Measuring the Effectiveness of Forced Collaboration on Multitouch Tables

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) (Software)

Author:	Michael Hudson
Supervisor:	Shamus Smith
Date:	Friday, 6 November 2015
University:	University of Newcastle, Australia
Faculty:	School of Electrical Engineering and Computer Science
Word Count:	11,250

Signed:

Student: Michael Hudson

Signed:

Supervisor: Shamus Smith

Table of Contents

1 Abstract

In educational contexts the prevailing wisdom in recent decades has been that collaborative learning is one of the most effective methods of teaching, with higher knowledge retention and greater levels of learner engagement resulting from collaborative learning. This project aims to determine whether or not learners can be forced to collaborate, and what effect that may have on the learner's perception of the activity and their partners. This will be shown through the development and testing of a multitouch game that is able to be run in multiple modes, one that forces players to collaborate in order to complete it, and similar mode that is able to be completed by an individual. This study will show that learners can not only be forced to collaborate but that learners actually exhibit a positive response to being forced to collaborate, through an increased sense of working towards a common goal and improved communication The techniques used to force collaboration among the players are simple, and could be implemented or adapted for use in educational contexts through the creation of multitouch educational games.

2 Introduction

The quality of education a person receives over their lifetime is influenced by many factors, due to this the comparison and analysis of different teaching styles and their applications are becoming an area of ongoing interest, with the focus most recently being on collaborative learning (Johnson & Johnson, 2009). Furthermore, recent research such as that by Chen (2008) has acknowledged the benefits of collaborative and computer aided learning styles. With the increasing availability of multitouch tables there is a burgeoning field of research in to computer supported collaborative learning that is facilitated through the use of multitouch tabletops, such as the work of Higgins (2011) and Hsiao et al (2014).

With collaboration being such an effective form of education causing improvements in learner engagement and content retention (Prince, 2004) it seems prudent that methods of improving the amount and quality of collaboration among learners should be researched. This project aims to study whether or not learners can be forced to collaborate and what effect this has on the learners' perception of their partners and the activity that they are completing. This project will attempt to determine this through the study and analysis of learner's interactions with a multitouch application that provides a forced collaboration mode and a non-collaborative mode, and observing the effects this has on the participants.

The results of this study may lead to insights in to the encouragement or forcing of learners to collaborate, as well as the practicality and use of multitouch tables in computer supported collaborative learning contexts.

3 Literature Review

When attempting to learn something new people benefit by attempting that learning with other people as they achieve more learning outcomes than those that learn through individualistic or competitive learning (Gökmen, 2009). By tasking students to cooperate and solve problems collaboratively student engagement and content retention increases, providing a marked benefit for those that learn in a collaborative environment (Prince, 2004).

Thomson et al. (2009) defined collaboration as a process involving ‘formal and informal negotiation’ and ‘involving shared norms and mutually beneficial interactions’. Thomson et al. also note that collaboration is ‘a multidimensional, variable construct composed of five key dimensions’ and provide a statistical model for measuring this collaboration. A number of other studies (Capponi et al. 2010; Liu & Kao, 2007) measured collaboration by analysing the communication patterns used by participants of their study and determining the effectiveness of that communication.

When new technologies are introduced and appropriated to educational contexts ‘over-generalization’ and ‘over-expectation’ of the technology tends to occur (Dillenbourg & Evans, 2011). Dillenbourg & Evans (2011) stated that in order for a new technology to better educate students the application of that new technology will require contextualization and pedagogical goal setting, as well as finding an appropriate place within the learning process. In a modern teaching environment where collaboration among students is desired there is an advantage for students who are taught with access to technology which fosters that collaboration (Liu & Kao, 2007).

With the proliferation of smartphones such as the Apple iPhone and Android devices in modern society there is a possibility that they can be used in order to promote collaboration and learning between people as these smartphones are equipped with multiple wireless communication technologies that can be used for the sharing of documents and information. Liu & Kao’s (2007) findings indicated that in groups of more than two students, where mobile device screens would be difficult for the group to view together, participation among group members increased when the students were provided with a shared display. Liu & Kao (2007) also describe how students were able to discuss and reason about more complex information when provided with a shared display.

While multitouch tables can offer new and exciting pedagogical applications the technology itself is not inherently applicable to any single educational context or method, it is the affordances offered by the multitouch tables that present the greatest opportunity for educational contexts (Dillenbourg & Evans, 2011). The inherent multi-modal and social nature of multitouch tables (Dillenbourg & Evans, 2011) is one of the defining features that differentiate multitouch tables from personal computers in educational contexts. Hsiao et al. (2014) showed that students were able to learn more effectively when sharing a multitouch table than when sharing a personal computer.

Scott, et al. (2002) observed that students using individual displays rather than a shared workspace suffer as a result of a lack of eye contact and loss of visual focus due to difficulty in maintaining a consistent mental model of the problem at hand and the inability to easily resolve conflict when each user's mental model of the problem is inconsistent. The immediate and accurate removal of ambiguity that is afforded by simply being able to point at something shows how effective and succinct interpersonal direct interaction is (Frohlich, 1993), and how useful such a tool could be in aiding collaborative learning.

Students that have access to tools and environments that foster collaboration learn more effectively than those that don't. The students that work collaboratively also have greater knowledge retention and participation rates (Chen, 2008; Gökmen, 2009; Hudson, 2004). Higgins et al. (2011) categorised a selection of literature that explores the use of multitouch tables in educational contexts and found that the existing research suggested that multitouch tables can have a great impact on the communication patterns of participants and 'the importance of designing the multi-touch table activities … in such a way as to maximise the types of interactions that support learning'.

Given all of the benefits that a collaborative learning process provides and the inherent social nature of multitouch tables this project aims to provide a number of methods of not only enabling collaboration from users, but demanding it through the development of new software. By requiring users to collaborate and recording the interactions between participants we will be able to determine if it's possible to design software that demands effective collaboration from users and what effect this has on the users' perception on the activity and their partners.

4 Methodology

For the development of the prototype other than some initial project establishment an iterative development cycle was used. An iterative development cycle was chosen to allow the rapid evolution of a prototype without requiring a lot of design and research up front which may have been scrapped later due to an evolving understanding of the problem since 'Facilitating change is more effective than attempting to prevent it' (Fowler & Highsmith, 2001).

4.1 Establishment and Design

The first stage of the project was focused on the design of the prototype as well as allowing time for technology research and evaluation. During this stage basic paper prototyping and whiteboard outlines were used to aid in the design of the prototype. Technological research was undertaken in order to determine what technology may aid in the implementation of the designed prototype, including language and framework choices and tooling such as source control and build systems.

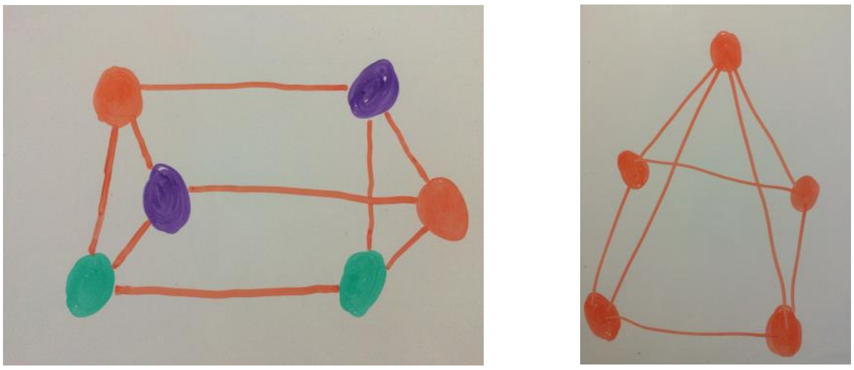


Figure 4.1 Whiteboard prototypes of the levels 'Prismatic' and 'Pyramid'

Confluence[1], a collaborative online document editing tool, was used heavily during this period for the maintenance of all living documents, such as the design document, meeting minutes and a diary of design decisions and tasks that were completed.

[1] Available at https://www.atlassian.com/software/confluence/

4.2 Develop

This stage was focused on the implementation of the prototype designed in the first stage. Development of this prototype was done using an iterative development approach (see Figure 4.2). All code was tracked using a hosted distributed version control service; namely Bitbucket[2]. Bitbucket was chosen as the source control management system primarily due to my familiarity with it and the relative feature-parity it has with the primary competitor, GitHub[3]. The issue tracking system of Bitbucket was used in order to keep track of major tasks that needed to be completed as well as to keep track of any bugs that could not be fixed in a trivial amount of time.

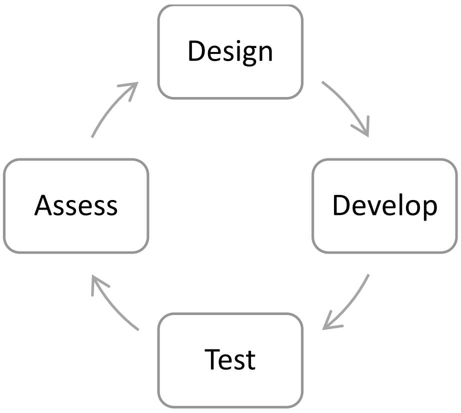


Figure 4.2 The iterative development cycle

4.3 Test

The Test phase in Figure 1 included both automated unit tests which are run as part of a regular build and hallway usability tests, which is when one tests a feature of a prototype using someone close by in order to get a quick idea of what improvements need to be made. The 'Assess' phase is an opportunity to reflect and consider the newly developed feature and consider what effect it had on the developing prototype.

[2] Available at https://bitbucket.org/

[3] Available at https://github.com/

4.4 Assess

The assessment phase makes us of the feedback and knowledge gained from the test phase and is used as a time to reflect on the effects that the recently tested feature has on the product or project as a whole. During this phase decisions are made about what to design next, such as extending the recently tested feature, tweaking it or scrapping it entirely. As feature requirements were created or modified in this phase they would be updated in a Confluence document used to keep track of what needed to be developed, and would drive the action taken in the Design phase.

4.5 User Study

This project required the use of a user study in order to determine the how effectively the prototype that was developed encouraged collaboration among the participants. These tests were designed to provide the data necessary to make an informed decision on whether or not forced collaboration is effective, or perhaps provide evidence to suggest that demanding collaboration from users does not work.

This would typically require ethics committee approval due to the use of human subjects, but as participants could only be chosen from the Software Engineering Final Year Project Group this was not necessary. Participant consent was still required for the user study due to video recording of the user testing sessions with the prototype being necessary in order to determine when the participants were collaborating.

A Likert-like questionnaire was also employed in order to gauge each participant's perception of the activity and the other participant completing the activity with them, see section 7.1 User Study for further details.

4.6 Analyse

The data collected during the User Tests stage of the project required analysis in order to allow a judgement on the problem statements to be reached. This analysis primarily consisted of tabulating and aggregating the results of the Likert-like questionnaire and transcription of the footage from the user study to a collaboration sequence diagram.

5 Design

5.1 Requirements

In order to ensure that the application being designed would be fit for purpose a number of requirements were created in order to guide development. These requirements were chosen as a direct result of research in to what available technologies were capable of and what previous studies in computer supported collaborative learning had done.

5.1.1 Functional Requirements

Functional requirements are requirements that describe parts of the system, and how it will interact with other parts of the system. These describe features to be implemented, most of which are necessary for the application's development to be considered a success.

Id	F01	**Name**	Display Dots
Description	Dots have been chosen as the primary element that participants will interact with. These Dots should be of reasonable size so that users of the application will be able to easily see them on the target platform display. These Dots should also be able to be arbitrarily coloured.		
Justification	As this application will be focused on multiple concurrent users interacting with the system on a multitouch table they will likely be viewing the application from multiple angles. Dots, which are circles, are able to be viewed from any angle and appear to be the same. This will aid in maintaining a shared mental model of the problem which Scott (2002) observed to be beneficial to users.		

Id	F02	**Name**	Dragging of Dots
Description	The system will need to be interactive in some way in order for there to be any way for the participants to actually be able to use the system.		
Justification	Dragging is a simple action that maps easily to the expectations of users when dealing with objects on a plane.		

Id	F03	Name	Handle dragging events for multiple Dots
Description	As the application is targeted to groups of people it will need to be able to handle dragging events occurring on multiple Dots simultaneously.		
Justification	The application will not be able to test collaboration among users if the application doesn't allow them to perform actions simultaneously.		

