Design Guidelines for Multiplayer Video Games on Multitouch Displays

Oct 30, 2009
Electronic Visualization Laboratory
University of Illinois at Chicago

1 Introduction

Multi-touch displays are touch-sensitive displays that allow users to interact by using their fingers. These displays differ from traditional touch-screens commonly used in ATM machines and other consumer devices in that they are able to register a large number of distinct touch points (typically in the tens) simultaneously. This allows users to interact with the display using more than one finger or hand, which expands the possible interaction schemes and enhances user experience. Additionally, a multi-touch screen allows multiple users to simultaneously interact with the display when the screen surface is large enough to accommodate them. This enables possibilities for a new generation of collaborative, multi-user, interactive applications.

The enhanced user experience has made multi-touch displays an attractive option for many applications in which multiple people need to collaborate to achieve a common goal. For example, museums are exploiting multi-touch displays to build interactive exhibits in which visitors work together to complete specific tasks supporting a common learning goal [14]. Scientists are also leveraging these displays to generate interactive visualizations to support collaborative investigation and analysis of complex, large datasets. More recently, video game developers have started to view multi-touch displays as a new platform for multiplayer games. Traditional computer games such as Pong [4] have been "ported" to multi-touch platforms, allowing a social, multiplayer experience.

While there are well-established design paradigms and principles for games targeting traditional video game hardware environments (such as video game consoles and PC's), these techniques fall short when applied directly to radically different hardware platforms interfaces such as multi-touch displays. As the hardware interface changes, the interaction scheme associated with the game can change dramatically, which in turn can significantly influence many aspects of gameplay. For example, a multiplayer game for a multi-touch display has to present a consistent view for all participants regardless of the position they are viewing it from. There have been some attempts at porting existing video games to multi-touch platforms (For example, Warcraft [1] and The Sims [11]). However, these ports often resort to emulating the mouse with input from the multi-touch screen, resulting in a single-player game that does not take full advantage of the multi-user capability of multi-touch displays. No multiplayer games, that effectively utilize multi-touch screens, have been directly ported from traditional video game hardware without a complete redesign of the game.

The goal of this paper is to aid game developers in understanding how the unique capabilities of multi-touch displays can be leveraged to create unique forms of gameplay that offer highly engaging multi-user game experience. Additionally, the limitations of the technology and their effect on gameplay are also illuminated. We discuss these issues in the context of a number of games that were developed during a semester long video game design course. During this course, groups of students majoring in Art and Computer Science designed and developed, from the ground-up, video games for TacTile, an LCD-based multi-touch display that supports as many as 500 simultaneous touches [6,14]. It is important to note that we define Multi-touch displays in

this paper as displays that can sense more than 2 simultaneous touches. Therefore we do not include iPhone games as part of this discussion.

The rest of the paper is organized as follows. Section 3 reviews previous research on multi-touch video games. In Section 3, we discuss the methodology used in the research leading to the conclusions of this paper. Section 4 describes the development platform. Section 5 describes three different multiplayer video games designed for TacTile, and documents different approaches for effective multiplayer game play. Section 6 summarizes the lessons learned and provides some guidelines for perspective developers of multi-touch video games.

2 Related Work

Tabletop and multi-touch displays have attracted large interest from the human-computer interaction community due to their intuitiveness and suitability for group-oriented tasks. There is a large body of research on the use tabletop and multi-touch displays for collaborative tasks. A classical example is the work of Scott et al [7]. However, the majority of work on these devices focuses on general-purpose, work-related tasks, rather than video games. While many of the conclusions and guidelines found in that research can be conceivably generalized to games, there are a number of unique factors that have to be considered when designing *multiplayer* video games, including enjoyment and competition between players.

In recent years, the interest in the application of multi-touch and tabletop displays to multiplayer gaming has increased. A number of classical video games have been ported to multi-touch platforms. One example is a multi-touch adaptation of the Atari Pong game [4]. The game is played with a single gesture that consists of a two-finger tap to form a racket.

