Multiplayer game design on the DiamondTouch table

Daan Scheerens
d.scheerens@student.utwente.nl

ABSTRACT
With the arrival of digital tabletop systems new interaction methods using multi-touch are supported, allowing for new possibilities in computer games. This paper presents a game design for multiplayer game on the DiamondTouch table. The game is designed by identifying the characteristics of the DiamondTouch and relating them to game characteristics. By doing so the important design aspects for multiplayer games on the DiamondTouch are found.

Keywords
Multiplayer games, multi-touch, digital tabletop systems

1. INTRODUCTION
Digital tabletop systems are tabletop surfaces that are augmented with sensing and display capabilities [5]. These augmentations allow interaction between multiple humans (seated around a table) and a computer. Thus with the arrival of digital tabletop systems, application designers are offered new possibilities for the design of applications that require interaction between multiple users and the computer. Such applications use so called multi-user interface.

Before the era of digital tabletop systems application designers were limited to the single user interface of traditional desktop computers. The most common input devices for desktop computers, keyboard and mouse, are designed for a single user and therefore make the desktop computer unsuitable for multi-user applications. For most applications this limitation does not really create a problem since the tasks that have to be performed do not require a multi-user interface. However there are certain areas in which applications could really benefit from a multi-user interface. For example, in meetings and discussions digital information is often shared between the participants, such as presentations, documents, notes, etc. A single shared user interface would allow for easy exchange of information, which improves communication. Another area that could benefit from a multi-user interface is the field of multiplayer gaming [1,2].

Multiplayer games are games with two or more human participants. To support these multi-user activities on desktop systems, games are typically using networking capabilities of the computer. As a consequence the application actually requires multiple computers, one for each player. One of the disadvantages of this approach is that the players have to focus on their personal interface, thereby hampering the direct interaction between the players [2]. Console game often use split-screen views to support multiple users. Although the users are sitting more closely to each other, they are still directed towards the screen making it hard to focus on one of the other players. When using a digital tabletop system instead the participants are facing each other and share the interface (like in traditional board games), resulting in a richer social interaction [6]. This is a great benefit for games in which the players have to extensively interact with each other, such as role playing games. Other types of multiplayer games might not be able to benefit from the digital tabletop platform. For example in most first person shooters there is only very limited need for close interaction between the players.

Although it seems obvious that a multi-user interface for some multiplayer games would be appreciated it might not be that easy to recognize what features will constitute to a joyfully game experience. Furthermore embedding these features into the design of a game may not be a straightforward task. Current research has focused on the development of frameworks that attempt to aid the designer to implement these features into a game [2,3,4]. For example the STARS framework [2], helps with presenting game information in the right form, taking into account that some information might be important to all players, whereas other information is only relevant for a few players.

This research identifies the aspects that are important for the design of a multiplayer game for digital multi-touch tabletop systems. For this research I will use the DiamondTouch [5]. Since there might be many aspects involved in the design of a multiplayer game for the DiamondTouch, the focus of the research will be on the design of a multiplayer game that uses as many of the features offered by the DiamondTouch system as possible. So the main research question is:

What design aspects are important for a multi-touch multiplayer game on the DiamondTouch?

To find an answer to this question a design for a multiplayer game on the DiamondTouch is made using the following approach. First a literature study of the characteristics of multi-touch in general is made and a literature study of the DiamondTouch is made to find the abilities and limitations of the system. After these have been found a quick study of multiplayer games is made to determine characteristics that are suitable for the DiamondTouch. By combining the results from these studies a design for a multiplayer game will be made. At this point the design aspects should have been found. To verify whether these are indeed important for a multiplayer game on the DiamondTouch a prototype of the design will be implemented, which will be tested and evaluated.

2. MULTI-TOUCH
Multi-touch is an input method that allows a person to use multiple fingers (or other limbs) to interact with a computer system that has a touch sensitive surface. Often the touch sensitive surface serves as a display, allowing for direct interaction. Several varieties of multi-touch systems exist. Some use vertical wall mounted surfaces, others are embedded into table surfaces and some are mobile handheld devices (such as the iPhone). This section will give an overview of some of the systems that are available. The DiamondTouch will be discussed in more detail in section 3.
2.1 SmartSkin

