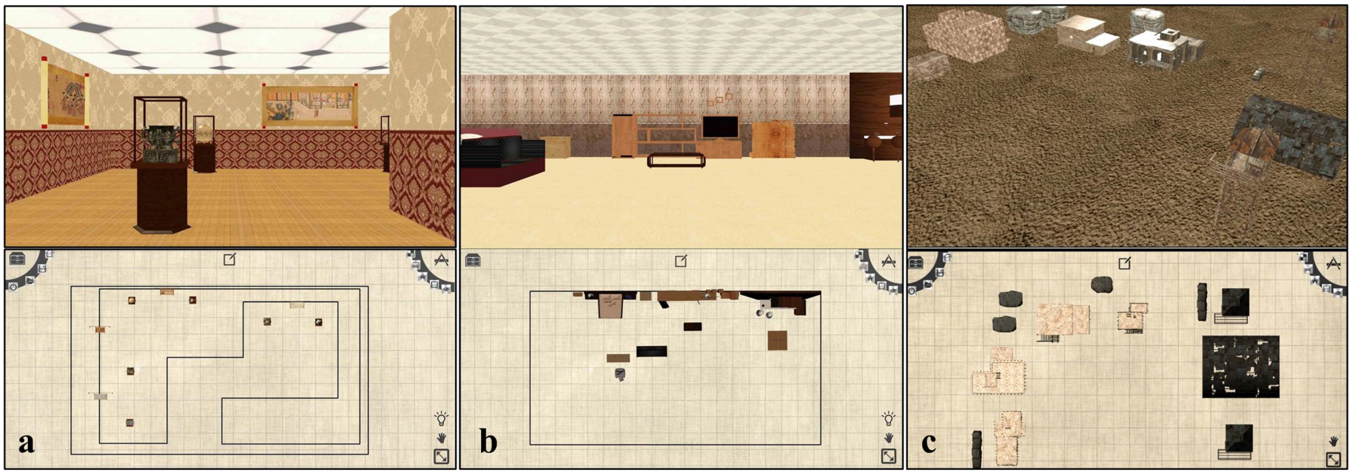


Figure 13: Virtual museum, home decoration, and shooting game scenes designed using our system.

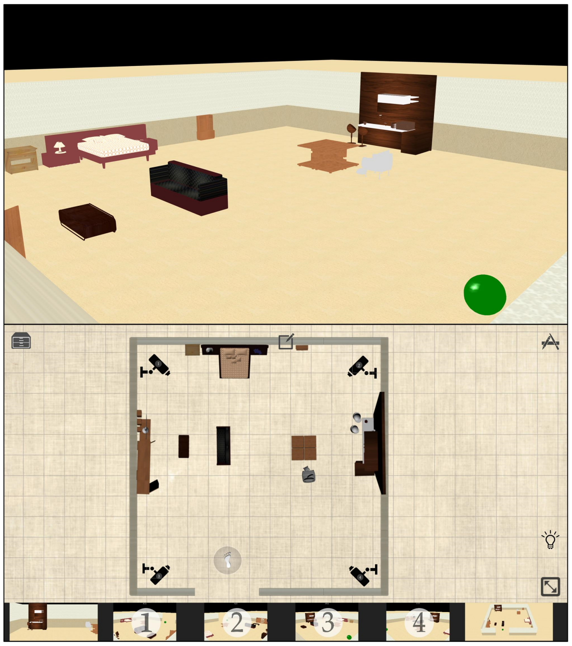


Figure 14: Fixed virtual cameras at significant locations to record the information of user's movement and manipulations.

5.1 Study A

5.1.1 Participants

A total of 25 undergraduate volunteers (15 male, 10 female) were recruited to participate in the study. The age of the sample ranges from 20 to 28 years ($M = 22.58$ years, $SD = 2.17$ years).

5.1.2 Experimental Task and Procedure

The task proceeds as follows: 1. The participants are given instructions for using our design system. 2. They then experience the system and practice to use various functions of the system (including adding and removing objects, moving and rotating objects, changing object properties, etc.). 3. Next, an empty virtual room is provided, and participants are requested to add furniture to the room and design the interior to their liking. 4. Finally, after completing the design, the participants are asked to complete an online questionnaire.