Id	F04	Name	Passing Dots between participants
Description	If a player is holding a Dot and wishes for another player to take control of it without having it return to the centre of the play area (see requirement F08) then those players will likely wish to transfer the Dot in some unoccupied space in the play area.		
Justification	The game's levels will increase in difficulty (see requirement N05) as the game progresses. Part of this increase in difficulty is the introduction of more Dots on screen that must be managed at the same time. It is expected that one player will have mistakenly grabbed a Dot that may actually be required by another participant and that they wish to pass this Dot to that other participant. In order to make the application align to a player's expectation that the system behaves similarly to a real world model of the system multiple players will need to be able hold on to a Dot at the same time without the Dot returning to the centre of the play area until all of the players have relinquished control of the Dot.		

Id	F05	**Name**	Display Shadow Dots
Description	There should be an obvious goal for the players to strive towards. This should be facilitated through the use of Shadow Dots, which appear similar to the Dots that the players can interact with but are different enough that they don’t get the two types of Dot confused.		
Justification	Similarly to requirement F01 Shadow Dots are equally observable and understandable from any angle. Shadow Dots also provide a clear indication of what Dots they are willing to accept as a solution, as the colours of the Shadow Dots and Dots are similar.		

Id	F06	**Name**	Display Connections between Dots
Description	Dots should be able to have Connections with other arbitrary Dots. These Connections should appear as solid lines connecting the Dots, and these lines should follow the Dots accurately as they are moved by the players.		
Justification	Connections between Dots adds an additional layer of complexity to the levels. Without this additional complexity the application would simply be an exercise in manual dexterity, which wouldn’t work towards answering the problem question of this study.		

Id	F07	**Name**	Display Connections between Shadow Dots
Description	Similarly to F06 the Shadow Dots in the level must be able to display Connections between themselves.		
Justification	As the goal of each of the levels is to match a jumbled pile of Dots to a matching set of Shadow Dots the Connections between the Dots that the players interact with will need to have some sort desired configuration.		

Id	F08	Name	Dots spring back to centre of play area
Description	Once all touches of a Dot have been released the Dot should return to the centre of the play area. This should occur as a smooth animation so that players understand where the Dot they just released went.		
Justification	The chosen method of forcing the participants of the study to collaborate was to require that they would both need to have their hands on the table in order to actually complete the level. By both having a common goal that neither participant can complete individually and by needing to engage in 'formal and informal negotiation' (Thomson et al., 2009) it can be said that the participants are collaborating.		

Id	F09	Name	Game level definitions
Description	The levels of the game should be defined in a format that makes it easy to add more levels to the application.		
Justification	Hardcoding all of the level definitions by hand within the application would eventually lead to difficulty in modifying the code should something about the levels change. It would also make it difficult to reorder the levels if needed.		

Id	F10	Name	Level completion detection
Description	When players arrange the Dots in a valid configuration such that they match the underlying Shadow Dots sufficiently (exact positioning of Dots is not necessary) and Connections between the Shadow Dots then the application should be aware that the level has been completed successfully.		
Justification	As the goal of the game is to align a set of Dots with a set of Shadow Dots such that their configurations match it is vital that a reliable method of determining when this situation occurs.		

Id	F11	**Name**	Level progression
Description	When the Dots are placed in to a valid configuration that fulfils requirement F10 then the next level should be loaded if one exists.		
Justification	Having the users complete a single level would likely not provide any meaningful results.		

Id	F12	**Name**	Record time taken to complete levels
Description	The time taken for players to complete a level should be recorded so that it can be analysed.		
Justification	It may be interesting to see if there is any correlation between the time required to complete a level and the collaboration settings of the application for a given group.		

Id	F13	**Name**	Method to switch collaboration on and off on a per-level basis
Description	The application should have a method of enabling or disabling forced collaboration on a per-level basis.		
Justification	As it may not be known at design time whether a level will be used for the forced collaboration mode or the non-collaborative mode there must be a mechanism in the application or level definitions (see requirement F09) to allow the mode to be easily selected.		

Id	F14	Name	Summary page of results
Description	A page displaying a summary of the results recorded by the application.		
Justification	In order to record the results from the activity it would be nice to have a well formatted and easily digested summary of any data recorded, such as time taken to complete a level (see requirement F12).		

Id	F15	Name	Show messages to the user to request an action
Description	The game should stop, preventing user input so that any important messages can be displayed to the participants.		
Justification	At several points in the application it will be necessary to inform the users that some milestone has been hit. See Section 7.1 User Study for more details.		

5.1.2 Non-functional

The following non-functional requirements are criteria that can be used to define the operation of the system, or requirements that are related to performance or user experience design. The absence of one or multiple of these requirements in the finished application may not result in the application not being fit for purpose, but may result in the application providing a sub-optimal experience for the developers or participants in the user study.

Id	N01	Name	Game should feel responsive
Description	As players interact with the system the Dots should accurately track the touches of the participants. The game shouldn't appear to be jerky or slow to react to user input.		
Justification	It is beneficial for the participants if the application feels good to use, and part of what makes an application feel good to use is that the user feels like they are in control and that the system is behaving as they expect.		

Id	N02	**Name**	Easily deployable application
Description	The application should be deployable with minimal amount of difficulty.		
Justification	As there is no guarantee what environment the participants will be completing the user study in it is important that application can be up and running as quickly as possible so as not to waste the coordinators time unnecessarily.		

Id	N03	**Name**	Ability to rapidly prototype project
Description	The application should be able to be iterated on as quickly as possible.		
Justification	As a student project with very relatively loosely defined requirements requiring the creation of new software it is highly likely that some experimentation in to various technologies or approaches to solutions will need to be made. By shortening the time it takes to complete a full iterative development cycle (see Figure 4.1) the validity and suitability of a developmental direction can be ascertained without implementing an incorrect solution any further than is necessary.		

Id	N04	**Name**	Application is accessible to colour blind participants
Description	Colour blind participants should be able to complete the game.		
Justification	As the colour blindness of the participants is not known and there is a disproportionate number of males participating in the user study it would be prudent to design the levels with this consideration in mind. With 1 in 12 males suffering from some form of colour vision deficiency (Rigden, 1999) there is over a 50% chance that one of the participants in this study will have a form of colour blindness.		

Id	N05	**Name**	Level difficulty should increase in each mode
Description	Each mode will contain a separate set of levels, each of these sets of levels should get more difficult as they are completed.		
Justification	As players become more familiar with the activity and come up with their own strategies for completing the levels they will become more efficient at completing them. One of the primary ways in which a game can be considered fun is by introducing more difficulty that challenges the player without alienating them.		

Id	N06	**Name**	The goal of the levels should be obvious
Description	The players should understand what state the game needs to be in to proceed, and should only have difficulty in determining how to achieve that state.		
Justification	Obstructing the solution to a puzzle from players in an unfair way frustrates users and doesn't actually test their ability to complete the puzzle.		

Id	N07	**Name**	The application should run on a wide variety of hardware
Description	The application should run on a reasonably wide variety of hardware.		
Justification	Similarly to requirement N02 the destination environment for the application cannot be known for certain. For this reason it is recommended that the application be able to run adequately on moderately powerful hardware.		

5.2 Design Language

In designing the application a design language was created early on to refer to the core concepts of the game.

5.2.1 Dots

Dots are the only form of interaction that the players have with the game. They are richly coloured circles that begin the game in the centre of the play area. As seen below in Figure 5.1 the Dots have a shadow applied to them, which is used to give them a sense of depth and thickness, further cementing the idea in the users mind that these Dots are physical entities that exist on the other side of the screen that the user can interact with. This follows from the material design philosophy[4] which focuses on user interface design that is grounded in physical reality.

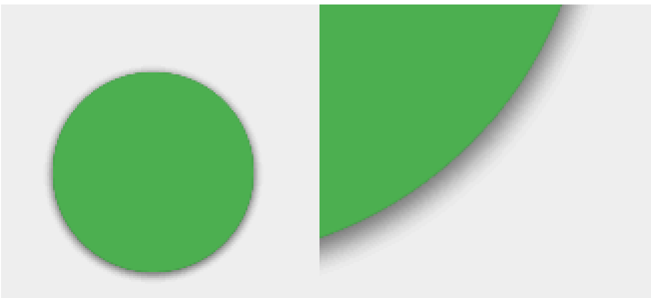


Figure 5.1 A Dot, and a closer look at the applied drop shadow

[4] Google's Material Design principle - https://www.google.com/design/spec/material-design/introduction.html

5.2.2 Shadow Dots

A Shadow Dot is very similar to a Dot, but they exhibit paler colours than a Dot and are not able to be interacted with by the user. By using a paler colour there is a clear differentiation between which elements of the game are interactive and which elements are static.

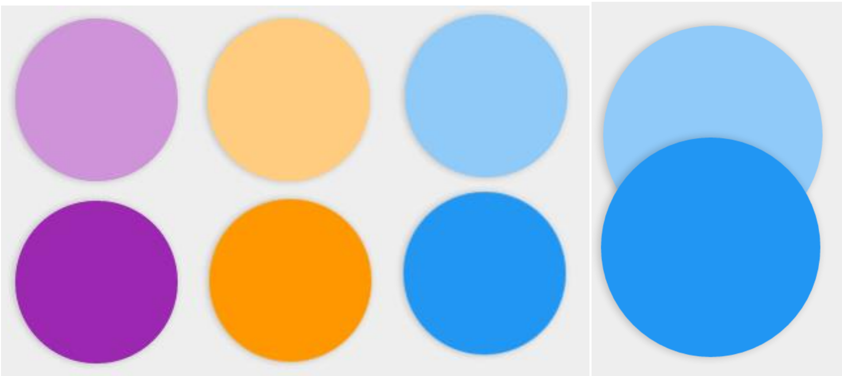


Figure 5.2 Dots and matching Shadow Dots

As shown above in Figure 5.2 Dots always appear in front of the Shadow Dots. This is to prevent the Shadow Dots from potentially obstructing the Dots which the players need to be able to interact with and goes towards the partial fulfilment of requirement N06.

5.2.3 Connections

Connections are lines that sit between Dots and Shadow Dots in order to provide a greater level of challenge to the players, who in order to complete the puzzle need to arrange the Dots in to the same configuration as the Shadow Dots. These Connections are able to connect any two Dots, regardless of colour differences or existing number of connections.

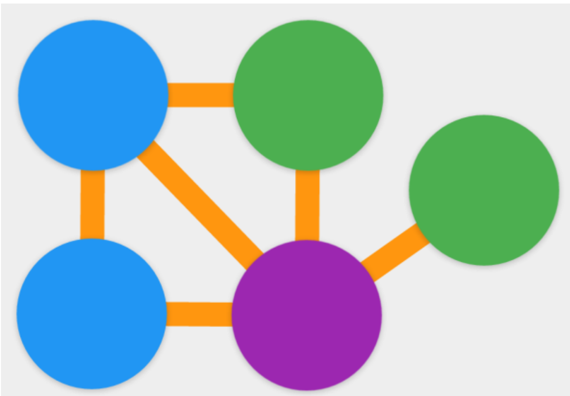


Figure 5.3 Connections linking multiple Dots

5.3 System Interactions

The actual gameplay mechanics can be modelled in sequence diagrams in order to show the possible states the game can be in, and what interactions should occur within those states. The following sequence diagrams show how the users, the user interface and the game logic interact with each other throughout the applications use.

5.3.1 Normal Play with Forced Collaboration

The main gameplay loop consists of the users touching and moving Dots and attempting to match them with the Shadow Dots configuration displayed beneath them. This set of interactions is quite simple, and is the core gameplay mechanic that players will experience while using the application. The interactions are very similar when there is no forced collaboration (see Appendix F).