The architecture of the SmartSkin system [8] is closely related to that of the DiamondTouch. It consists of a touch sensitive surface on which images are projected from above. The touch sensitive surface is composed of a mesh shaped sensor grid. Transmitter and receiver electrodes are placed perpendicular on the surface and are separated from each other using an insulating material. Each of the transmitter electrodes is driven by a unique wave signal. Once a conductive and grounded object (for example somebody’s hand) approaches the crossing of a transmitter and receiver electrode, that point acts as a capacitor. This causes the wave signal of the transmitter electrode to be propagated to the receiver electrode. The amplitude of the wave signal on the receiver electrode is affected by the proximity of the object to the crossing point, allowing the system to sense the distance between the object and the SmartSkin surface. Since each transmitter electrode is driven by an unique signal, it is possible to construct a 2D image of the object’s distance to the surface. This is done by analyzing the signals of the receiver electrodes. Although the system is able to detect multiple touches at the same time, it can not distinguish different users.

2.2 CueTable

Gross et al. [9] present a custom build table with touch sensing capabilities, called the cueTable. An acrylic glass sheet covered by tracing paper forms the surface of the table. The surface is illuminated with IR light, by LEDs that are placed at two opposite sides of the glass sheet [11]. Once a person touches the table, the IR light is frustrated which causes a light spot to appear at that point on the surface. This spot is detected by a camera with a wide angle-lens underneath the table. Images are displayed on the surface of the table using two tilted mirrors and a projector, which are placed under the table. A visible light filter that allows only infrared light to pass, is applied to the camera so that it will only register touches and it is not influenced by the images displayed on the surface.

In contrast to the SmartSkin system, the cueTable uses bottom up projection to display images. Also instead of using capacitive coupling to detect touches, the cueTable uses the phenomenon of Frustrated Total Internal Reflection (FTIR) and a camera to detect touches [11]. This method does not allow to register the distance of an object that is close to the surface of the table. Finally, like the SmartSkin system, it is able to register simultaneous touches, but it can not distinguish different users.

2.3 Visual Touchpad

Another method to construct a touch sensitive surface is by using computer vision techniques. Malik and Laszlo [12] use two cameras to create a vision based touchpad, the Visual Touchpad. The touchpad is a simple black panel with a white border. Two webcams are placed so that they have a full view of the panel. Using computer computer vision techniques, the raw images from the webcams are processed to track the users hands.

Because two cameras are used, depth information can be extracted from the images. This allows the system to detect the distance of the fingertips to the surface of the touchpad. Additionally finger orientation is extracted from the images to support more advanced interactions.

The Visual Touchpad is another example of a multi-touch system that is unable to distinguish different users. However, by adding markers to the fingertips, this system can be extended to support the user identification feature. Unlike the SmartSkin and CueTable systems, the surface of the Visual Touchpad does not act as a display, so it does not support direct interaction.

3. DIAMONDTOUCH

The DiamondTouch has been developed in response to experiences with a collaborative workspace in an earlier research on Human-Guided Simple Search at MERL [5]. One of the results was that participants thought that collaboration could be improved if multiple users would be able to interact with the system simultaneously. These are exactly the most important characteristics of the DiamondTouch, it is able to handle simultaneous input and uniquely identify the users.

3.1 DiamondTouch system setup

Roughly sketched, the DiamondTouch system is composed of a touch sensitive tabletop surface with an overhead mounted projector that projects images on the surface. Both the projector and the touch sensitive surface are connected to a computer. A sketch of the setup for a DiamondTouch system is given in Figure 1.

![Figure 1 DiamondTouch setup](image)

A capacitive coupling technique is used to give the surface touch sensitive capabilities. Two perpendicular arrays of antennas, connected to a transmitter unit, are placed just below an insulating surface on top of the table. Users of the system are capacitively coupled with a mat that is connected to a receiver unit. When the user comes in close proximity to the surface of the table a capacitively coupled circuit is completed and allows the DiamondTouch system to detect the location of the users presence. Since there is a different signal for each mat, users can be identified by the signal that is received. As the distance between the user and the table surface decreases, the strength of the received signal increases. A threshold is used to distinguish between hovering over the table and actually touching the surface. To compensate for background noise in the signals, the threshold is adjusted dynamically (though it is possible to set a fixed threshold).

3.2 Body presence

So far, apart from the simultaneous input and user identification characteristics of the DiamondTouch, two other properties have been found. First, instead of only detecting discrete touches, the DiamondTouch is also able to detect the presence of a body close to the surface. The other characteristic (discussed later) is that a user has to remain in contact with one of the receiver mats.
The body presence that can be sensed by the DiamondTouch is usually a human body part (finger, hand, arm, etc.), but in fact it can be any current conducting object that is connected to a receiver wire of the DiamondTouch.