5.1.3 Measures

To examine the usability of our design system, we measure perceived usefulness (PU) and perceived ease of use (PEU) based on the Technology Acceptance Model.

5.1.4 Results

We rely on two descriptive statistics to describe the scores of the two indicators. The results in Figure 15 show that the scores are all higher than 6 (representing a good reference score), which suggests both a high perceived usability and high ease of use.

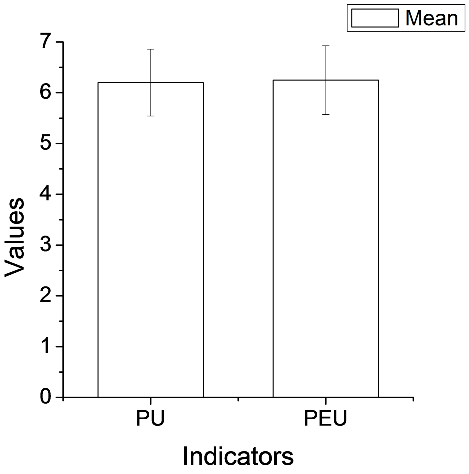


Figure 15: Descriptive statistics of PU and PEU.

5.2 Study B

5.2.1 Experimental Design

This study includes an experiment that adopts a single-factor within-subject experimental design. The within-subject factor is the design mode, which considers two different options: real-time

Table 1: Measures of dependent variables.

Dependent variables	Indicators	Measures
Design efficiency	Objective indicators	Modification times
		Valid design time
	Subjective indicators	Satisfaction in first design
		Satisfaction in final design
User experience	Enjoyment	The design mode is interesting, and my curiosity is stimulated
		The design mode keeps me happy during my task
	Satisfaction	I think the design system is good
		I am satisfied with this design system
	Behavior intention	I intend to use the system to design my room
		I'd be happy to use this design system again
	Immersion	I'm totally immersed in the design process
		I feel like I'm in this virtual space

Table 2: Descriptive statistics results of user experience.

Aspect	Mode A	Mode B
Enjoyment	13.08 (1.52)	12.44 (1.87)
Satisfaction	13.36 (0.81)	12.28 (1.79)
Behavioral Intention	13.20 (1.00)	12.28 (2.19)
Immersion	12.72 (1.28)	11.76 (2.42)

feedback for VR experience (Mode A) and non-real-time feedback (Mode B). The dependent variables include the design efficiency and user experience. The influence of different modes' presentation order in each experiment is counterbalanced.

5.2.2 Participants

A total of 25 undergraduate volunteers (13 male, 12 females) are recruited to participate in the experiment. The age of the sample ranges from 21 to 29 years ($M = 23.84$ years, $SD = 2.23$ years).

5.2.3 Experimental Task

The participants are asked to design a room in two different modes (Mode A and Mode B). The objective in both cases is the same: Each participant needs to place 8 pieces of furniture in a virtual room provided by the system. To avoid that participants repeat the same design in the second mode, the 8 pieces of furniture are not entirely the same.

In Mode A, the participants design the room entirely in the VR environment. They just need to speak out how to move and place the 8 pieces of furniture, while a researcher manipulates the furniture following their instructions via the combined 2D and 3D interface of our system. The participants can observe all operations in real-time, and then express new requests. In Mode B, the participants design the room and move the furniture in the 2D and 3D interface by themselves. After that, they put on VR glasses to experience a simulation of their own designed room.

5.2.4 Procedure

The procedure is as follows. 1. First, the researcher shows the experimental instructions to the participants. 2. Then, each participant is asked to design a room with design mode A or B. 3. When participants have completed the room design, they need to experience the room also in either the 2D, 3D, or the VR interface, and evaluate their satisfaction with the room design. 4. A given participant then may elect to modify the room design again. If unsatisfied with the design, they can go back to the application, modify the design and repeat the process until they are satisfied with it. The researcher records the self-rated satisfaction with the design before every modification. 5. Then, the participant is asked to design another room design with the other design mode. 6. Finally, the participant is asked to complete an online questionnaire.

Moreover, the researcher records the modification times, valid design time of each participant in the two modes.