Esenther et al [8] describe two multiplayer, multi-touch games, along with a software platform for rapid game multiplayer game development. In the first game, the players compete to find a special ball with a swirling pattern out of groups of blue balls. The hardware platform allows the identity of the player touching the table to be reliably deduced. Therefore, the game can distinguish which player finds the special ball first. In the second game, the players cooperate with each other to eliminate balls from the screen by touching them. Some special balls require that at least two people touch them simultaneously to be eliminated. Although the platform allows for distinguishing touches and associating them with different users, this ability requires special hardware setup which is not available in most commodity multi-touch platforms.

Khaled et al [9] describe two collaborative, multiplayer games. Both games revolve around moving a set of items on the screen and arranging them in some fashion, with the players collaborating to complete the task in a limited time.

Wolfe et al [10] describe a low-cost, projector-based, multi-touch targeted at game developers. A sample game is illustrated, with players cooperating to eliminate asteroids. Missiles are fired by touching the surface, destroying asteroids in the their vicinity.

Tse et al [11] study the behavioral patterns in cooperative game play to deduce guidelines for mutli-player video games for multi-touch displays. Their study uses ports of two existing single-user, commercial games to a multi-touch tabletop platform. The interaction scheme with the games was transformed to accept a rich set of gestures for performing different commands.

There has also been some interest in the use of multi-touch, collaborative gaming for social development. Examples are found in [12] and [13].

These studies offer good examples of multiplayer games for tabletop and multi-touch devices.

However, most of them focus on the user-experience and social interaction between players that arise in multi-touch, tabletop displays. However, there has been little work that illuminates effective gameplay concepts that can be practically utilized by prospective game developers to build engaging multiplayer games for multi-touch, tabletop displays.

3 Methodology

To investigate multi-touch games as emerging platforms for video games, we conducted a study as part of a video game design course. The aim of the study was: investigate how unique capabilities of multi-touch could be leveraged to develop engaging multiplayer video games, and derive a set of principles that could be used by prospective game developers. The study was conducted as part of a video game design course taught simultaneously at the University of Illinois at Chicago and Louisiana State University. The 29 graduate and undergraduate students who took the course were majoring in either Computer Science or Art. Eight teams were formed composed of three to four students. Each team was tasked with developing a multiplayer game concept suitable for a tabletop, multi-touch display, and implementing the game on the TacTile system. The materials of the course were designed to illustrate a vertical slice of the video game design process. Therefore, the projects emphasized completeness and polish of the final game products. At the end of the semester, the games were demonstrated to a panel of judges. The panel was assembled from expert video game developers from the industry, as well as Computer Science faculty with research background in Computer Graphics, Human-Computer Interaction, and Learning Sciences. The panel ranked the games on criteria that included gameplay design, interaction design, art design, sound design, and technical achievement. The top three ranking games were further analyzed and form the bases of the guidelines described below.

4 Development platform

The video games were developed on TacTile, a table-top, LCD-based, multi-touch display with a 52-inch screen that supports resolutions up to 1080p HD. The device was built using consumer off-the-shelf components. The first version of TacTile was assembled at the Electronic Visualization Laboratory at the University of Illinois at Chicago. Subsequent clones were later built at Louisiana State University, and the Science Museum of Minnesota. TacTile is capable of tracking up to 500 fingers simultaneously across the display surface, making it ideal for groups of 2-4 people. The finger tracking utilizes the Frustrated Total Internal Reflection (FTIR) tracking technique [2]. The device uses 3 infrared cameras that are tucked underneath the LCD display. A custom tracking application receives images from the cameras and analyzes them using traditional computer vision techniques to determine finger location.

The video games were implemented using Processing [3]. To simplify the game development, a Java API was developed to provide game developers easy access to the finger tracking information without the need to implement computer vision code.

5 Sample Developed Games

At the end of the semester, the students presented their game concepts and demonstrated the games in front of the judges' panel and audience. The judges played the games on the TacTile multi-touch display. The panel then ranked all 8 games that were developed during the semester long course, with the top 3 games selected, and their team members invited to submit a written report of their experiences. These games, along with the reports were later analyzed to draw the results of this paper.

The top ranking game was Zombie Apocalypse, a multiplayer, cooperative, puzzle-solving game

in which the players have to guide three characters safely away from zombies. The level can only be won if all the characters cooperated to solve the puzzle before the zombies overran them.