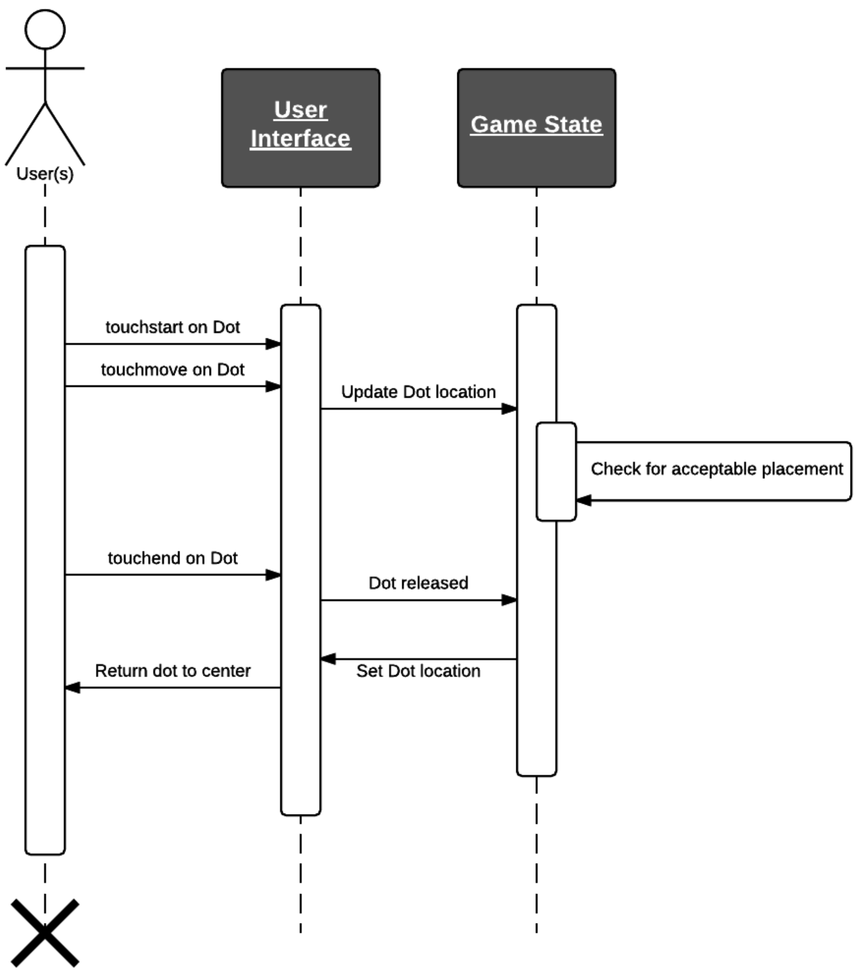


Figure 5.4 Sequence diagram of normal gameplay

5.3.2 Game Winning State

When the players manage to match the configuration of the Dots with the underlying Shadow Dots configuration the gameplay will progress in the same manner regardless of whether the players are being forced to collaborate or not.

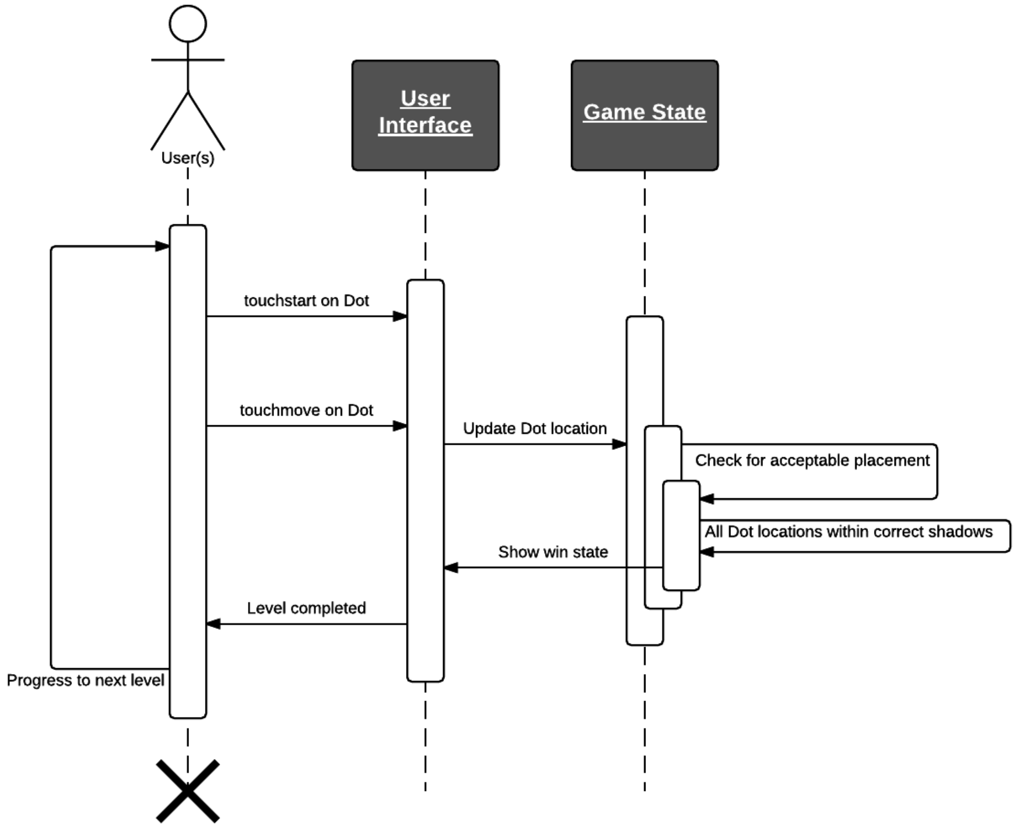


Figure 5.5 Sequence diagram of a game-winning state

5.3.3 Player-to-player Interactions

There are other interactions that the players engaged in with the system that focused on indirectly changing the game state through interactions with their partner. One such method of this interaction is the ability to pass Dots between players (see requirement F04).

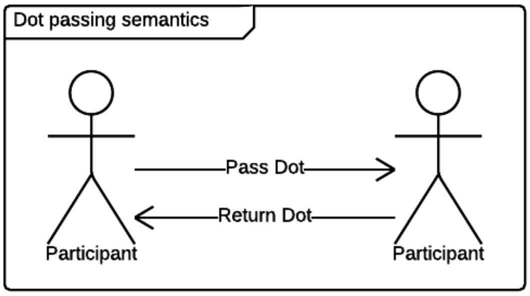


Figure 5.6 Interaction diagram of Dot passing

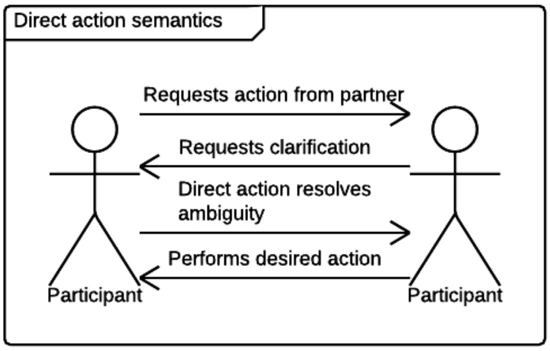


Figure 5.7 Interaction diagram of direct action

Another interaction that players undertook was the clarification of requests for action. This most commonly presented in one player requesting that their partner move a Dot to a specific Shadow Dot's location. As it was common that there were multiple Dots of the same colour in the level at the same time players had difficulty uniquely identifying the Dots through vocal clues alone. This led to the player requesting action to point at the Dot they were talking about with an unoccupied finger during the forced collaboration mode (see section 8.3.3 Removal of Ambiguity for more).

6 Implementation

6.1 Technology

The application was developed as a javascript application, initially based on the code from the Software Engineering Final Year Project. This was done due to the amount of research that was done as part of that project in to the multitouch capabilities of existing libraries such as HammerJs[5] being insufficient for multi-user interactive systems. Similarly to this project the Final Year Project aimed to build a system that would run on as wide a range of hardware devices as possible and was easily deployable.

[5] http://hammerjs.github.io/

6.1.1 Architecture

The application is built on top of Backbone[6], which is a framework used for building data-rich javascript applications. Backbone is currently used to handle the routing of HTTP requests to the correct javascript component, namely, the Dots application being tested. The application also makes use of the Bootstrap Material Design CSS Framework[7] in order to adhere to the material design philosophy. A Node.js[8] web server is used to serve the application to clients through a browser.

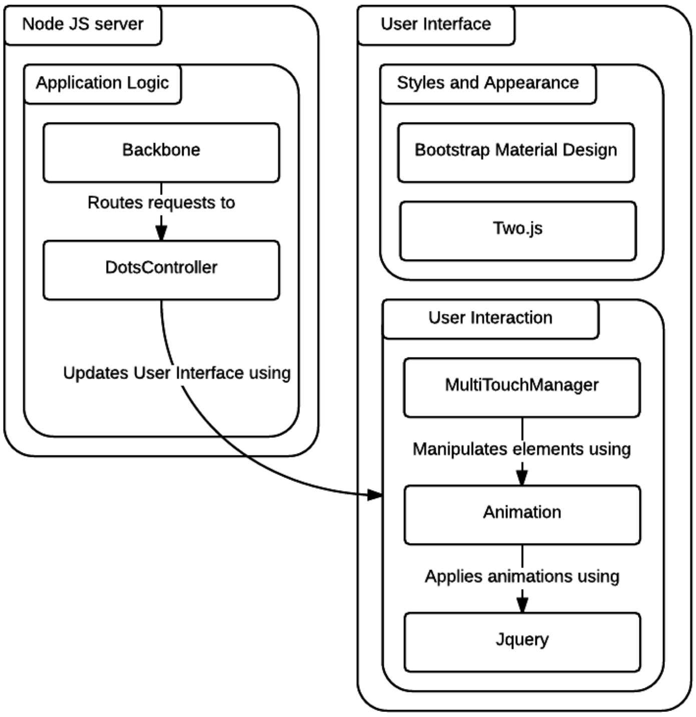


Figure 6.1 Architecural responsibilities of system components

[6] http://backbonejs.org/

[7] https://fezvrasta.github.io/bootstrap-material-design/

[8] https://nodejs.org

6.1.2 User Interface

The Dots application's user interface worked in a series of layers. Users interacted with the system on the Dots layer. As the users manipulated the elements on this layer their positions were updated with associated javascript objects. By translating the state of the user interface in to a core set of objects the game logic could be written such that the javascript objects were considered the source of truth. As players progressed through the levels the Dots layer and Shadow layer would be replaced with the user interface of the incoming level. If any messages were required to be displayed to the users they would be displayed in the status layer, preventing the users from interacting with the Dots layer below it.

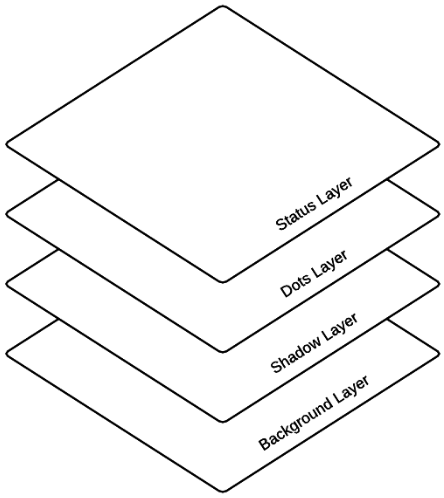


Figure 6.2 User Interface Layering

The Two.js library[9] is used to handle the drawing of Connections between Dots and Shadow Dots. These lines use the colour palette from the material design specification, in order to maintain a cohesive aesthetic with the rest of the application.

[9] https://jonobr1.github.io/two.js/

6.1.3 Existing System

The development of the application was initially based off of the Software Engineering Final Year Project. The Final Year Project was a multitouch application developed for the Nursing and Midwifery faculty in order to enhance their tutorial structure. This application seemed to have much of the code in place that would be required for this study, however it required heavy refactoring, and much of the technology being used was not required for this project. Due to this much of the architecture was carried over. While this initially accelerated development over time as the two projects diverged it became too difficult to bring over features from one system to the other, and so the projects were simply allowed to diverge.

The primary components of the existing system that were leveraged were the MultiTouchManager and Animate components. The MultiTouchManager was a system that allowed HTML elements to have behaviours attached to them that would handle the standard browser events that occurred on the element. A behaviour was simply a collection of functions that would be called whenever an appropriate event occurred on the HTML element, which for this project was exclusively Dots. The most prominent example of these behaviours being applied is the DraggableBehaviour, which when applied to an element would allow that element to be dragged by the user. Other behaviours exists, such as a RotateTranslateScaleBehaviour, which is similar to the DraggableBehaviour, but handles multiple touches differently, allowing rotation of the HTML element and allowing the element to grow and shrink when a pinch gesture was used on the element.