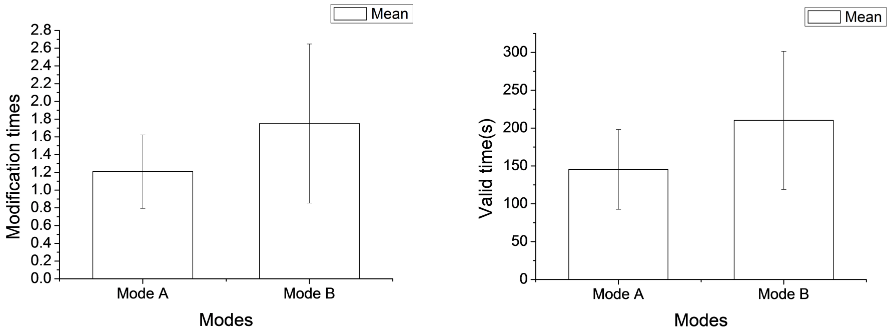
5.2.5 Measures

Design efficiency is measured with 2 subjective indicators and 2 objective indicators. The user experience is measured with 4 indicators, namely enjoyment, satisfaction, behavior intention, and immersion. These indicators and the corresponding measures are presented in Table 1.

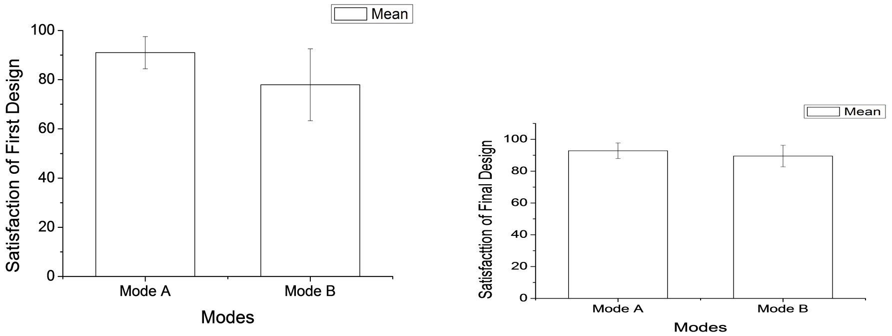
5.2.6 Results

We analyze the results in two steps, beginning with the design efficiency and then considering the user experience.

Difference Tests for Design Efficiency in Different Modes. The values of the two objective indicators (modification times and valid design time) and the two subjective indicators (satisfaction for first design and final design) in different interaction modes are given in Figure 16. To further investigate the differences of these indicators in different interaction modes, four paired-samples T-tests are conducted. The results reveal that the difference between Mode A and Mode B on modification times ($t = 2.982$, $p = 0.006 < 0.01$), valid design time ($t = 4.990$, $p < 0.001$), satisfaction in first design ($t = 5.074$, $p =< 0.001$), and satisfaction in final design ($t = 2.787$, $p = 0.010 < 0.05$) are all significant. Compared with Mode B, Mode A costs participant less modification time, and less



(a) Modification times (b) Valid design time



(c) Satisfaction with first initial design (d) Satisfaction with final design

Figure 16: Descriptive statistics results of design efficiency, each comparing Mode A and Mode B.

design time. This means that participants can design a satisfying room faster and smoother in Mode A compared to Mode B. Moreover, compared with Mode B, Mode A leads to higher satisfaction in both the first design and the final design. This means that Mode A not only helps to achieve a better first design than Mode B, but also contributes to achieving a better final design, although more repeated modifications are made in Mode B. Hence, overall, the results in Figure 16 show that the real-time feedback mode for VR experience has a higher efficiency.

Difference Tests for User Experience in Different Modes. The values of four user experience indicators in different interaction modes are shown in Table 2. To further investigate the differences of these indicators in different interaction modes, four paired-samples T-tests are conducted. The results reveal that the difference in enjoyment ($t = 2.179$, $p = 0.039 < 0.05$), satisfaction ($t = 3.260$, $p = 0.003 < 0.01$), and behavior intention ($t = 2.623$, $p = 0.015 < 0.05$) between Mode A and Mode B are all significant. The difference for immersion ($t = 2.039$, $p = 0.053 > 0.05$) is marginally significant. This means that participants appear to derive more pleasure and fun when using mode A compared to mode B. When designing a room with Mode A, they are more immersed in the design process and more satisfied with it. Finally, they prefer Mode A for designing their own rooms in the future. Overall, the above results show that the VR experience real-time feedback mode leads to a better user experience.