A similar property is found in the SmartSkin multi-touch system, where the signal strength is used to estimate the distance between a body and the table surface. This information can be used to extend to scope interaction methods on such a system. For example one could think of a game where the user has to try to catch a mouse that is walking over the table. To make the game more challenging and more realistic the mouse could run away if it senses the presence of a person (try sneaking up on a mouse in reality). By using the information about the distance of the hand of a user over the table this behavior could be simulated.

When testing the ability of the DiamondTouch to detect the presence of a hand (or another part of the body), it became clear that the level of receptiveness depends on the contact between the user and the mat. If the area of mat that was in direct contact with the user was increased, the DiamondTouch was able to sense a user from a greater distance above the table. The system was able to detect a hand up to about 1 cm when the user was standing on the mat. If the user was sitting instead of standing on the mat the distance increased to about 3 cm. A maximum distance of about 4 cm was found when the user was in direct contact with the receiver unit (the wire was connected directly to the users hand instead of the mat).

From these results can be concluded that the game described above is not going to work on the DiamondTouch, because the maximum distance between the surface and user that can be sensed is too short. However this distance might be enough for other applications, therefore this characteristic of the system will not yet be considered as irrelevant. Note that it might be possible to increase the maximum distance that can be detected by using an additional sensor, for example with a camera. Such a setup is however beyond the scope of this research.

### 3.3 User identification

Another characteristic found so far is that users have to be in contact with a mat, otherwise the system will not detect their input. Since the location of the mat is usually fixed (users are most likely standing or sitting on them) a restriction is placed on the users freedom of movement. As the DiamondTouch table is large enough for multiple persons to be standing or sitting at its sides, it might be natural for the users to shift positions. While this is physically possible, it becomes a problem since the users need to stay in contact with their mat.

When designing the game thought should be given to this restriction: a game that requires the user to be physically moving around the table can not be played on the DiamondTouch (for example a virtual version of table tennis with multiple players that rotate positions).

An interesting side effect from the technique used in the DiamondTouch to detect and identify users is that physical contact between the users has an effect on the input of the system. If a user X touches user Y, any input made by user Y is also made by user X. This happens because user Y carries two signals and causes the system to receive both the signals from user Y and user X (although at a different strength). In most cases this behavior is not appreciated, however it also possible to design an application in which behavior is actually part of the interaction methods.

### 3.4 Overhead projection

As mentioned before, an overhead mounted projector is used to project images on the surface of the table. From the manufacturers point of view the advantage of this setup is the simplicity of the hardware, compared to bottom up projection or touch-sensitive LCD technologies. In practice this setup has the advantage that objects or body parts above the table do not occlude the areas underneath, because the image is projected partially on top of the objects or body parts. So overhead projection improves visibility compared to ‘under the table’ displaying. In [10] an interesting interaction method is presented; when the user makes a special gesture by laying a tilted horizontal hand on the table surface, private information is displayed onto the users hand. This interaction method clearly demonstrates the practical advantage of the overhead projection setup.

Suppose however that someone is bending over the table to reach an area at the opposite side. A large part of the image will probably be projected at the back of the person that is bending over, which might prevent other users to be able to see important information. Therefore it would be impractical for users to have to reach for areas that can not be reached without having to bend over the table. Also note that it is not an option either for users to walk around the table to be in range of the area of their interest, because they have to stay in contact with their mat. So this leads to yet another limitation of the DiamondTouch: user input should be clustered in the near vicinity of users.

### 3.5 Antenna layout

The DiamondTouch is equipped with a simple antenna layout which senses the presence of a body in two arrays: rows and columns. A consequence of this layout is that there is not a complete 2D figure available of the proximity of bodies. Instead of being able to detect the shape of a body as is possible with the SmartSkin system, the DiamondTouch can only detect bounding boxes.

![Figure 2 Touch identification problem](image-url)
point. A problem occurs when a user touches the surface with two or more fingers. In such a case a maximum of \( n^2 \) blobs can be detected, where \( n \) is the number of touches. When only considering a signal snapshot of the input signals, the system cannot tell which of the blobs are the actual touches. This problem is illustrated in Figure 2. For the system to distinguish the real touches from the 'phantom' touches, it needs additional information. The only extra information that is available to the system is the history of the signals, but it can indeed be used to solve the problem just described. If the previous touch points are known, then the system can use this information to make reasonable assumptions which of the blobs in the current snapshot correspond to real touches.