The second game was Galaxy Commander, a clone of the iPhone-based Galcon [5]. The game converts the single-player iPhone application into a two-player game in which players compete to conquer each other's planets.

The third game was Ball Buster, a multiplayer competitive game similar in style to the Pong in which a group of two to four players compete to shoot balls at other players' targets on the opposite end of the screen, while protecting their own targets with shields.

We describe each of these games in some detail, and discuss unique gameplay and interaction concepts that were found in each game.

5.1 Zombie Apocalypse

Zombie Apocalypse is a multiplayer, cooperative, puzzle-solving game that revolves around three characters who have to be rescued to a safety zone and away from hordes of zombies. During the game, the players have to navigate the characters around a number of obstacles and solve a number of puzzles that require coordination between the characters.

The game provides a top-down view of the current level, showing the entire map along with all the obstacles, characters, and zombies. The characters can be moved by touching them on the screen, and dragging one's finger to the desired destination. This forms a path that the character automatically starts following. Each character is also armed with one weapon that can be activated by tapping one's finger inside a circle that surrounds the character. By moving the finger while it is inside the surrounding circle, the direction of firing can be specified allowing the player to target zombies. The characters have weapons with varying potency and firing rate. For example one character has a flamethrower that is effective, but has a short range. Another character has a machine gun with a fast firing rate, but low potency. Figure 1 illustrates this.

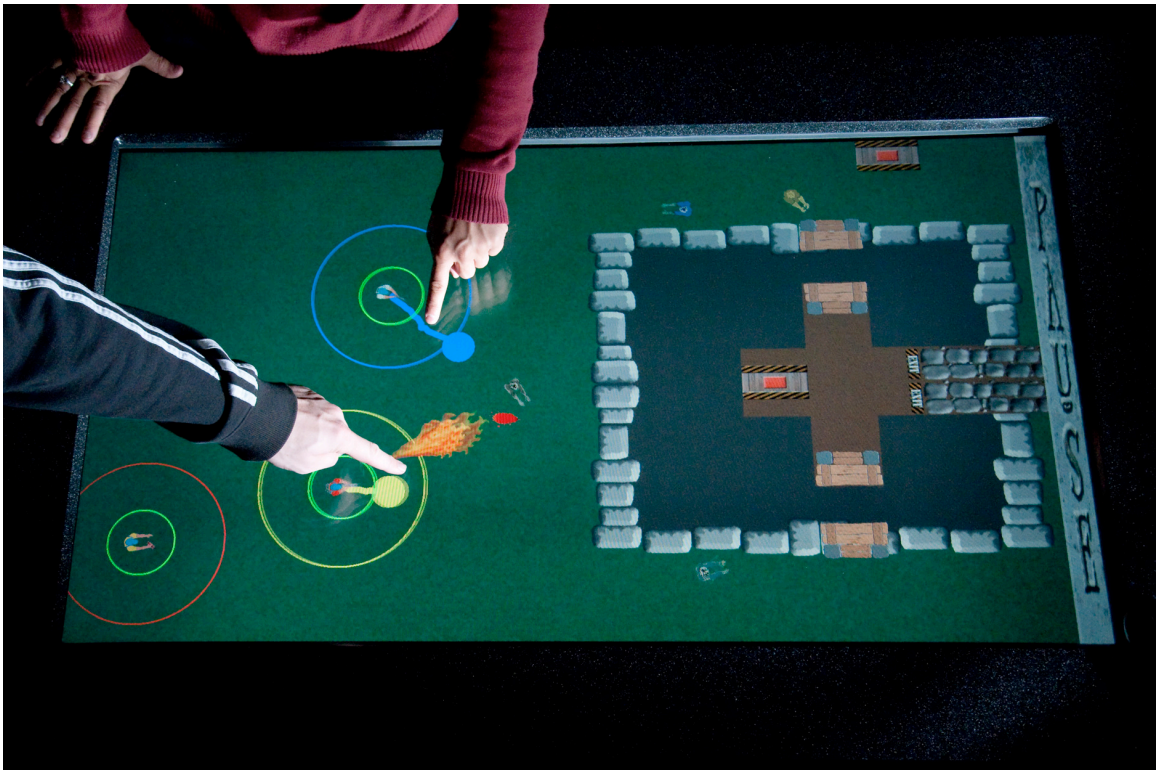

Figure 1. Players interacting with Zombie Apocalypse. Characters can be moved by dragging them across the screen. A tapping gesture is used to activate a weapon.