The Animate component abstracted much of the animation of HTML elements away from the calling code. Animations such as movement and scaling of elements required direct interaction with the HTML element's CSS properties, which is not the responsibility of calling code such as the MultiTouchManager. This modularity and separation of responsibilities means that each module is focused on providing only one function or feature to the product. This also means that should a new animation module or new module for managing touches be required that only the component that provides that functionality needs to be updated

6.1.4 Build System

Gulp[10], a javascript based build system was researched and integrated with the project during the establishment phase. Gulp made it easy to watch for changes to source files and then recompile and redeploy the application as that change was made. The BrowserSync[11] plugin, which allows the injecting of resources or live reloading of an application when changes are made, was also employed, allowing testing to occur on a multitouch device connected to any of the development machines over a wi-fi network. As code changed a network connected Microsoft Surface Pro 2 or iPad Mini would reload the application. This allowed for extremely rapid prototyping to occur for multiple devices.

6.1.5 Deployment

As the application was being served by a Nodejs webserver the application was able to be deployed to and tested on multiple operating systems, including a Windows 7, Windows 8.1, Windows 10, OS X and Amazon Linux EC2 instance. On systems where the code could be retrieved and the correct dependencies installed the code could be updated and tested rapidly. When performing the user study this property was not necessary, however it was vital that the user test be able to occur from any computer that the multitouch tables were connected to. In order to facilitate this the application was deployed to an EC2 instance on Amazon's cloud hosting solution, AWS[12], allowing any internet connected computer to load the application in a web browser.

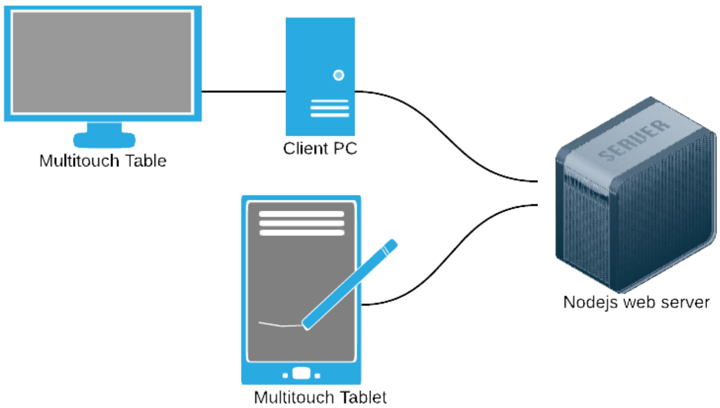


Figure 6.3 Deployment architecture diagram

[10] http://gulpjs.com/

[11] http://www.browsersync.io/

[12] https://aws.amazon.com/

6.2 Level Design

Levels for the game were first designed on a whiteboard. This enabled a general feel of how difficult the level may be to be visualised, and also gave a good basis to work off of when converting the drawn image in to a JSON level definition. Many levels could be rapidly designed and potential issues with those levels could be spotted before any time was spent implementing and testing the level. One example of this is the level 'Orion', which drew inspiration from the constellation which is its namesake. Initially this drawing contained many more Dots than the final version of the level, and was much closer to the original constellation's form. Due to there only being two players that would be playing the game at a time having more than ten Dots that may need to be held in place simultaneously would not be possible.

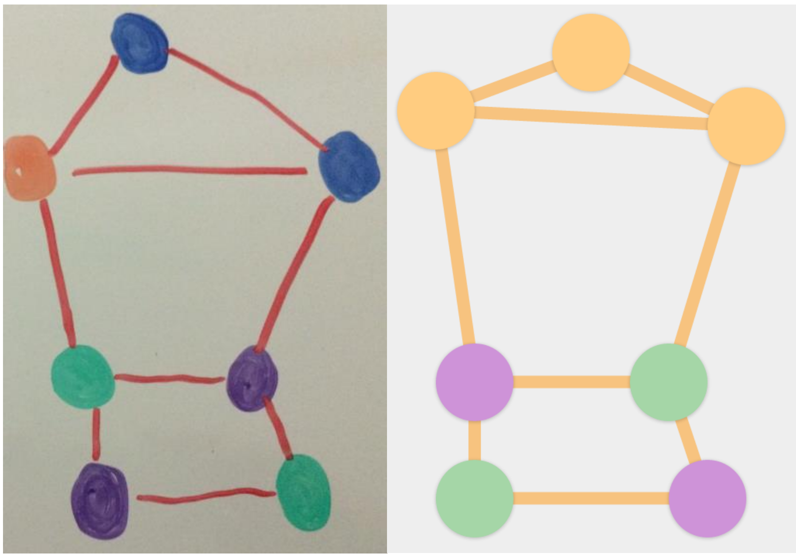


Figure 6.4 The level 'Orion', drawn on a whiteboard and as it appears in-game

Some levels were also designed in order to showcase the specific concepts that the game made use of. Some of these levels became the training levels that all participants would complete at the start of the user study. These levels aimed to teach the players that the colours of the Dots were important, that the Connections between Dots were important, and that these two concepts could be combined arbitrarily in order to make the levels more difficult.

6.2.1 Level Definitions

The levels for the game were stored in a JSON format, which is a basic nested key-value format. Below is an example level definition, several level definitions used in the application can be found in Appendix C. This level definition format was created in order to fulfil requirement F09.

Table 6.1 Level Definition Example

```
{
    levelName: 'Boxxy',
    springBack: true,
    dots: [
        {id: 1, colour: 'red'},
        {id: 2, colour: 'red'},
        {id: 3, colour: 'purple'},
        {id: 4, colour: 'purple'}
    ],
    shadows: [
        {id: 1, x: -400, y:  300, colour: 'red'},
        {id: 2, x: -400, y: -300, colour: 'red'},
        {id: 3, x:  400, y:  300, colour: 'purple'},
        {id: 4, x:  400, y: -300, colour: 'purple'}
    ],
    connections: [
        {from: 1, to: 2},
        {from: 2, to: 3},
        {from: 2, to: 4},
        {from: 4, to: 3}
    ],
    accepts: 'allColourAllConnections'
}
```

The ‘levelName’ property is a unique name used to differentiate the levels from one another, despite what order they may appear during the game. This made it easy to tell which levels and in which order the players had actually completed them without simply relying on non-descript integer Ids.

The ‘springBack’ property is what determined whether or not the Dots would return to the centre of the play area when all touches were released. This was implemented as a direct result of requirement F13

The ‘dots’ array holds a number of Dots, which are given an Id and a colour to display. The colours available are taken from the material design specification[13]. By requiring that the colours were taken from the material design specification the code could automatically decide based on whether a Dot or Shadow Dot was being rendered what shade of that colour to apply. For most colours the material design specification lists 10 primary shades of that colour, as well as four accent shades. The Dots were rendered using the 500 level shade of the specified colour. An example colour swatch can be seen in Appendix G. The id property is a unique Id that is used primarily for level acceptance criteria (discussed below), but is also used to create unique predictable Ids for HTML elements, allowing a link between the user interface code and the game logic code to be programmatically determined at runtime.

The ‘shadows’ array hold all of the shadows for the level. These shadow elements are what define the Shadow Dots to be rendered, and consist of the same properties as a Dot, as well as a location to be displayed in the play area, relative to the its centre. This relative positioning is why some Shadow Dots have negative values for their ‘x’ and ‘y’ coordinates. Similarly to the Dot definitions the Shadow Dots are given a unique Id that allows a programmatic link between the HTML element and the object representation in code to be formed.

The ‘connections’ array holds a number of Connections between Dots. These connections objects only hold a pair of Ids for the Dots they connect together, and these same connections are applied to the Shadow Dots with the same Id. For this reason it is important that the Dots and Shadow Dots are defined with matching id and colour properties, otherwise the game will appear to be illogical, or not work as expected. It would then seem logical to define the Dots and Shadow Dots as one object representation rendered in two ways, once as a Dot and once as a Shadow Dot, but this mixes the responsibilities of the definitions in the level description objects. This would also unnecessarily limit the creation of levels with more Shadow Dots than Dots, which was a proposed method of increasing the difficulty of the levels over time.

The ‘accepts’ property can either be a javascript String naming a particular acceptance strategy or an array of possible solutions. Due to time constraints making the development of a more general purpose algorithm unconscionable a number of smaller, more easily testable, suite of acceptance criteria strategies were defined. The possible options for the accepts property are as follows;

---

[13] https://www.google.com/design/spec/style/color.html

Table 6.2 Level completion accetpance strategies

Value	Effect
`'idMatch'`	The level will be considered complete only when all Shadow Dots are sufficiently covered by a Dot with the same ‘id’ property.
`'allColourAllConnections'`	The level will be considered complete when all Shadow Dots are covered by a Dot with the same ‘colour’ property and the number of connections on the Shadow Dot and the Dot covering it are equal. In the above example the two purple Dots are interchangeable as they have the same colour property and are both associated with two elements of the connections array, however the red Dots are not as they have a differing number of connections.
`Array`	If an array is defined for the ‘accepts’ property then that array must define all possible solutions for the level manually. Each possible solution is also defined as an array of objects with a dot and a shadow property, which hold the ‘id’ of a Dot and Shadow Dot pair. When determining if a level is complete the application will loop through every potential solution and determine if the Dot with an ‘id’ matching the dot property is currently within the acceptable bounds of the Shadow Dot with an id matching the shadow property. If any of the possible solutions are completely satisfied, that is, all pairs within that solution are satisfied, then the level is said to be complete.

Table 6.3 Example manually specified level solutions

Array Acceptance Criteria Example
`accepts: [ [{dot: 1, shadow: 1}, {dot: 2, shadow: 2}], [{dot: 1, shadow: 2}, {dot: 2, shadow: 1}] ]`

6.3 System Testing

As this system was going to be used by people that had not seen the system before it was important that the system was tested by novice users throughout the project lifecycle. When a progress milestone had been hit or a feature was mostly complete novice users would be shown the system and asked their opinion on it. This took place informally, but allowed for the discovery of bugs, problems or gaps in the system.

The level of intuitiveness of the system was also ascertained during this novice user system testing, as the users simply had a multitouch device (Microsoft Surface Pro 2 or iPad Mini) placed in front of them and were asked to use the application. It is through this testing that requiring a way to pass Dots between players was discovered, as until that point only a single developer has used and tested the system.

During system testing one of the most heavily tested parts of the system was the acceptance criteria for level completion. Due to time constraints a general algorithm for detecting if the configuration of Dots that the players had made was a winning configuration was not developed, but several smaller strategies were. These strategies and their application to a level had to be tested thoroughly to ensure that the user study would not be impacted by game-breaking bugs.

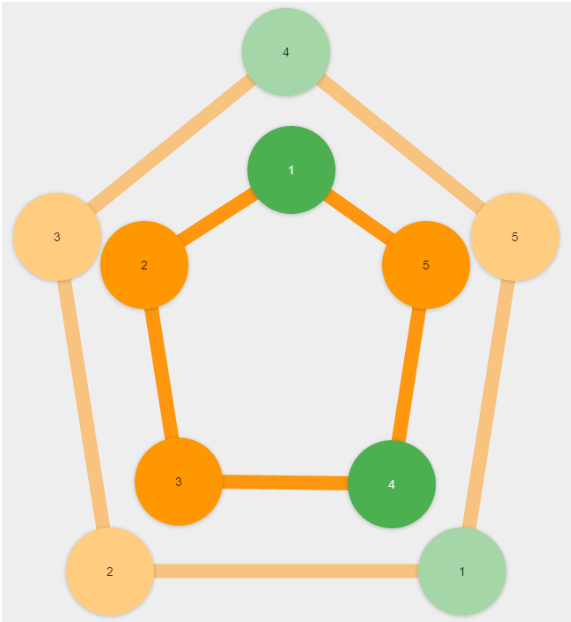


Figure 6.5 Testing a level with Dot and Shadow Dot Ids displayed

During the testing of the level acceptance criteria the Id number of the Dots could be seen, and the game was set to not have the Dots spring back to the centre regardless of their forced collaboration settings (the level's 'springBack' property, see 6.2.1 Level Definitions). This allowed for easier debugging when logic errors seemed to be occurring in the acceptance criteria for the level, as well as being able to complete the level with a mouse as a multitouch device was not always available.

6.4 Current Issues and Future Enhancements

While the system was able to be used in order to test the hypothesis for this study due to increasing time pressure many planned features were cut, and some features that were implemented were implemented poorly in an effort to get them done on time. Without any serious experience in javascript development there were many hard lessons that had to be learnt during the development of the application.