Gestures made with a certain pose of the hands cannot be detected, because this requires a complete 2D image of the surface. So the only gestures that can be recognized on the DiamondTouch are touches whose proximity to the surface or location changes over time (for example drawing a cross).

### 3.6 Response time

The update frequency (number of times the surface is scanned) of the DiamondTouch is 30 Hz. As a result touches and gestures will be delayed for at least 33 ms before they arrive at the computer application. However there are additional delays before an application receives touch information. First the data has to be transferred from the DiamondTouch to the computer. Once it has arrived at the computer, the data will first pass through a device driver, which might do some processing of the data first before it is passed to the application. Finally if tracking techniques are used, this will cause the touch information be delayed even more. How much this delay will be is unknown. It depends on several factors, such as the computer system being used and how much of the data is preprocessed. However experience has shown a total delay around 100 ms, which is significant enough for users to notice it. The application designer therefore has to take care that situations which require a quick reaction of the user, have a reasonable reaction time margin which is greater than the 100 ms response time.

### 3.7 Characteristics and limitations

In the previous sections several characteristics and limitations of the DiamondTouch have been found that should be considered when making a design for a multiplayer game. The following characteristics were found:

- Users can interact with the system simultaneous.
- Input from users can be uniquely identified.
- Presence of a body close to the surface can be detected.
- Users have to be in contact with a mat.
- Images are displayed via overhead projection, so occlusion problems can occur if, for example, when someone bends over the table.
- Touches and proximity is sensed in two arrays, rather than a full 2D image of the surface, so the contours of a body touching the surface cannot be detected.
- There is a delay around 100 ms before an application receives touch information.

From these characteristics the following limitations were found:

- Different hand poses can not be distinguished, only gestures made by tracking touches are supported.

### 4. MULTIPLAYER GAMES

In this section several game characteristics are considered, that are expected to be of importance for the design of a multiplayer game on the DiamondTouch. Most of this section is based on personal experience and assumptions.

#### 4.1 Real-time and turn taking games

When considering the timing of actions that can be performed in a game by the users, two types of games can be distinguished: turn taking games and real-time games. In turn taking games, such as in the Civilization series\(^1\), players have to take turns in which they can perform actions. During a turn a player usually has a limited number of actions that can be performed, for example moving units. Also during a players turn, timing is not important because the state of the game will not change except by actions performed by the player that is currently taking turn. This is in direct contrast to real-time games, where players can perform actions simultaneously. In such games timing is important during the whole game as the game state is constantly changing because of actions taken by other players. It might also change without other players performing actions, for example a moving ball will cause the game state to change over time as well.

Since the goal of this research is to design a game that uses as much of the capabilities of DiamondTouch as possible, turn taking games are no candidate. This is because there is no need to uniquely identify the input of users in turn taking games, since there is only one user at a time allowed to be performing actions.

#### 4.2 Point of view

The most important part of a game is its playfield, as this is a representation of the current game state (ignoring text-based games). Usually players will have interact extensively with the playfield, which makes it also an important aspect in the design of a game. There are several different perspectives from which the game can be viewed. A common distinction is made between 1\(^\text{st}\) person view, 2\(^\text{nd}\) person view, platform view and top down view. These views are not mutually exclusive, so it is not always straightforward to classify the view of a game. Some games even alternate between different points of view. If the point of view changes the actions that a player can perform are most likely to change as well. In turn this might require other interaction methods for each different point of view.

##### 4.2.1 1\(^{\text{st}}\) Person view

In the first person view a player sees the playfield through the eyes of a virtual character. The view is displayed using 3D graphics. Because this view is focused on a single player it is not suitable for a multiplayer game on the DiamondTouch system, where the users have to share a display. It is possible to divide the screen into different views for each player, but this would remove the option to have shared interaction areas on the table surface.

##### 4.2.2 3\(^{\text{rd}}\) Person view

This view is much like the 1\(^{\text{st}}\)-person view, with the only difference being that the player is able to see the virtual

character (instead of looking through its eyes). Since the playfield is displayed using 3D graphics, this view is only suitable for the DiamondTouch if all the virtual characters are visible at the same time (or at least those that are still active).