To successfully complete a level in the game, the three characters have to escape to the safe zone without getting caught by the zombies. This requires coordination and cooperation between all the players. For example, in the first level, the characters are on one side of a river that runs across the entire map, whereas the safe zone is on the other side of the river. On each side of the river there is a button that causes a bridge to be lowered when one of the character is standing on it, allowing the other two characters to pass over the river. However, one character has to remain standing on the button for the bridge to be lowered as the bridge get retracted as soon as that character steps off of it. Additionally, whenever the button is pressed, an endless horde of zombies starts coming from the side of the screen towards the characters. To solve the puzzle, one character has to be guided to stand on the button on one side of the bridge while others cross. After the two characters pass to the other side of the rover, one of them has to be guided to step over the second button, allowing the poor guy still left on the wrong side to pass the bridge to that side, and then the three characters can proceed to the safe zone.

Figure 2. Cooperative aspect of Zombie Apocalypse. The flame-throwing character has opened the top bridge for the remaining character. The machine gun character now holds the switch for the lower bridge. It is up to the other players to make it there and cross over to the safe area. However as a horde of zombies attack, it is difficult and requires a team effort to clear the zombies.

5.1.1 Findings

One of the principle issues that have to be taken into account when developing a game for a multi-touch, tabletop display is the orientation of the view. In traditional video game platforms, each player usually has his/her own screen, or the screen is split to show a separate view for each player. In either case, the player is looking at a separate screen or a non-overlapping area of the screen. Therefore, the game does not need to present a consistent orientation or view for all players. In the case of tabletop, multi-touch displays, the players usually stand on different sides of the display to maximize screen use. Therefore, multi-touch games have to present a single consistent view for all the players. The Zombie Apocalypse game presents a top-down view of the entire level on the screen, which makes it viewable consistently from all four sides of the display.

In traditional multiplayer games that depend on team cohesion and cooperation, it is not always straightforward for the players to devise a strategy to complete the objectives of the game. This is because players normally have access to a different, limited, and self-centered view of the level. Thus, in order to complete a level, players usually resort to trial and error. An advantage of mapping the entire game level to the physical display is that all players have complete situational awareness. This enhances the potential for social interaction between the players. These interactions include information sharing and strategic planning before, or during, game play. This fosters higher-level problem-solving discourse, which can be exploited by serious games that have an underlying education goal.

Another common issue that arises in cooperative, multiplayer games that have multiple characters is partitioning the control of characters among all players. The traditional scheme of dealing with this is to assign a single character to every player, and allowing the player exclusive control over that character only during the entire level. A limitation of this approach is that the characters have to have balanced roles in the game. This includes comparable abilities, and equal influence over the events and outcomes of the game. Ignoring this principle would likely lead to frustration of players who have limited power or influence. In Zombie Apocalypse, this is no longer an issue. The players can easily shuffle between characters by simply touching the desired character on the screen. Moreover, players can negotiate in real-time among themselves who should be controlling what character. This decision can be based on which player is standing closer to the character's position, for example. More importantly, this allows all the players to experience all the different characters, leading to more options and possibly increased player engagement. This scheme however, works only in cooperative games in which all the players are working towards a common goal. Different control schemes have to be devised for competitive games.

5.2 Galaxy Commander

Galaxy Commander is a competitive two-player game inspired by the iPhone-based game Galcon [5]. Much like the original Galcon, the game revolves around two teams, a red and a blue team, each controlled by one of the two players who compete to conquer all the planets in the galaxy. The two players stand on opposing sides of the display. Each player starts off with one planet and proceeds to conquer more planets throughout the game. Planets conquered by one of the players continuously produce ships which can be later used to invade other planets. The number of ships stationed in each planet is indicated on the planet. The main method of conquering additional planets is by dispatching a fleet of space ships from one of the planets already controlled by the player to a new one. This can be done by holding down an allied planet and dragging ones finger to another planet. The fleet will either attempt to occupy neutral planets, seize opposing planets or reinforce allied planets. An attempt to seize is successful if the number of ships in the invading fleet exceed the number of inhabitants garrisoned on the planet. The overall objective of each team is to dominate the opponent's planets and conquer all of them. Additional special abilities such as shields and weapons of mass destruction have also been introduced. The deployment of these enhancements follows the same drag and release gesture used for attacking planets.