6.4.1 Code Structure

Much of the code was added in as needed as a quick test to see if a given idea was sound and would fulfil the needs of the requirement being implemented at the time. Due to time constraints a refactoring of these solutions never occurred, and so the code is highly unstructured and difficult to read and understand. A lot of the game logic could be moved to functions so that the main game loop would be more readable and responsibilities for certain parts of the game would be more isolated than they currently are.

6.4.2 Inline HTML

The Dots game logic adds raw HTML elements to each of the display layers, and these HTML elements exist within the main code. In this sense the user interface and game logic are actually intertwined throughout. This was done due to issues that occurred when attempting to load HTML elements from external templates that did not seem like they would be resolvable in time to get the application ready for the user study.

6.4.3 Hardcoded Shadow Dot Locations

As seen in the level definition file the locations for the Shadow Dots are hard coded values. While this works fine on a 1080p display, such as the target multitouch table, smaller devices were not able to display all of the Dots on some levels as they were placed off-screen. This mean that initial testing of the multitouch nature of the application was limited to very small areas of interaction as testing was performed on an iPhone 5s, and it was several weeks before the Microsoft Surface Pro 2 and iPad mini were available for testing.

6.4.4 Improper Isolation of Responsibilities

Because the MultiTouchManager maintains its own representation of the element’s location when animating the element manually, such as when returning the Dots to the centre of the play area, the MultiTouchManager’s location values for that element aren’t updated. This resulted in the DotsController module manually manipulating the MultiTouchManager’s location values for each Dot’s HTML element once the animation was completed. This meant that for the 120 milliseconds the animation required to complete it was possible to touch the location where the Dot was initially released and the Dot would stop returning to the centre of the play area. The Dot would then track the new touches position, but it would be offset towards the centre of the play area some distance depending on how much of the original animation had completed.

6.4.5 Poor Data Collection Abilities

At the end of the activity a table displaying a list of the levels completed, whether they were set to be a forced collaboration level and the time taken to complete the level is displayed. This was due to not having the time to set up a more flexible data collection framework that would be able to save these results to some sort of permanent data store.

6.4.6 Difficulty Handling Changing Resolutions

The Two.js library which was used for drawing the Connections between the Dots and Shadow Dots operates only updates the location of each Connection's endpoints when an update event occurs, which is 60 times a second. This also only occurs if the Two.js instances are set up to do so. Since the Connections for the Shadow Dots and the Connections for the Dots exist on different layers there are two instances of the Two.js library running at once, and only the instance handling the Connections between the Dots is set to update the locations of each Connection's endpoints. This lead to two problems. The first is that if the Dots are moved around the screen fast enough the Connections which are rendered just below the Dots can be seen. The second issue this causes is that because the Connections between Shadow Dots weren't expected to move at all their endpoint locations are never updated after they are rendered. This means that any time the application viewport changes size, which can occur when accidentally zooming during gameplay or switching to fullscreen mode in the browser, the Connections between the Shadow Dots may appear out of place, confusing players.

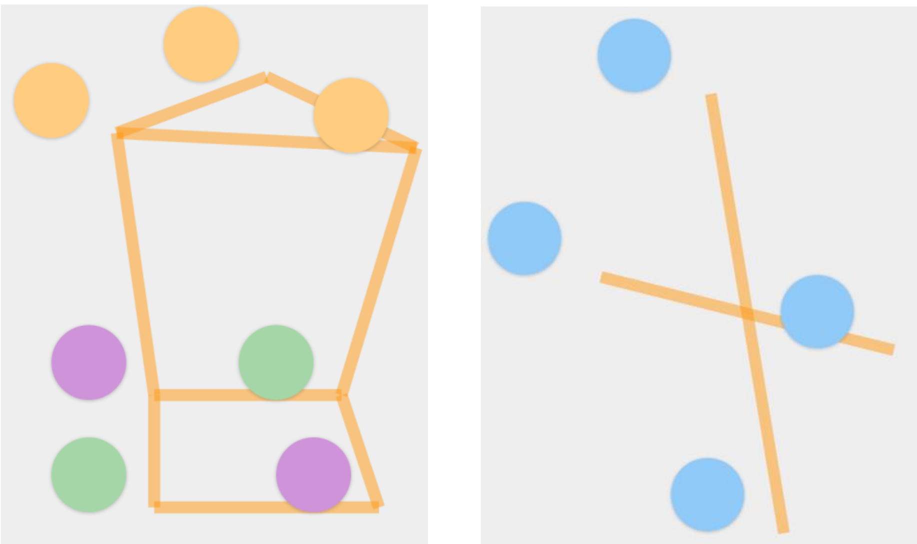


Figure 6.6 A bug in the Connection drawing code for the Shadow Dots layer

7 Evaluation

7.1 User Study

Participants will enter a room containing the multitiouch table, guided by an instructor. The instructor will inform the players of the structure of the activity as well as telling the participants the goal of the game (see Appendix D). While completing the task the participants will be video and voice recorded. These recordings were later transcribed into communication sequence diagrams in the same manner as Liu and Kao (2007).

Figure 7.1 shows the physical layout of the activity. The two players will be positioned across from one another, with the application running on a multitouch table set up between them. Any recording equipment should be able to capture the position of each player as well as any interactions they have with the table or other participant. The two participants will attempt to complete the puzzles, under the instruction that there is no time limit and no failure state for the activity. The participants will not be informed that the goal of this study is to measure collaboration, or that there is significance in the participants talking to each other or pointing at the screen.

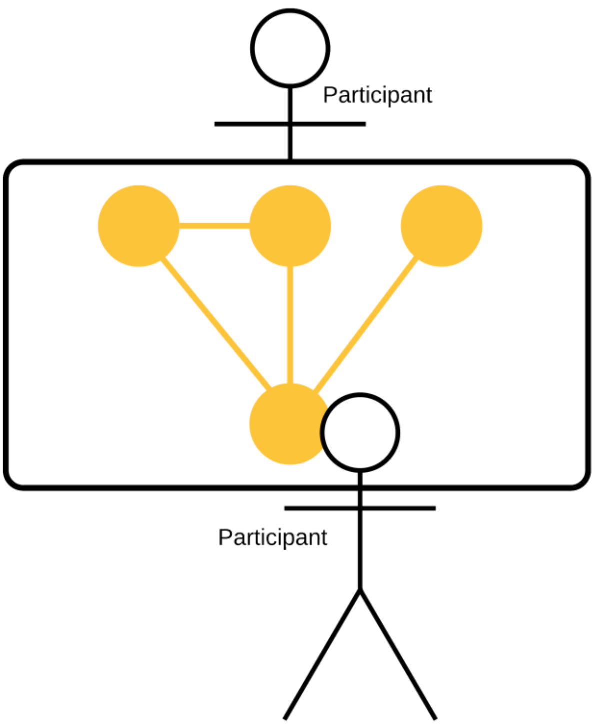


Figure 7.1 Physical layout of activity during user tests

After completing the activity the participants will each be provided with a four point Likert-like questionnaire which will ask the participants questions about their perception of the activity and their partner, whether they felt like they were collaborating and how effective they felt that collaboration was.

7.1.1.1 Balanced Double-Crossover Study

The participants will be taking part in a balanced double-crossover study in order to eliminate order dependence. One form of the activity will require both players to have their hands operating the multitouch table at the same time in order to complete the activity, while the other will allow the activities to be completed by an individual. This approach was chosen due to the low number of participants that were able to be organised, with an aim of getting somewhat representative results without requiring as many participants as would be required for a traditional comparative study.

All participant groups will complete an initial training phase designed to teach them some of the basic gameplay concepts that will be required to complete the activity. This training phase is primarily included in order to ensure a minimum level of competency with the application among the participants. Following this the participants will complete two distinct modes of the application.

Table 7.1 Game mode order for each group

Group	**Mode 1**	**Mode 2**
Group 1	Forced collaboration	No collaboration
Group 2	Forced collaboration	No collaboration
Group 3	No collaboration	Forced collaboration
Group 4	No collaboration	Forced collaboration

The levels were designed and split so that both the ‘forced collaboration’ levels and the ‘no collaboration’ levels were of approximately equal difficulty (see Appendix A). The difficulty of the levels was determined through a combination of the number and variety of Dots, the number of Connections and how many solutions there were to the puzzle. These rough estimates were confirmed through system testing (see section 6.3 System Testing).

7.1.1.2 Likert-like Questionnaire

A four point Likert-like questionnaire (see Appendix E) will be used to measure the participant’s perception of the activity in a number of categories. The questions have a strong focus on the participant’s thoughts of their partner and the effectiveness and necessity of that partner. A four point scale was chosen in order to force the participants to make a decision about the statement presented to them. The questionnaire also employed item reversals, requiring participants to evaluate the same statement twice, once framed positively and once framed negatively. This was done in order to minimise the effects acquiescence bias, a person’s consistent tendency to record agreement rather than disagreement’ (Smith & Fischer, 2008).

7.1.2 Participants

For this study a small pool of participants was selected. These participants were fellow members of the Software Engineering Final Year Project group. Due to the limited diversity of the pool of participants it’s possible that the results obtained may not be representative of the results of a larger study with a broader participant base.

Table 7.2 Pros of using members of the Final Year Project group for the user tests

Pro	Consequence
Easily accessible	The members of the Final Year Project have a dedicated group meeting day, where they all get together. This made it very easy to schedule and organise the user study
Familiar with multiouch applications and technology	Due to the group Final Year Project being a multitouch application all participants are guaranteed to have a strong minimum understanding of multitouch user interface paradigms. This eliminates the need for any training or introduction to multitouch

Table 7.3 Cons of using members of the Final Year Project group for the user tests

Con	Consequence
Participants have worked together extensively over the course of the year as part of the Final Year Project group	As they have worked together it's possible that certain participants or participant pairs are quite comfortable collaborating with one another. It's possible that because of this they will be more likely to collaborate when they are not forced to

8 Results

8.1 Collaboration Sequence Diagrams

Participants were audio and video record during the user study so that it could be determined when they were collaborating. Following the completion of the user study the resulting footage was analysed to determine at what points throughout the activity the participants could be said to be collaborating. For this study the participants are said to be collaborating when any of the following criteria are met;

- Both participants are interacting with the application at the same time
- The participants are talking to each other to resolve ambiguities or develop a strategy for completing the level (Frohlich, 1993; Thomson et al., 2009)

Participants were said not to be collaborating when any of the following criteria are met;

- One participant is trying to interact with the application but their partner is preventing them from doing so
- One participant has begun trying to complete the current level without waiting for their partner to join
- One participant has taken their hands away from the table and leant back from the activity. This participant is said to have disengaged from the activity.

In instances where both players disengaged from the activity, such as leaning back after the completion of a difficult level, then their previous collaboration setting was maintained until one of the above criteria are met. That is, participants that are collaborating and simultaneously disengage from the activity are said to still be collaborating, and participants who are not collaborating immediately before being simultaneously disengaged from the activity are said to still not be collaborating.

Once the footage had been analysed all of the collaboration events were placed on a timeline with one-second resolution. For each group this produced a table similar to Table 8.1 below. Data was also recorded and added to the timeline for the training levels and time spent fulfilling the first survey.

Table 8.1 Example collaboration timeline

Second	Collaborators	Event
0	0	Beginning of training levels, participant X begins activity
1	0	
2	0	
3	2	Participant Y engages the activity
4	2	
5	2	
6	2	
7	0	Participant Y disengages from the activity
8	0	
9	0	
10	2	Participant Y suggests swapping two Dots to Participant X
11	2	
12	2	End of training levels

Subsequently the timeline data pertaining to time spent completing the forced collaboration mode and non-collaborative mode were extracted and placed on to a relative timeline, where events from each mode began at a new relative time of 0. This allows the two timelines to be overlayed in a collaboration sequence diagram. The below graphs (Figure 8.1 to Figure 8.4) are similar to those produced by Liu & Kao (2007), and display the number of participants that are said to be collaborating over time.