6. CONCLUSION

This paper presents a user experience-driven design genre for 3D virtual scenes based on a laptop-style device with dual touch screens. Based on this genre, we developed a simple, portable, and participatory 3D virtual scene design system that aids the designer in flexibly and rapidly creating virtual scenes according to an end user's VR experience feedback. The system is portable and easy to learn and use, and can also be used remotely via a network. Our user evaluation shows that such a setup is rated as having a high usability and that it improves the efficiency of the design process as well as the satisfaction with the resulting designs. This study opens up compelling new avenues to further explore this new space of user interfaces based on multiple screens for participatory design. We will continue to explore which kinds of operations are better-suited for 2D vs. 3D viewing to yield simpler and more comfortable user interactions. Additionally, we are working to apply our system to a diverse range of applications, including firefighter training, anti-terrorism drills, among others.

7. ACKNOWLEDGMENTS

We would like to thank all reviewers for their valuable comments. This work is supported by the China National Key Research and Development Project (2016YFB1001403), the National Natural Science Foundation of China (under Grants 61472225, 61272243, 61503217), and the Shandong Provincial Science and Technology Development Program (2016GGX106001, 2016ZDJS01A09).

8. REFERENCES

[1] Max Payne. `https://en.wikipedia.org/wiki/Max_Payne`.

[2] D. Aliakseyeu, S. Sriram, J.-B. Martens, and M. Rauterberg. Interaction techniques for navigation through and manipulation of 2 d and 3 d data. In *EGVE*, volume 23, pages 179–188. Citeseer, 2002.

[3] G. Bailly, L. Nigay, and D. Auber. Navrna: visualization-exploration-editing of rna. In *AVI*, pages 504–507. ACM, 2006.

[4] C. Bateman. Heretic kingdoms: Reluctant hero - designer diary. `https://www.rpgwatch.com/show/article?articleid=9&ref=0&id=111`.

[5] R. Chang, F. Wang, and P. You. A survey on the development of multi-touch technology. In *APWCS*. IEEE, 2010.

[6] D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kcklich, A. Moore, and K. Stringer. Player-centred game design: Player modelling and adaptive digital games. In *Proceedings of the digital games research conference*, volume 285, page 00100, 2005.

[7] X. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice. Duet: exploring joint interactions on a smart phone and a smart watch. In *SIGCHI*, pages 159–168. ACM, 2014.

[8] Y. Chen, H. Yu, C. Miao, B. Chen, X. Yang, and C. Leung. Using motor patterns for stroke detection. `http://or.nsfc.gov.cn/handle/00001903-5/338026`.

[9] C.-C. P. Chu, T. H. Dani, and R. Gadh. Multi-sensory user interface for a virtual-reality-based computeraided design system. *Computer-Aided Design*, 29(10):709–725, 1997.

[10] I. Efanov and J. Lanir. Augmenting indirect multi-touch interaction with 3d hand contours and skeletons. In *CHI*, pages 989–994. ACM, 2015.

[11] L. Ermi and F. Mäyrä. Player-centred game design: Experiences in using scenario study to inform mobile game design. *Game Studies*, 5(1):1–10, 2005.

[12] D. Fiorella, A. Sanna, and F. Lamberti. Multi-touch user interface evaluation for 3d object manipulation on mobile devices. *Journal on Multimodal User Interfaces*, 4(1):3–10, 2010.

[13] M. Fjeld, M. Bichsel, and M. Rauterberg. Build-it: a brick-based tool for direct interaction. *Engineering Psychology and Cognitive Ergonomics (EPCE)*, 4:205–212, 1999.

[14] C. Forlines, A. Esenther, C. Shen, D. Wigdor, and K. Ryall. Multi-user, multi-display interaction with a single-user, single-display geospatial application. In *UIST*, pages 273–276. ACM, 2006.

[15] Fukuda, Tomohiro, Kaga, Atsuko, Izumi, Hideaki, Terashima, and Takanori. Citizen participatory design method using vr and a blog as a media in the process. *International Journal of Architectural Computing*, 7(2):217–233, 2009.

[16] W. Gai, C. Yang, Y. Bian, M. Dong, J. Liu, Y. Dong, C. Niu, C. Lin, X. Meng, and C. Shen. Ubimaze: a new genre of virtual reality game based on mobile devices. In *MobileHCI*, pages 591–593. ACM, 2016.