Figure 3. Players mobilizing their ships for attack in Galaxy Commander with a simple drag gesture. Both players are linking planetary resources together across multiple planets to form a larger attack fleet.

5.2.1 Findings

Similar to Zombie Apocalypse, Galaxy Commander also uses a top-down view that shows the entire level. Since the game was designed to be playable by two players, the textual elements such as the number of ships in each planet are rendered in two orientations, making it easier to both players to read. Additionally, the main icons such as the planets and ships were symmetric, making them easily recognizable from both sides of the display. On the other hand, the user interface elements, which consist of two buttons to activate the special abilities are replicated for each player, and positioned close to the side on which the player is standing. Much like the Zombie Apocalypse, the main interaction area can be accessed by both players who compete to conquer more planets, and invade each other's planets. A side effect of this is that a player can dispatch ships from his/her own planets, as well as other planets owned by the opponent. Although the rules of the game prevent this from happening, there is no easy way of technically enforcing this rule. Most touch displays (including TacTile) identify touch points only, and cannot associate these points with a specific user. This led to some unexpected interactions when the game was demonstrated to judges and audience members. For example, one player prevented his opponent who was attempting to dispatch ships to invade one of his planets by physically blocking the opponent's hand and preventing her from completing the drag gesture to dispatch invading ships. While these situations are technically termed "cheating", their spontaneity greatly enhanced the enjoyment of the game by the players.

5.3 Ball Buster

Ball Buster is a fast-paced, competitive, multiplayer game that was designed to be playable by

two to four players, with each player standing on one of the four sides of the display. Each player has a rectangular Goal Area where five targets are located. The Goal Area is positioned near the side on which the player is standing. The goal of the game is to hit the other player's targets and protect one's own. A player attacks opponents by "shooting" balls from within the Goal Area. A shot gesture consists of pulling back and releasing a touch. The objective is to eliminate all opponents' targets while protecting their own. To defend a barrage of incoming balls, a player can form temporary shields. A shield is created using a simultaneous two-finger tap gesture in a player's Defense Area, deploying a straight-line barrier between the two fingers. A ball ricocheting off a shield changes ownership and direction, causing it to become a hazard to opponents. A player is eliminated if he/she looses all his/her targets, and the last remaining player becomes the winner.