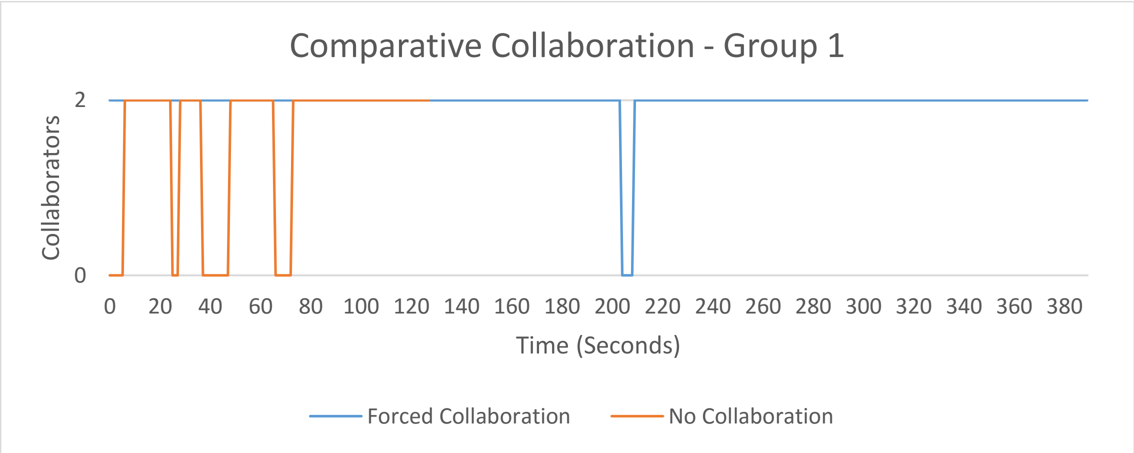


Figure 8.1 Composite collaboration sequence diagram for group 1

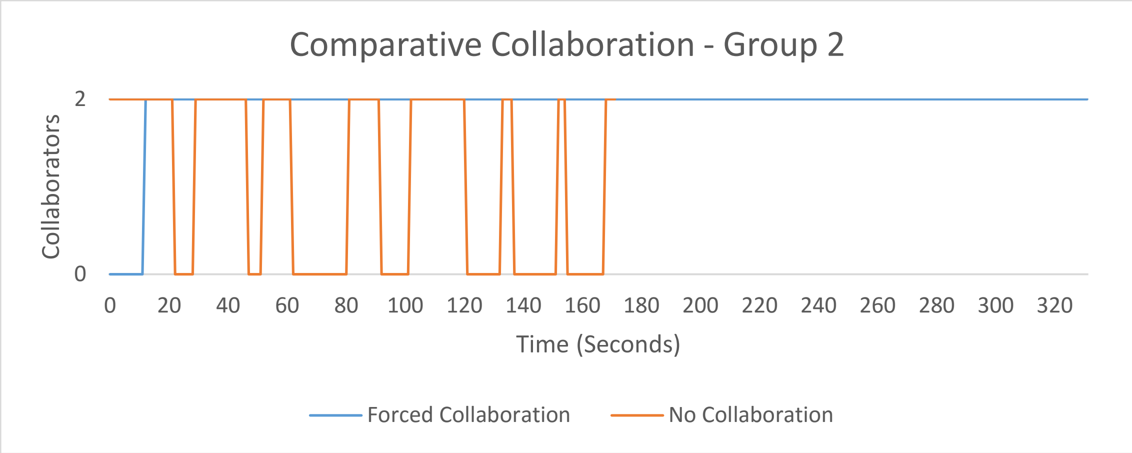


Figure 8.2 Composite collaboration sequence diagram for group 2

Most groups required more time to complete the levels in the forced collaboration mode, despite the levels being of similar difficulty, or in some cases, even isomorphic to levels in the non-collaboration mode. This is likely due to the extra time participants had to spend communicating with their partner in order to complete the level, as they were unable to freely move their hands without the Dots they were holding returning to the centre of the play area.

These collaboration sequence graphs also show that participants switched between collaborating and not collaborating a lot more often when they weren't forced to collaborate. Since participants were no longer required to keep their hands on the multitouch table in order to complete the levels it was easier for the dominant participant to simply complete the level themselves rather than explain it to their partner. This occasionally lead to instances of uncertainty in the non-dominant participant where they would hold their hand a short distance away from the screen, but before they were able to interact with a Dot the dominant partner would move their hand to that position and move the Dot themselves.

This behaviour was more prominent in groups where one participant was much more dominant than their partner, specifically Group 2 and Group 4. These groups also initiated verbal communication with each other much less frequently than the other groups. This may be reflected in Figure 8.7, as these two groups also have the largest gaps between their percentages of time spent collaborating between the forced collaboration mode and non-collaboration mode.

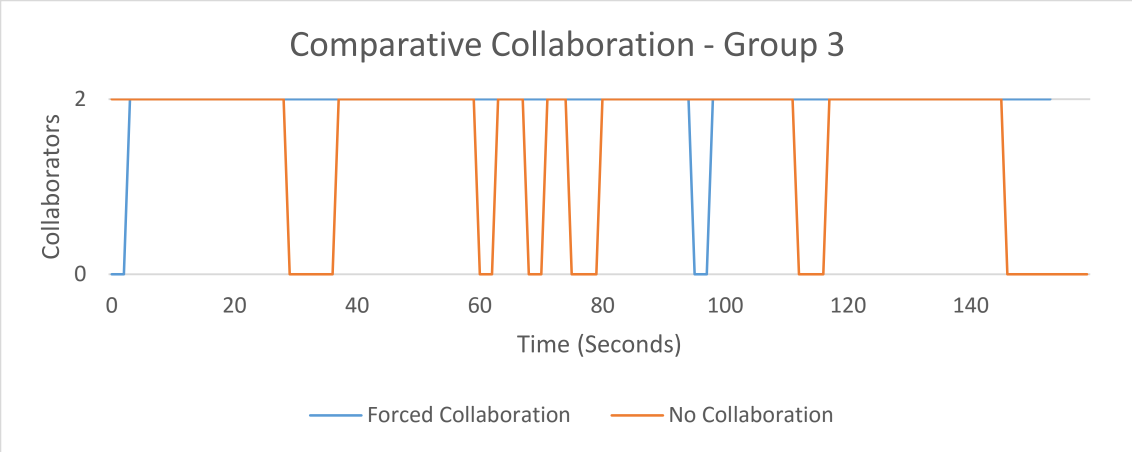


Figure 8.3 Composite collaboration sequence diagram for group 3

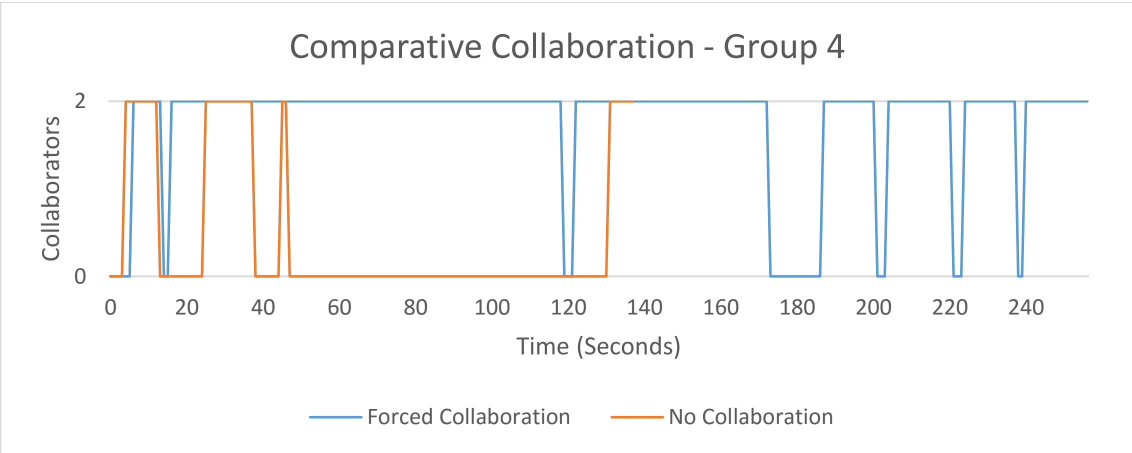


Figure 8.4 Composite collaboration sequence diagram for group 4

Figure 8.5 and Figure 8.6 are composed of the overlayed timeline data of all groups, allowing comparison of the timelines for each group based on what mode the activity was in. The perceived busyness of Figure 8.6 is indicative of observations made during the user study that when participants are not forced to collaborate a more dominant player will begin completing an activity on their own. This presents itself in the collected data as a large number of events occurring where one participant would disengage from the activity, be subsequently compelled to reengage and collaborate shortly afterwards, followed by disengaging from the activity again (see section 8.3.4 Repeated Disengagement for more).

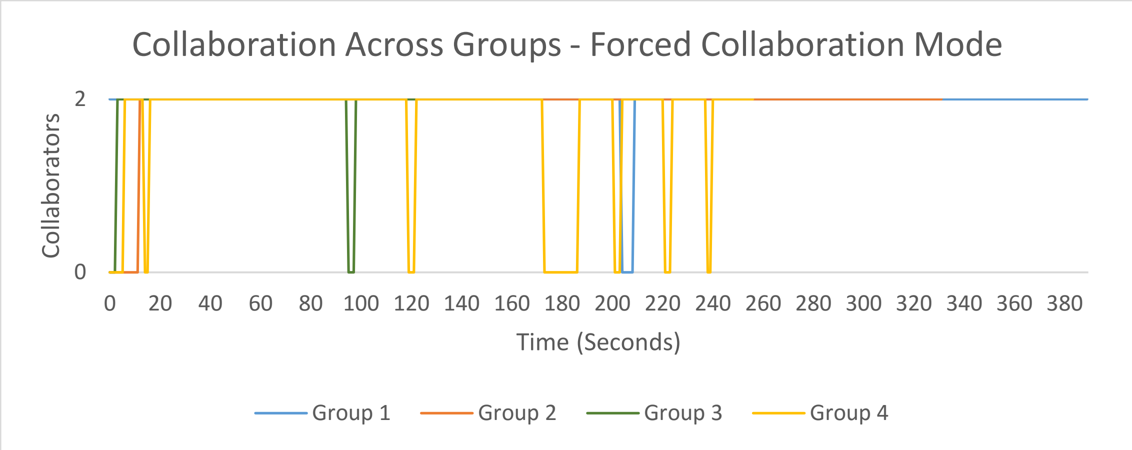


Figure 8.5 Collaboration sequence diagram for all groups in forced collaboration mode

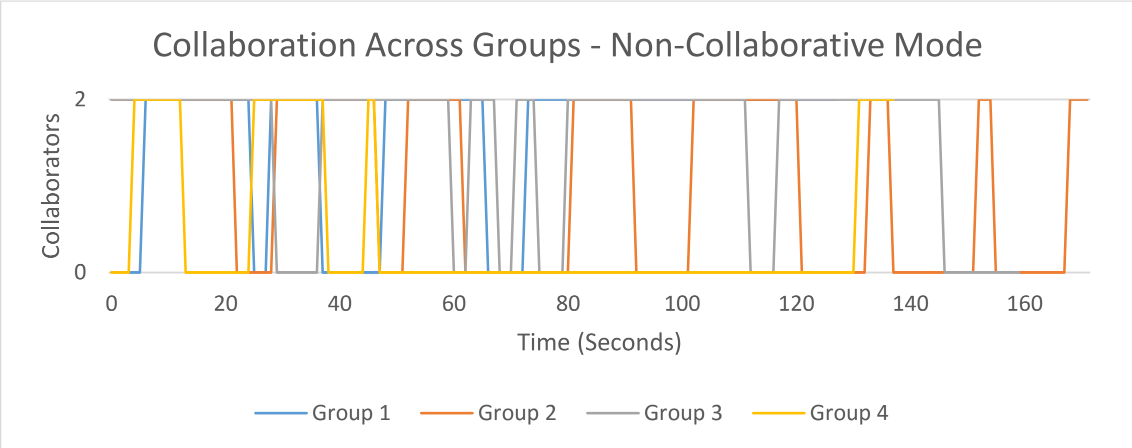


Figure 8.6 Collaboration sequence diagram for all groups in non-collaborative mode

Figure 8.7 shows the amount of time that participants were considered to be collaborating based on what mode of the application they were in at the time. Groups consistently spent significantly more time collaborating as a percentage of the time spent doing the activity, with the lowest time spent collaborating for the forced collaboration mode (88%) being 8% more than the greatest level of collaboration in the non-collaboration mode (80%).