4.2.3 Platform view
A classic example of a game with a platform view is Super Mario Bros\(^2\). In these games the player is looking at a virtual character from the side. Usually this character can only move freely in the horizontal and vertical directions, but is restricted to a fixed depth (although multiple depth levels might exist). Since the screen is in most cases not large enough to display the entire playfield, scrolling has to be used. For multiple players scrolling becomes a problem, as players might be moving in different directions. So for a multiplayer game on the DiamondTouch the platform view can only be used if either the whole playfield can be displayed or all players will remain visible once the screen scrolls to another part of the playfield.

4.2.4 Top down view
In the top down view the playfield is viewed from above. In contrast to the other views a player is usually not represented by a virtual character. Instead players are likely to control areas and/or multiple units on the playfield. This point of view is most suitable for the DiamondTouch, but as with the 3rd person and platform views, care has to be taken that the important parts of the playfield for each player are visible at all time (else some players might have unfair advantages over others).

4.3 Navigation
Since all games are some sort of representation of a virtual environment wherein the players have to execute their actions, navigating through the environment is a common task performed by the players. The navigation in one game might be very different from another one. For example in a chess game navigation is merely just a matter of moving pieces from one cell to another. However in a first person shooter like Unreal Tournament, the navigation task consists of walking (and jumping) through a 3D environment.

Although navigation might vary for each game they all have a certain degree of freedom. The amount of freedom can be viewed as the number of dimensions found in the game, which are either 2 or 3 dimensions. Usually the number of dimensions is related to the type of graphics used, 2D or 3D. Furthermore the navigation can be split into two components: positioning and rotating.

A common method of navigation found in 3D games for positioning is by using the keyboard to move left, right, forward or backward. Rotating is accomplished by moving the mouse. For a multi-touch surface, navigation in a 3D environment is not as intuitive as for traditional computer systems with a mouse and keyboard. The task of rotating can be easily translated, because the equivalent of moving a mouse is moving with a finger over the multi-touch surface. However the task of positioning is less straightforward. One option is to display buttons on the surface that act as a virtual keyboard, but then the multi-touch interface loses its value. Another option is to use gestures to perform the positioning task, for example drawing a line in a certain direction indicates moving into that direction. It requires however some training to get used to this method of input since a player has make complex and asymmetric movements with both hands, which is hard thing to do for most persons [13]. In multiplayer games there is often a lot of action going on, requiring the players to quickly respond to certain events, making this input method for navigation unsuitable.

In 2D game environments there are many different methods of navigation available for traditional computer systems. Often rotation is not even used in these games, making it possible to use very simple navigation. For example in most platform games, all a player can do is move left, right, up and down. Positioning can in that case be done simply by using 4 keys or by moving the mouse. In 2D environments for multi-touch systems the task of positioning can be accomplished, simply by touching the desired location on the surface. Rotating actions are usually performed by using two touches, where the line through the two points is used as the direction of the rotation [10]. These input methods for navigation are much easier then those for 3D environments, making a 2D environment the preferred setting for a multiplayer game on the DiamondTouch.

4.4 Tasks to perform
Navigation is just one task that can be performed in games. Often there are a lot of other tasks a player has to execute in order to win or perform better. One can think of a player that has a number of items at its disposal which can be used during the game. To use one of the items, the player has to select and/or activate it. Apart from navigation, this would be one of the extra tasks a player has to perform during the game. For each of these extra tasks there is also some input method required to execute them. In games for traditional computers systems this is done by pressing one or more keys and/or using the mouse to select options from a command panel or menu.

Since the DiamondTouch does not have buttons, this will limit users to interact only via the surface of the DiamondTouch. If buttons still are a desired metaphor (they are common in the graphical user interfaces of most operating systems), virtual buttons can be displayed on the surface. A problem may occur if the buttons can obscure important information, for example, if they are displayed over a part of the playfield. Therefore the amount of buttons should be kept to a minimum, although this will reduce the number of actions that can be performed during the game.

A designer of multiplayer games has to take care that these actions can be performed quickly and do not require too much attention from the player, so the player will not be distracted from the main task in the game. Players have to be aware of the actions of other players, since these might have an influence on the players performance. Therefore if the player has to direct too much attention away to perform certain tasks, it might have a negative impact on the players performance, especially if the game is fast paced.

For the DiamondTouch this requirement is even more important because there is usually a lot of interaction between the players, so a player should not be bothered by having to perform relatively time consuming actions that draws the attention away. Therefore the input required for these tasks should be kept as simple as possible, so complex gestures should be avoided. For the situation where the player has items at its disposal, a good input method would be to display them as icons, close to the physical location of the player. The player can then select and activate them simply by touching the icon that represents a certain item.