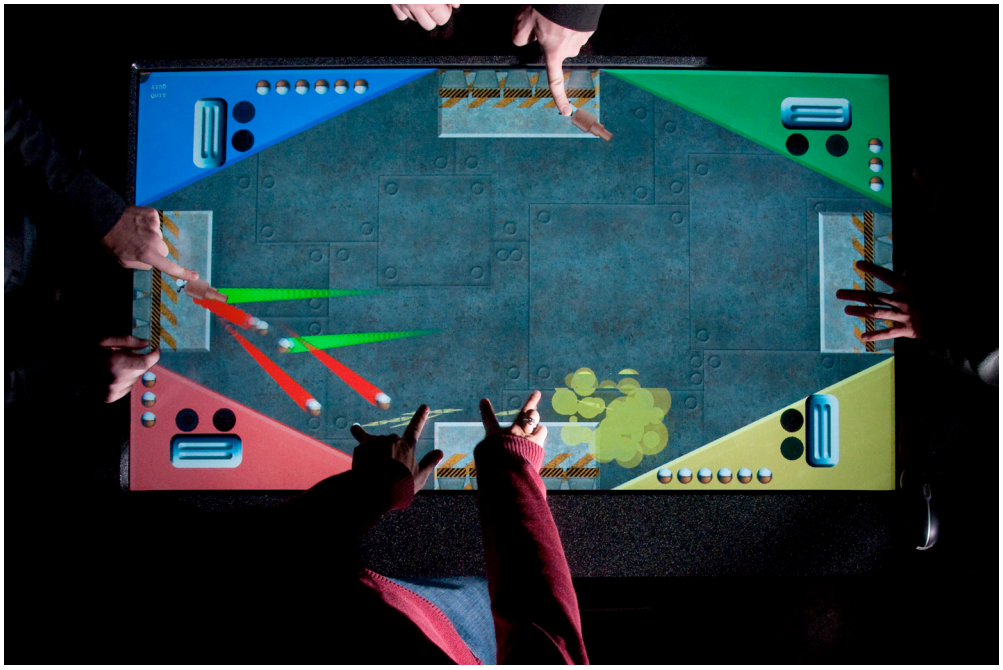

Figure 4. The player on the far left and at the top show the mechanics of the drag and release fire control. The player on the bottom shows how two finger taps rapidly deploys shields.

5.3.1 Findings

The control schema for Ball Buster consists of a small set of gestures. The entire game is played with only two gestures: pulling back and release to shoot and two finger touch to defend. Additional cues are given by the graphical representation of cannons and barriers that appear at an instance of a touch.

Although most multi-touch applications use gestures that are intuitive and easy to perform by the average users, other factors have to be considered when developing gestures for multi-touch video games. For example, the time required by the average player to complete the gesture, and potential fatigue from repetition are also two important factors. An earlier prototype of Ball Buster had a 'draw' gesture that allowed players to deploy shields by brushing with one's finger. Although this gesture was found to be more intuitive than the final two-finger tap, the drawing gesture required more time to complete, leaving the players with less time to react to an incoming ball. Therefore, although more intuitive, the draw gesture was dropped in favor of the two-fingers

tap gesture which can be completed faster by players. Additionally, tapping does not generate finger friction with the display surface, minimizing player fatigue (or what we fondly call "Flaming Finger Syndrome").

6 Lessons Learned and Design Guidelines

Multi-touch displays offer great potential for a new generation of engaging, multiplayer game play. The fact that all the players share the same input and output device creates opportunities for social interaction between the players, and foster high-level, problem-solving discourse between them. This can be used to create a wide variety of gaming experience ranging from purely entertainment-centered game play to serious games supporting a learning goal. A comparison of multi-touch versus traditional game design, and general guidelines for multi-touch game interaction design is summarized in Table 1.

It is tempting to apply the same techniques and principles for traditional video games to multi-touch platforms However, due to the uniqueness of this platform, game developers have to learn to break from some of the previous notions, specifically game interaction and control techniques.

After analyzing the games presented in Section 5 along with the opinions of the judging panel, we have derived a set of guidelines for prospective multi-touch game developers. We discuss these below.

6.1 Design games to be orientation independent

In multi-touch devices, players are usually standing on different sides of the device, with each player getting a view of the game from a different angle. For the game to be enjoyable by players, it should present a consistent orientation to all the players regardless of which side they are standing on. A good technique is to design the game with a top-down vantage point. Additionally, using semi-symmetrical shapes for the major helps players easily recognize them. Text is inherently difficult to deal with. Therefore it should be kept at minimum. If text is required, it should be rendered in at least two orientations whenever possible.

This principle should not only be applied to the graphics, but should also be extended to game logic. For example, a game that includes a gravity component in a horizontal direction would be difficult play by multiple players standing on different sides of the screen, as the game logic would not be easily grasped.

6.2 Control and interaction

Input devices for more traditional games usually revolve around a dedicated controller with a finite number of inputs. A multi-touch device offers a single input device with a potentially infinite number of input schemes. A challenge with most multi-touch platforms is that it is not technically feasible in most situations to associate touches with a particular user. This imposes significant changes on game interaction to traditional video game platforms. Whether this is an issue or not depends on the nature of the game. A cooperative game where all the players are working together to achieve a common goal does not necessarily need to distinguish between all the different players. The notion that all the players are treated equally by the game adds more potential for social interaction between the players.