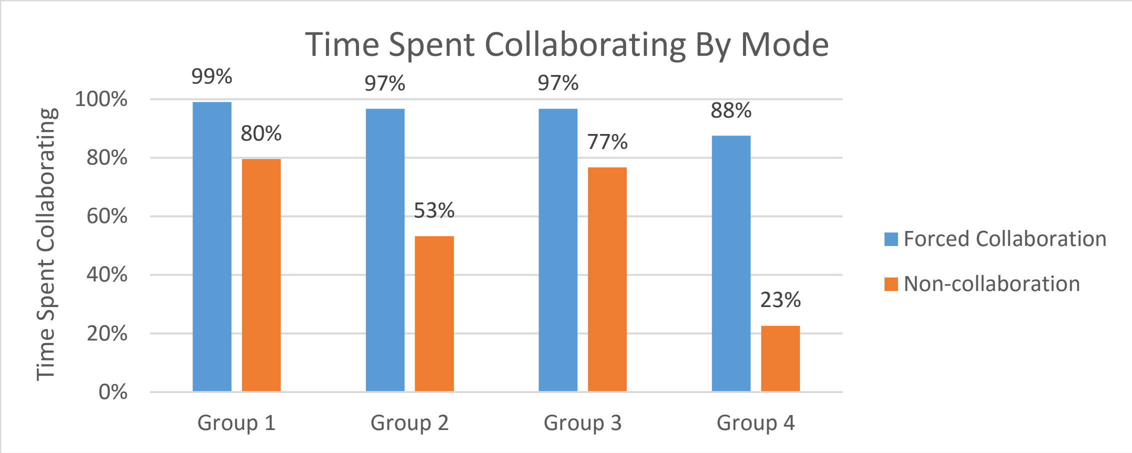


Figure 8.7 Bar graph of the percentage of time spent collaborating, split by mode

8.2 Likert-like Questionnaire

The Likert-like questionnaire was designed to ask respondents the same question twice, once in a positive manner and once in a negative manner (see section 7.1.1.2 Likert-like Questionnaire). For the purposes of analysis each question was given an ID. Positively framed questions have an ID of the form PQ#, where # is the number of that question set, read as "positive question #". Similarly, the negatively framed question has an ID of the form NQ#, read as "negative question #". Each positive and negatively framed version of a question have the same number at the end of their ID. For example, PQ4 and NQ4 are the same question framed positively in the case of PQ4, and negatively in the case of NQ4.

Table 8.2 shows the list of questions the participants were asked following the forced collaboration and non-collaborative modes of the game during the user study, ordered so that each question set pair is adjacent.

Table 8.2 Likert-like questions that participants were asked to respond to after each game mode

ID	QUESTION
PQ1	I achieved better results working with my partner
NQ1	I would of achieved better results by not working with a partner
PQ2	I am satisfied with my participation in the activity
NQ2	I wasn't satisfied with my level of participation in the activity
PQ3	I felt encouraged to work with my partner
NQ3	I didn't feel encouraged to work with my partner
PQ4	I played a more dominant role towards completing the activity
NQ4	My partner played a more dominant role than me towards completing the activity
PQ5	My partner and I communicated effectively
NQ6	My partner and I had trouble communicating effectively
PQ6	My partner and I worked well together
NQ6	My partner and I didn't work well together
PQ7	My partner and I had a shared goal while completing the activity
NQ7	My partner and I seemed to have different goals while completing the activity
PQ8	My partner was vital for me to complete the activity effectively
NQ8	My partner was holding me back from completing the levels effectively

8.2.1 Aggregated Responses

The two tables in Appendix I show the distribution of responses for each of the above questions, aggregated for each game mode. Because these tables show us the distribution of responses for both the negatively framed responses and the positively framed responses the negatively framed responses should be inversely equal to the positively framed responses, that is, the same number of respondents that answered a positively framed question with 'Strongly Agree' should answer the negatively framed question with 'Strongly Disagree'. This leads to a reduced list of questions, containing only the positively framed incarnation of each pair of opposing questions.

Table 8.3 Bias in Responses Responses For Forced Collaboration Mode

Question Set	Strongly Negative	Negative	Positive	Strongly Positive
Q1	0.00%	0.00%	-25.00%	25.00%
Q2	0.00%	-12.50%	0.00%	12.50%
Q3	-12.50%	0.00%	37.50%	-25.00%
Q4	12.50%	25.00%	25.00%	-62.50%
Q5	0.00%	0.00%	0.00%	0.00%
Q6	0.00%	0.00%	12.50%	-12.50%
Q7	0.00%	0.00%	0.00%	0.00%
Q8	0.00%	0.00%	-25.00%	25.00%

Table 8.4 Bias in Responses For Non-collaboration Mode

Question Set	Strongly Negative	Negative	Positive	Strongly Positive
Q1	0.00%	-12.50%	-12.50%	25.00%
Q2	0.00%	-12.50%	12.50%	0.00%
Q3	-25.00%	0.00%	37.50%	-12.50%
Q4	25.00%	-12.50%	-25.00%	12.50%
Q5	0.00%	12.50%	-12.50%	0.00%
Q6	0.00%	0.00%	0.00%	0.00%
Q7	-12.50%	12.50%	-25.00%	25.00%
Q8	25.00%	25.00%	-50.00%	0.00%

Table 8.3 and Table 8.4 show that there was a lot of bias displayed in the responses received, indicating that some of the questions may have been difficult to understand, particularly question sets Q3, Q4 and Q8, which have especially large error rates. However, given the low number of participants, the error rates are slightly inflated compared to what would normally occur in a broader study.

Table 8.5 Distribution of Responses for Forced Collaboration Mode

Question Set	Strongly Negative	Negative	Positive	Strongly Positive
Q1	0.00%	0.00%	25.00%	75.00%
Q2	0.00%	6.25%	25.00%	68.75%
Q3	6.25%	0.00%	31.25%	62.50%
Q4	18.75%	25.00%	25.00%	31.25%
Q5	0.00%	0.00%	50.00%	50.00%
Q6	0.00%	0.00%	31.25%	68.75%
Q7	0.00%	0.00%	37.50%	62.50%
Q8	0.00%	0.00%	12.50%	87.50%

Table 8.6 Distribution of Responses for Non-collaborative Mode

Question Set	Strongly Negative	Negative	Positive	Strongly Positive
Q1	0.00%	18.75%	56.25%	25.00%
Q2	0.00%	6.25%	31.25%	62.50%
Q3	12.50%	25.00%	43.75%	18.75%
Q4	25.00%	31.25%	25.00%	18.75%
Q5	0.00%	6.25%	68.75%	25.00%
Q6	0.00%	12.50%	62.50%	25.00%
Q7	6.25%	6.25%	62.50%	25.00%
Q8	25.00%	12.50%	37.50%	25.00%

8.2.2 Participant Perception of Question Sets

By taking the distributions from Table 8.5 and Table 8.6, and applying the corresponding weights from Table 8.7 to each response category and summing the result the general response trend can be graphed in order to more easily understand the perceptions of the participants after completing each game mode (see Appendix J). Figure 8.8 shows that the participants agreed with the positively framed statements more, and disagreed with the negatively framed statements more when they were participating in the forced collaboration mode.

Table 8.7 Likert-like Questionnaire Response Category Weights

Response Category	Strongly Negative	Negative	Positive	Strongly Positive
Weight	-2	-1	1	2

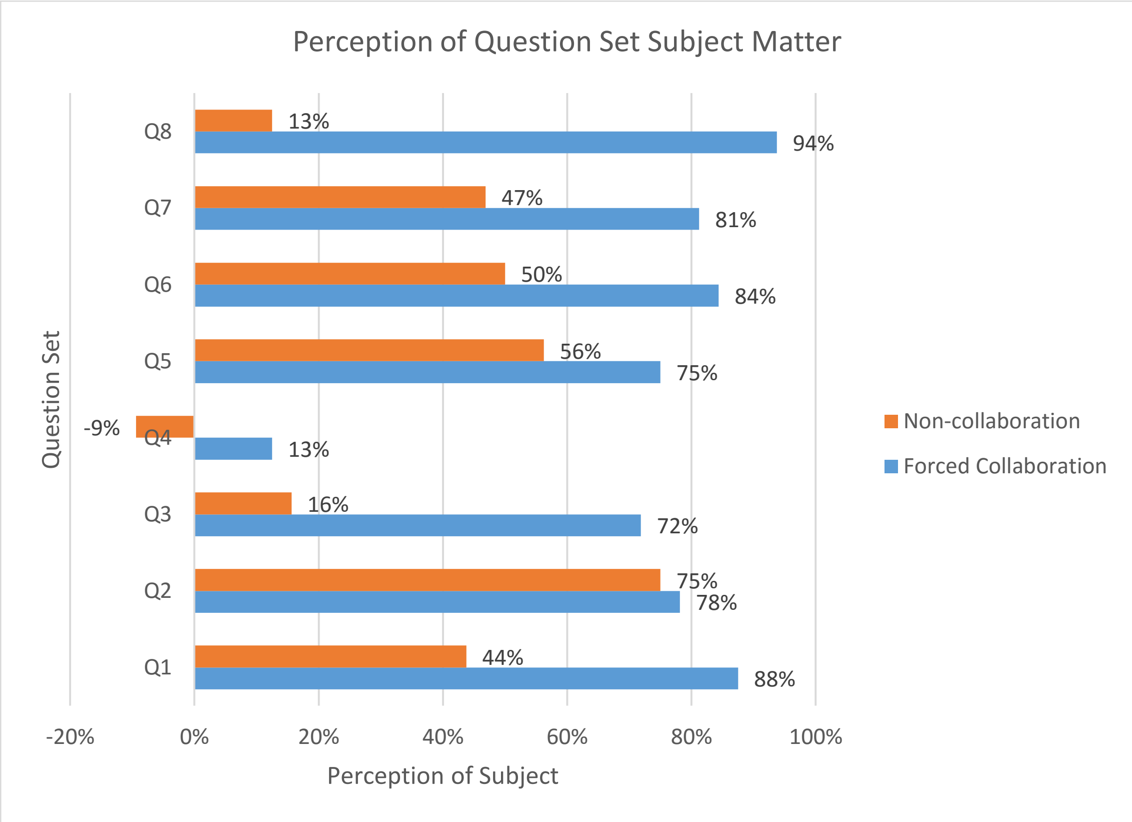


Figure 8.8 Participant Perception of Question Set Subject Matter

8.3 Observations

As well as data collected towards answering the problem statements of this study a number of observations about how the participants interacted with the application were made. These observations help to better describe the types of interaction that participants executed that were not being directly measured.

8.3.1 Strategy Development

All groups developed and discussed strategies for solving the puzzles as they progressed through the activity. The most common strategy used was to determine any unique Dots that could be used as something of a keystone piece, which once placed would allow the proceeding Dots to be placed easily.

Another common strategy was an incremental change strategy, where participants would place a large number of the Dots on Shadow Dots of the same colour, and then proceed to coordinate with each other in order to swap pairs of Dots in order to get closer to a solution.

8.3.2 Play Area Ownership

The physical layout of the user study lead to groups splitting the play area in half, and each participant becoming responsible for managing the Dots on their half of the play area. Language such as “That Dot is yours” was used repeatedly, by multiple groups, often without any preceding conversation to establish this ownership.

8.3.3 Removal of Ambiguity

During the forced collaboration mode players would often need to communicate with their partner in order to execute a potential solution the participant had found. However, as the forced collaboration mode often required both participants to have their hands on the table at the same time in order to fully understand the puzzle they had trouble simply vocalising which Dot they were referring to when multiple Dots of the same colour existed. Participants would proceed to contort their hand in order to point with an unoccupied finger. Direct actions such as this minimise the amount of effort required in order to achieve a desired result, meaning that the collaborative effort is actually the most optimal (Frohlich, 1993).

When a typical instance of this behaviour would occurred participants would employ both vocal cues with direct physical cues. One example of this is participant 2 telling participant 1 ‘That yellow dot you've got there has to be there", while indicating with an unoccupied ring finger to remove the ambiguity of which yellow Dot they meant, as well as what Shadow Dot it should occupy.