The interaction methods found in traditional systems, using menu windows are no good candidate for games in which a player has to be able to quickly respond to certain events. This is because making a selection from a menu is relatively time consuming, and requires too much attention from the user. Also there is no keyboard available to quickly execute an action with one key press. Because a game designer for the DiamondTouch cannot make use of a keyboard within the game and menu windows are not suitable, the designer is limited to only a few actions that a user can perform during game. As a result the game itself should not be too complex (requiring many different actions). However if time is not an important factor within the game, the complexity of the game can be increased, as users have enough time to perform more complex interactions.

4.5 Goal and interaction

On an abstract level the goal for the players of a game is to achieve the highest performance. This is not to be confused with the goal of the game itself, which is of a very different nature (the game might for example be for educational purposes or just for entertainment). The performance measure used to determine how well a player is doing, is of little interest when designing a multiplayer game for the DiamondTouch, since there is no difference in this aspect compared to traditional computer systems. It is more interesting to consider how the performance of a player is affected by the other players.

If the performance of a player is not affected by the other players, there is no interaction between the players in the virtual world. For example consider a race track, where the goal is to be the first to complete three laps. If the racecars of the players cannot collide with each other, there is no virtual interaction between the players, since the actions of one player do not influence the performance of another. As a result there might also be little interaction in the real world between the players (for example shouting because some player just hit your car). Since multi-touch systems are among others appreciated for their ability to support rich social interaction [6], the game should try to stimulate this. Therefore virtual interaction between the players is desired.

When the players can have an influence on the performance of other players, a distinction can be made between competitive and collaborative games. This is an important distinction because the DiamondTouch was developed in a need to improve collaboration in workspaces. The face to face setup allows for easy communication between the participants, which is essential for effective teamwork. Also multiple users can interact with the system simultaneously, so group members can work on a specific task at the same time. These features make the DiamondTouch very suitable to support collaborative multiplayer games. However, there is no reason to think that it is less suitable for competitive multiplayer games.

In competitive games, players compete and try to outperform each other. While doing so a player might be taking actions to increase the performance of the player itself and actions to decrease the performance of other players. In the previous example of the race track, a player could try to push its opponents off the road to decrease their performance.

If players have to work together to win the game, this game will be classified as a collaborative game. For such games, there is a global performance instead of individual performances. There can also be mixtures of competitive and collaborative games, for example if the players are split into teams.

The DiamondTouch amplifies the difference between competitive and collaborative games, by extending it from the virtual domain to the real world. In the virtual domain the difference is has already been mentioned; for collaborative players can perform actions to improve the performance of other players, where in competitive games actions can decrease the performance of other players. Since the DiamondTouch stimulates person to person interaction the difference is also noticeable in the real world.

For collaborative games, players will be encouraged to discuss their actions and plans. After all they have to work together to achieve the best possible performance. On the other hand in competitive games, human to human interaction will be quite different. Players may have to hide information which could be beneficial opponents. This can for example be information that is displayed on the surface, so a player has to physically block the view for its opponents. In turn this may cause the other players the become very interested and they may try to find a way of getting to know this information anyway, for example by trying to push the blocking player away.

In a good multiplayer game design for the DiamondTouch such human to human interaction is being used to improve the game experience. The designer therefore should try to find a way of encouraging interaction in the real world (which is made easy with the DiamondTouch).

4.6 Session lapse

Another important aspect is the course of the game. This includes the initiation of the game (how the game is started), the joining and leaving of players during the game, whether players have to wait during the game and when the game is ended. As seen before with other game aspects, there are a large number of variations possible, with the most suitable choice depending on the nature of the game. In general the most suitable choice depends on the pace of the game, and the number of players involved.

For fast paced games, short game sessions are preferred. This is because such games require a high level of alertness making long sessions exhausting. Also since fast decision making is vital in such games, the actions to perform are limited and relatively simple, which could make the game boring if a session lasts too long. A typical game session would last in the order of several minutes. When the game is not fast paced and complex decisions have to be made, such as in strategy games, longer game sessions are more appropriate (sessions last in the order of hours). The pace of the game is not of great importance when it comes to designing a multiplayer game for the DiamondTouch, however the length of the session is. When the users are standing around the DiamondTouch, long game sessions are impractical because it is relatively uncomfortable to be standing for a long time. If instead the users are seated around the surface longer game sessions are no objection from an ergonomic point of view.