[17] W. Gai, C. Yang, Y. Bian, C. Shen, X. Meng, L. Wang, J. Liu, M. Dong, C. Niu, and C. Lin. Supporting easy physical-to-virtual creation of mobile vr maze games: a new genre. In *SIGCHI*, page to appear. ACM, 2017.

[18] M. Good and L. Tan. Vr in architecture: today's use and tomorrow's promise. *Virtual Reality World*, 1994.

[19] M. Huang, Y. Chen, W. Ji, and C. Miao. Accurate and robust moving-object segmentation for telepresence systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2):17, 2015.

[20] K. Kin, T. Miller, B. Bollensdorff, T. DeRose, B. Hartmann, and M. Agrawala. Eden: A professional multitouch tool for constructing virtual organic environments. In *SIGCHI*, pages 1343–1352. ACM, 2011.

[21] M. Li, K. M. Scharf, and L. Kobbelt. Mobilevideotiles: video display on multiple mobile devices. In *MobileHCI*, pages 621–626. ACM, 2016.

[22] Y.-J. Liu, C.-X. Ma, and D.-L. Zhang. Easytoy: plush toy design using editable sketching curves. *IEEE Computer Graphics and Applications*, 31(2):49–57, 2011.

[23] K. Lyons, D. Nguyen, D. Ashbrook, and S. White. Facet: a multi-segment wrist worn system. In *UIST*, pages 123–130. ACM, 2012.

[24] K. Lyons, T. Pering, B. Rosario, S. Sud, and R. Want. Multi-display composition: Supporting display sharing for collocated mobile devices. In *INTERACT*, pages 758–771. Springer, 2009.

[25] H. S. Nielsen, M. P. Olsen, M. B. Skov, and J. Kjeldskov. Juxtapinch: exploring multi-device interaction in collocated photo sharing. In *MobileHCI*, pages 183–192. ACM, 2014.

[26] S. Qiu, L. Wang, and L. Wong. Pressure-based touch positioning techniques for 3d objects. In *SIGGRAPH*, pages 199–200. ACM, 2016.

[27] R. Ramakers, J. Schöning, and K. Luyten. Paddle: Highly deformable mobile devices with physical controls. In *CHI*, pages 2569–2578. ACM, 2014.

[28] G. Robinson, J. M. Ritchie, P. N. Day, and R. G. Dewar. System design and user evaluation of co-star: An immersive stereoscopic system for cable harness design. *Computer-Aided Design*, 39(4):245–257, 2007.

[29] A. Sankar and S. Seitz. Capturing indoor scenes with smartphones. In *UIST*, pages 403–412. ACM, 2012.

[30] E. Schaefer. Postmortem: Blizzard's Diablo II. `http://www.gamasutra.com/view/feature/131533/postmortem_blizzards_diablo_ii.php`.

[31] A. Schmitz, M. Li, V. Schönefeld, and L. Kobbelt. Ad-hoc multi-displays for mobile interactive applications. In *Eurographics*, volume 29, page 8, 2010.

[32] J. Schüler and S. Engeser. Incentives and flow experience in learning settings and the moderating role of individual differences. *Contemporary motivation research: From global to local perspectives*, pages 339–357, 2009.

[33] M. Serrano, D. Hildebrandt, S. Subramanian, and P. Irani. Identifying suitable projection parameters and display configurations for mobile true-3d displays. In *MobileHCI*, pages 135–143. ACM, 2014.

[34] J. Stewart. Different approaches to implementing player centered game design: A comparison study.

[35] P. Tandler, T. Prante, C. Müller-Tomfelde, N. Streitz, and R. Steinmetz. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In *UIST*, pages 11–20. ACM, 2001.

[36] J. P. Thalen. *Facilitating user centred design through virtual reality*. PhD thesis, 2013.

[37] L. Wang, H.-s. Zhao, W. Chen, C.-l. Yang, and X.-x. Meng. O3D-based personal museum designing system in virtual learning environment. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(1):55–71, 2011.

[38] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Effects of display position and control space orientation on user preference and performance. In *SIGCHI*, pages 309–318. ACM, 2006.

[39] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Table-centric interactive spaces for real-time collaboration. In *AVI*, pages 103–107. ACM, 2006.