On the other hand, a competitive game will likely need some way of distinguishing between the actions of different players. A good solution to this is to dedicate a private control area for each player inside which the player performs the actions the need to be identified. This area should be placed closer to the side on which the player is standing. A hybrid approach in which there are

private areas as well as shared areas accessible by all players is also useful in some competitive games. The identity of the player performing actions in a shared area can be sometimes disambiguated from the context. For example, the identity of a player touching an object in the shared area which he/she owns can be assumed to be the rightful owner of that object. This technically does not prevent an opponent from interfering with other players' objects. However, cheating is usually easy to detect as players can see all actions by all other players in the game.

6.3 Evaluate ergonomics of interactions

As with all other video game platforms, there is some potential for stress, fatigue, and possibly injury resulting from continuous and lengthy play. In multi-touch displays, there are two additional factors that might contribute to this that game developers should be aware of. In a large number of multi-touch, tabletop devices (including TacTile), users are usually standing around the table, as there is no sufficient room for sitting due to the fact that the depth of these devices is usually on the order of few feet. Users normally do not interact with the device for more than 30 minutes. Therefore, game developers should design levels to be completed within a time frame that is less than 30 minutes.

The input schemes used in the game should also be considered from an ergonomic point of view. For example, gestures should be designed to minimize stress that players might experiences. One particular gesture that we found to be potentially stress inducing is brushing. As players repeatedly brush their fingers on the screen, this increases friction with the display surface, causing some discomfort. Therefore, game developers should consider multiple gestures for a particular action, and evaluate them from the perspective of comfort.

6.4 Design around the limitations of multi-touch devices

The technology behind large-scale, multi-touch displays is still relatively new. These platforms still suffer from a number of limitations that lead to more challenges for game developers.

The majority of multi-touch platforms rely on optical tracking techniques to determine the position of fingers (usually with infrared light). This creates potential for interference from external light sources. Some multi-touch displays work properly only if deployed in dark rooms with controlled lighting conditions. We expect this problem to become less of an issue with time as more robust tracking techniques are developed.

7 Conclusion

As multi-touch displays continue to proliferate, it is inevitable that it will be a force in the realm of video games. Video game programmers will be challenged by this technology. Successful responses will be games that harness the innovations and limitations of touch technology to form wholly new gameplay paradigms. This paper has provided, through example, an initial discussion of some of these possibilities, and we intend to continue to investigate this further in future iterations of the game design class.

Acknowledgements

This work was partially funded by National Science Foundation Award# CNS 0821121 and the Science Museum of Minnesota. We would also like to thank Arun Rao - High Voltage Software, Mark Tulewicz - Incredible Technologies, Alex de Vera and Dom Olmstead - Chewy Software, for their invaluable feedback.

References

1. PQ labs Multi-Touch G2 Demo. URL: http://multi-touch-screen.com/demo.html

2. Han, J.Y. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In UIST 2005. pp. 115-118.

3. Processing. URL: http://www.processing.org/

4. Gross, T., Fetter, M., and Liebsch, S. 2008. The cuetable: cooperative and competitive multitouch interaction on a tabletop. In CHI '08 Ext. Abstracts on Human Factors in Computing Systems, ACM Press (2008), pages 3465-3470.

5. Galcon for iPhone. URL: http://www.galcon.com

6. TacTile. URL: http://www.evl.uic.edu/cavern/tactile/

7. Scott, S. D., Grant, K. D., and Mandryk, R. L.: System guidelines for co-located, collaborative work on a tabletop display. In Proc. ECSCW '03, Kluwer Academic Publishers (2003), 159-178.

8. Esenther, A. and Wittenburg, K.: Multi-user multi-touch games on DiamondTouch with the DTFlash toolkit. In Proc. INTETAIN '05, Springer (2005), 315–319.

9. Khaled, R., Johnston, H., Barr, P., and Biddle, R.: Let's Clean Up This Mess: Exploring Multi-Touch Collaborative Play. In CHI '09 Spotlight on Works in Progress. ACM Press (2009), pages 4441-4446.

10. Wolfe, C., Smith, D., and Graham, N. A Low-Cost Infrastructure for Tabletop Games, in Proc. FuturePlay 2008, ACM Press.

11. Tse, E., Greenberg, S., Shen, C., and Forlines, C.: Multimodal Multiplayer Tabletop Gaming. ACM Computers in Entertainment 5(2), article 12 (2007).