The number of players in the game has an influence on how the game is started and whether players can join or leave during the game. If the number of players is large, for example in MMORGPS like World of Warcraft1, it is appreciated if players can join and leave the game whenever they want. In a game like this a session can also be started, without having to wait for other players. This ad-hoc approach is not suitable for the DiamondTouch, which is designed to support a maximum of

only four players. Players that join later during the game usually have a disadvantage over those that were already playing it. Because of the relatively small amount of players supported by the DiamondTouch, such disadvantages would make it unattractive for a player to join a game that is already in progress. Therefore it is better to start the game when all participants are available and not to accept new players.

4.7 Conclusion
From the game aspects that were reviewed in this section the following can be concluded:

- To make use of the ability of the DiamondTouch to distinguish users, the game should be real-time instead of turn based.
- The most suitable point of view for a game on the DiamondTouch is a top down view, because it is the best to give an overview of the game state in which all information is visible that is important for each player.
- For easy navigation within the game, two dimensions rather than three should be available.
- Actions that a player has to execute during the game should be kept simple and limited in amount. The faster the pace of the, the more important this becomes.
- Interaction between players in both the virtual domain and the real world is desired, because this improves social experience during the game. The DiamondTouch is well suited for collaborative games, because it allows for easy communication between players. However since this is kind of trivial, and no reason has been found to think that competitive games are less suitable, I will design a game that is of competitive nature.
- The duration of the game should be in the order of minutes.
- If the game is to be started, all participants should be present and ready to play. During the game new players should wait until the current game is over, or replace an active player.

5. GAME DESIGN
Using the results found from the study of the DiamondTouch characteristics and the important design aspects found in the previous section, a design for a multiplayer game on the DiamondTouch has been made.

The basic idea of the game is much like the historic pong game. Each of the participants of the game are positioned at one of the four corners of the DiamondTouch. On the surface there are circular goals that have to be protected by the player at that corner. At the same time players have to try to score in the goals of the other players. The rest of the surface is an empty space wherein two balls are moving with a fixed velocity. Each player can place walls to reflect the balls, so they can prevent balls from entering their goal and they can try to direct the balls into the goals of their opponents. The balls can collide with other balls and they can also reflect on the sides of the screen. The game lasts for a certain amount of time and at the end the player that has scored the most goals wins the game. Figure 3 shows a sketch of the playfield.

Each of the players is assigned a color for easy visual identification. This color depends on which mat of the DiamondTouch the player is in contact with, and is either: red, green, blue or yellow. The goals are displayed in the color of the player that owns them. Walls placed on the surface are also displayed in the color of the player that placed them. Whenever a ball hits a wall it will receive the same color as the wall it just reflected upon. If a ball enters the goal of player, the player with the same color as the ball scores a point. In case a player scores in his own goal, the player loses a point.

![Figure 3 Sketch of the playfield.](image)

Initially the game starts out with a single ball in the center of the screen, moving in a random direction. A while later a second ball appears in the middle of the screen, which also starts to move in a random direction. Once a ball enters the goal of a player it will be reset to the center of the screen and it starts moving in a random direction again. Whenever a ball is brought into play or reset it will be gray. If a grey ball enters the goal of a player, that player loses a point (just like when a player scores in his own goal).

The amount of walls a player can place on the surface is limited to two walls. When a player places a third wall on the surface, the wall of the player that has been the longest in the game will disappear. Players can place walls by touching two points on the surface of the DiamondTouch. The wall will be placed on the line through these two points. Walls have a limited length, so when the distance between the two points is greater than the maximum length of wall, it will be centered between the two points. Players can only place walls within a certain radius from their goal, to prevent them from bending over the table and blocking the view for other players. As soon as a player touches the table the area in which a player can place walls is highlighted.

To take advantage of the features offered by multi-touch the walls can be used as paddles. While the player is still touching the surface on two points, the location and rotation of the wall can be adjusted, making it act like a paddle. It is also possible to control the position and rotation of both walls at the same time when the player touches four points. This can be accomplished by touching the table with two fingers with each hand.