12. Piper, A. M., O'Brien, E., Morris, M. R., and Winograd, T. SIDES: a cooperative tabletop computer game for social skills development. In Proc. CSCW '06, ACM Press (2006), 1-10.

13. Al Mahmud, A. Mubin, O. Renny Octavia, J. Shahid, S. LeeChin Yeo Markopoulos, P. Martens, J., and Aliakseyeu, D.: Affective Tabletop Game: A New Gaming Experience for Children. In Proc. TABLETOP '07, 44-51.

14. Leigh, J., Johnson, A., Renambot, L., "Advances in Computer Displays" (chapter), Advances in Computers, Volume 77, Marvin Zelkowitz (editor), Elsevier (ISBN 978-0-12-374812-6), 2009, pp. 58-79.

TABLE 1 : This table lists the differences between traditional console/PC games and multi-touch games. Offers insight and sites games examples to address these differences.

	Traditional Console/PC Gaming	Multi-Touch	Recommendations for multi-touch development	Example
Visual Orientation	The display is usually vertical and faces the user. This provides a single orientation to the game experience.	Players are often standing on different sides of a horizontally placed display. Consequently, the game should be orientation-independent to accommodate players regardless of which side they are standing on.	A good technique is to design the game with a top-down vantage point.	All the successful games mentioned is this paper use this approach. While there were other games that provided different vantage points (e.g. split screen views), they tended isolate the players from each other.
			Using semi-symmetrical shapes for major characters or important objects helps players easily recognize them.	Zombie Apocalypse displays a character's health as concentric rings rather than a bar at the "top" or "bottom".
			Textual elements should be kept at minimum since they are hard to read from all possible viewpoints.	Ball Buster was designed to be a game that was understandable, and with minimal text.
			If some text is required, it should be rendered in at least two orientations.	Galaxy Commander implements this technique of mirroring text well.
			This principle of symmetry should also be applied to the actual game logic.	Given the top-down point of view for the games mentioned. Movement and physics is confined to a 2D plane. This makes the interactions uniform to all orientations.

	Traditional Console/PC Gaming	Multi-Touch	Recommendations for multi-touch development	Example
Ergonomic Considerations	Provides tactile feedback via depression of button, pushing of a joystick, or clicking a button. Users have been shown to be able to use devices for long periods of time.	A large source of fatigue that occurs in users is the Flaming Finger Syndrome. This is caused by friction induced by rapid or longer-term interaction with the touch device.	Restrict fast-twitch interactions to minimize flaming finger syndrome. One method is to eliminate the need for excessive rapid dragging.	Ball Buster was designed with this in mind. Short finger flicks were used to launch balls. However a limit was placed on the number of balls fired. This ensured a period of rest. Additionally, the shields were implemented with a two-finger tap rather than a dragging motion.

	Traditional Console/PC Gaming	Multi-Touch	Recommendations for multi-touch development	Example
Interface ownership	Offer a dedicated controller with a finite number of inputs.	A touch device is a single input device. Consequently, no notion of ownership or player identification is implied.	Resist the urge to use software-based emulations of hardware interfaces where possible (e.g. virtual D-pads). It is difficult for players to both view the game screen and the interfaces at the same time.	Zombie Apocalypse originally used a virtual D-Pad interface at the corners of the screen for controlling the game characters, before it was eventually replaced with an interaction scheme that involved players directly manipulating the game characters.
			Minimize the need for players to memorize complex gestures.	A player can navigate through each of the games mentioned with a small set of simple gestures. For example Galaxy Commander's entire game play is based on a simple drag and release gesture.
			In most cases there is no way to associate a player with a touch. Try to leverage this to enhance cooperative play.	Zombie Apocalypse accomplishes this effectively by allowing players to control any character on the screen. Additionally, a character can be shared by having one player control movement while the other control shooting.
			For more competitive games, create a small-localized control area. This reduces the possibility of players interfering with opponents' controls, and minimizes confusion by the gesture recognition system.	An example of a well-designed control area is Ball Buster's Goal Area. The Goal Area provides a relatively secure area were a person can access fire control.