6. PROTOTYPE
I have implemented a prototype of the game that was described in the previous section, to see if it was a suitable multiplayer game design for the DiamondTouch. The prototype was implemented in the Java programming language and used a multi-touch abstraction framework [14]. This framework has a driver for the DiamondTouch. Unfortunately the version of the framework available to me at the time of writing did not support tracking and suffered from the ambiguous touch point problem described in section 3.5. Touching the surface with two fingers
resulted in the observation of four touch points by the framework. If tracking techniques would have been applied, the system would be able to filter the phantom touch points.

A screenshot of the game prototype with two players is displayed in Figure 4. This screenshot shows a game with two players, one in the upper left corner and one in the lower right corner. As can be seen, the player in the lower right corner has scored two goals. Touch points which where used to create a wall are marked with small circles and a thin line is drawn through them. This was done so the players would be able to clearly see how their walls (the thick lines) were placed.

Figure 4 Prototype implementation

While playing the game we immediately noticed that touching the DiamondTouch surface with two fingers resulted in the placement of two walls rather than one. The two walls were of equal length and were either parallel to each other or making a cross. This was a direct result from the ambiguous touch point problem and although it was not meant to happen the game was still playable to a certain degree. However instead of carefully placing walls the participants started to wave erratically around a ball in the hope that it would create walls in that area to change the color and direction of the ball. After a while this also resulted in little fights (of friendly nature) between the participants, who were trying to push each other away from the ball. Even though the game could not be played properly and I was unable to verify whether the design of the game was suitable for the DiamondTouch, it was clearly successful in stimulating human to human interaction. It was also possible to notice a small delay, in the order of a tenth of second, before the location of touches were updated (placement of walls). Fortunately this delay did not appear to much of a problem to the participants, so the response time of game was sufficient.

7. CONCLUSION

In this research I tried to find an answer to the question what design aspects are important for a multi-touch multiplayer game on the DiamondTouch. First the distinguishing characteristics and the limitations of this system were studied. The most important results were that it is able to detect simultaneous touches and can identify users. Also users should remain on a fixed location and interaction with the surface for a specific user should be kept in proximity of that user. Finally tracking techniques are required if gestures other than single touches or bounding boxes are to be detected.

Secondly these characteristics and limitations were used to find a set of related game characteristics and for each of these characteristics suggestions have been made for a game design. These form the design aspects that are being searched for and were summarized in section 4.7. For completeness they are presented here again.

- The game should be real-time instead of turn based.
- A top down view is the most suitable
- Navigation should be in two dimensions.
- Actions should be kept simple and limited in amount.
- The game should stimulate human to human interaction.
- Game sessions should last in the order of minutes.
- Players cannot leave or join while the game is in progress.

I designed and implemented a simple pong like game using these design aspects as a guideline. The prototype of the game was meant to verify whether the design aspects that were found would indeed be able to create an enjoyable game that used as much of the features offered by the DiamondTouch as possible. Although I was not able to fully test the game, initial experience has shown positive results. One of the characteristics of the DiamondTouch, the ability to detect limbs just above surface, was not used in the game design. This is because the maximum distance that could be sensed was too short for to find an application for it.

The design aspects found in this research should be used as a guideline, but should most certainly not be viewed as the absolute truth. In fact a more systematically experiment is desired to test the influence of the alternatives for each game characteristic that was discussed in section 4. It is very likely that the selection of one alternative might cause the suitability of alternatives for another game characteristic to change. For example if the game pace is slow, players would have more time available to make decisions and to perform more complex interactions. This would make it possible for the designer to provide players with more actions that can be executed during the game.

While reviewing the limitations of the DiamondTouch that have been found in this research, a question arises if it possible to overcome these limitations. Considering the limitation that users should remain on a fixed location, one might wonder if this is really necessary. In fact it is possible to relax this restriction a bit if the receiver wires are directly attached to the users, instead of being attached to mats that have to be in contact with the users. Of course this does not completely solve the problem, but it would give users more freedom to move around the table. Another important limitation of the DiamondTouch is that it is unable to detect the shape of a pose (from for example a hand). This poses a restriction on the type of gestures that can be recognized and thereby limiting the interaction possibilities. Some other multi-touch systems, such as the SmartSkin system [8], are able to detect these shapes but are unable to identify users. Perhaps it is possible to combine the DiamondTouch with technology from a system that can perceive a full 2D image of the surface. These questions could be answered by doing a research on how to overcome the limitations of the DiamondTouch.

This research has focused on the design of a multiplayer game for the DiamondTouch. Another interesting question is in what degree the results would be different if another multi-touch system was chosen for the research. This could also be investigated in a future research project.
REFERENCES