8.3.4 Repeated Disengagement

When participants were completing the activity during the non-collaborative mode there was a tendency for there to be a dominant participant. This participant would often begin completing the level on their own, usually without consulting their partner for help. The sometimes led to the other participant completely disengaging from the activity. The disengaged participant would often be encouraged by the dominant participant to take part in the activity. In one instance the dominant participant jovially said to the disengaged participant “Heh, Don't make me do everything!”

9 Conclusion

This study aimed to show that forced collaboration among participants was possible, and that forced collaboration would not negatively impact the perceptions that someone being forced to collaborate has of the activity, or of their partner. Previous studies have shown that collaborative learning produces better results through greater levels of engagement and knowledge retention (Prince, 2004). By building an application that is designed to be run on a multitouch table the inherently social nature of shared workspaces (Dillenbourg & Evans, 2011) can be leveraged in order to provide users with as few roadblocks to enabling collaboration as possible.

As a direct result of this study a multitouch javascript game was developed, which would force participants to collaborate by requiring both participants interact with the game in order to proceed. In order to measure the effects of this application a number of user studies were run, requiring several groups of users to complete the game, which operated in both a forced collaboration mode and a non-collaborative mode. Following the completion of these modes the participants were given a survey to complete that was used to gauge their perception of both the activity and their partner.

The results showed that not only could the participants be forced to collaborate, that by doing so the participants felt as though they were working more effectively together, and also communicated better. The results also showed that when forced to collaborate together that as a team the participants had a greater sense that they shared a set of goals with their partner.

Dillenbourg & Evans (2011) stated that contextualisation and pedagogical goal setting would be required for a new technology to provide a better education experience. If developing a forced collaboration application is better able to rally learners around a shared goal then there are possible applications for forced collaboration in educational contexts, most likely in the form of educational games.

10 References

Capponi, M. F., Nussbaum, M., Marshall, G., & Ester, M. (2010). Pattern Discovery for the Design of Face-to-Face Computer-Supported Collaborative Learning Activities, *13*, 40–52.

Chen, C. (2008). The effectiveness of computer supported collaborative learning on helping tasks in a mathematics course. *Dissertation Abstracts International. A, The Humanities and Social Sciences*. Retrieved from http://dl.acm.org/citation.cfm?id=1559099

Dillenbourg, P., & Evans, M. (2011). Interactive tabletops in education. *International Journal of Computer-Supported Collaborative Learning*, *6*(4), 491–514. http://doi.org/10.1007/s11412-011-9127-7

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. Retrieved from http://www.pmp-projects.org/Agile-Manifesto.pdf

Frohlich, D. M. (1993). The history and future of direct manipulation. *Behaviour & Information Technology*, *12*(6), 315–329. http://doi.org/10.1080/01449299308924396

Gökmen, S. (2009). *The Effects of Cooperative Learning on Learning Outcomes in an In-Service Training Course*. *Educational Sciences: Theory and Practice*. Middle East Technical University. Retrieved from https://etd.lib.metu.edu.tr/upload/12610509/index.pdf

Higgins, S. E., Mercier, E., Burd, E., & Hatch, A. (2011). Multi-touch tables and the relationship with collaborative classroom pedagogies: A synthetic review. *International Journal of Computer-Supported Collaborative Learning*, *6*(4), 515–538. http://doi.org/10.1007/s11412-011-9131-y

Hsiao, H., Chang, C., Lin, C., Chang, C., & Chen, J. (2014). The influence of collaborative learning games within different devices on student ' s learning performance and behaviours, *30*(6), 652–669.

Hudson, J. N. (2004). Computer-aided learning in the real world of medical education: Does the quality of interaction with the computer affect student learning? *Medical Education*, *38*(8), 887–895. http://doi.org/10.1111/j.1365-2929.2004.01892.x

Johnson, D. W., & Johnson, R. T. (2009). An Educational Psychology Success Story: Social Interdependence Theory and Cooperative Learning. *Educational Researcher*, *38*(5), 365–379. http://doi.org/10.3102/0013189X09339057

Liu, C. C., & Kao, L. C. (2007). Do handheld devices facilitate face-to-face collaboration? Handheld devices with large shared display groupware to facilitate group interactions. *Journal of Computer Assisted Learning*, *23*(4), 285–299. http://doi.org/10.1111/j.1365-2729.2007.00234.x

Prince, M. (2004). Does Active Learning Work ? A Review of the Research. *Journal of Engineering Education*, *93*(July), 223–231. http://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Rigden, C. (1999). “ The Eye of the Beholder ”— Designing for Colour-Blind, *17*.

Scott, S. D., Mandryk, R. L., & Inkpen, K. M. (2002). Understanding children’s interactions in synchronous shared environments. *Proceedings of the Conference on Computer Support for Collaborative Learning Foundations for a CSCL Community - CSCL ’02*, 333. http://doi.org/10.3115/1658616.1658664

Smith, P. B., & Fischer, R. (2008). *Multilevel Analysis of Individuals and Cultures*. (F. J. R. van de Vijver, D. A. van Hemert, & Y. H. Poortinga, Eds.). Lawrence Erlbaum Associates.

Thomson, A. M., Perry, J. L., & Miller, T. K. (2009). Conceptualizing and measuring collaboration. *Journal of Public Administration Research and Theory*, *19*(1), 23–56. http://doi.org/10.1093/jopart/mum036

Appendices

Appendix A Level Difficulties

Level Name	Difficulty (1 - 5)	Mode
First Steps	1	Training
Learn to share	1	Training
Line 'em up	2	Training
The Valley	1	Training
Boxxy	2	Training
A cross the universe	2	Forced Collaboration
Pom pom	2	Non Collaborative
Learn you a dot	3	Non Collaborative
Star	3	Forced Collaboration
Scaffold	3	Forced Collaboration
Dial M for Multitouch	3	Non Collaborative
Spliterator	4	Forced Collaboration
Prismatic	4	Forced Collaboration
Pyramid	4	Non Collaborative
Helix	4	Forced Collaboration
Yarn	4	Non Collaborative
Neigh neigh	4	Non Collaborative
Cat's in the cradle	5	Forced Collaboration
Orion	5	Non Collaborative

Appendix B Level Orders

Number in Set	Forced Collaboration Mode	Non-collaborative Mode
1	A cross the universe	Pom pom
2	Star	Learn you a dot
3	Scaffold	Dial M for Multitouch
4	Spliterator	Yarn
5	Prismatic	Pyramid
6	Helix	Neigh Neigh
7	Cat's in the cradle	Orion

```
{
    levelName: 'Yarn',
    springBack: false,
    dots: [
        {id: 1, colour: "orange"},
        {id: 2, colour: "orange"},
        {id: 3, colour: "orange"},
        {id: 4, colour: "blue"},
        {id: 5, colour: "blue"},
        {id: 6, colour: "blue"}
    ],
    shadows: [
        {id: 1, x: -250, y: -200, colour: "orange"},
        {id: 2, x:    0, y: -200, colour: "orange"},
        {id: 3, x:  250, y: -200, colour: "orange"},
        {id: 4, x: -250, y:  200, colour: "blue"},
        {id: 5, x:    0, y:  200, colour: "blue"},
        {id: 6, x:  250, y:  200, colour: "blue"}
    ],
    connections: [
        {from: 1, to: 4},
        {from: 1, to: 5},
        {from: 2, to: 4},
        {from: 2, to: 5},
        {from: 2, to: 6},
        {from: 3, to: 5},
        {from: 3, to: 6}
    ],
    accepts: [
        [{dot:1, shadow:1},{dot:2, shadow:2},{dot:3, shadow:3},{dot:4,
shadow:4},{dot:5, shadow:5},{dot:6, shadow:6}],
        [{dot:1, shadow:3},{dot:2, shadow:2},{dot:3, shadow:1},{dot:4,
shadow:6},{dot:5, shadow:5},{dot:6, shadow:4}]
    ]
}
```

```
{
    levelName: "Spliterator",
    springBack: true,
    dots: [
        {id: 1, colour: "purple"},
        {id: 2, colour: "purple"},
        {id: 3, colour: "cyan"},
        {id: 4, colour: "cyan"},
        {id: 5, colour: "red"},
        {id: 6, colour: "orange"}
    ],
    shadows: [
        {id: 1, x: -100, y:  200, colour: "purple"},
        {id: 2, x:  100, y: -100, colour: "purple"},
        {id: 3, x: -300, y: -100, colour: "cyan"},
        {id: 4, x:  300, y: -100, colour: "cyan"},
        {id: 5, x:  100, y:  200, colour: "red"},
        {id: 6, x: -100, y: -100, colour: "orange"}
    ],
    connections: [
        {from: 1, to: 6},
        {from: 6, to: 3},
        {from: 3, to: 1},
        {from: 2, to: 4},
        {from: 4, to: 5},
        {from: 5, to: 2}
    ],
    accepts: 'idMatch'
}
```

Appendix D Evaluation Script

You will be completing a number of puzzles on the multitouch screen between you. The aim of the puzzle is to align all the large manipulable Dots on the screen with a receiving Dot. There is no fail state for this game. The game will have three distinct phases;

- Phase 1 will be a few levels to help introduce the concepts and mechanics of the game.
- During Phase 2 and Phase 3 you will be completing puzzles in one of 2 modes. These modes differ only in that in one mode the Dots will return to the centre of the play area once they are no longer being touched, while in the other mode they will remain in place.

Following both Phase 2 and Phase 3 you will fill out a short questionnaire that will ask you questions relating to the Phase you just completed. These questionnaires do not cover the training Phase. In between each Phase a message will appear letting you know that you have just completed the previous phase and another is about to begin. During the activity I will not be able to respond to any questions about the activity, nor give any hints

Appendix E Likert-like Questionnaire

Post Activity Set Questionnaire

Participant Designation:

Set: 1 ◯ 2 ◯

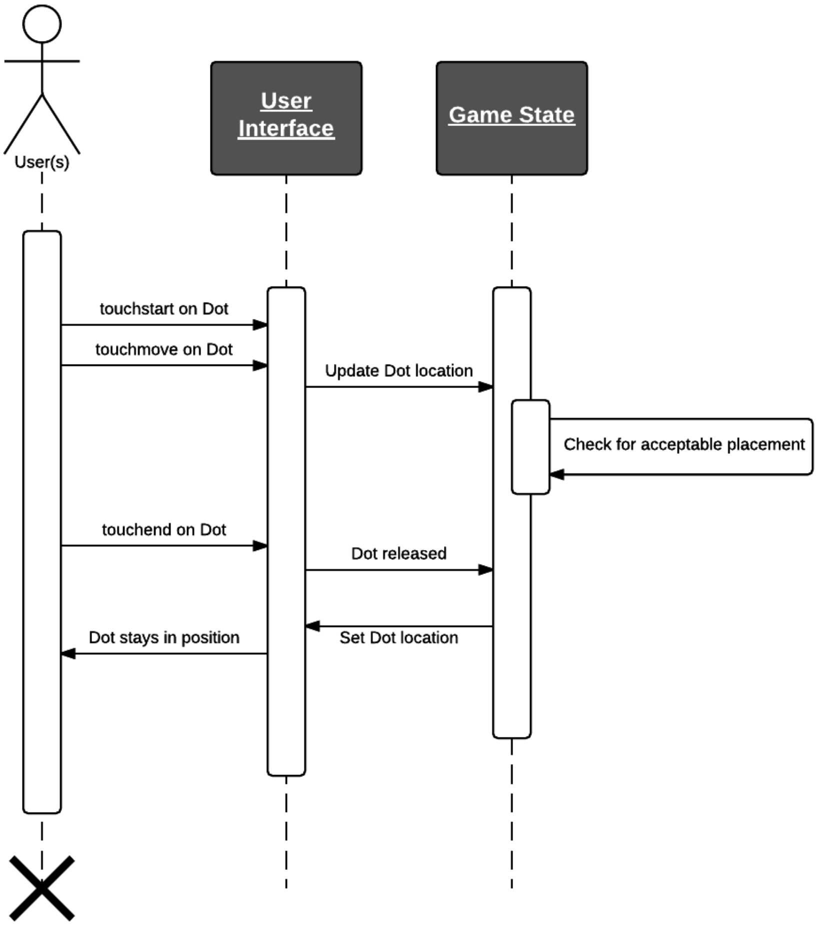
For each of the questions below please fill in the circle in the column that best characterises how you feel about the given statement.

Statement	Strongly Disagree	Disagree	Agree	Strongly Agree
I wasn't satisfied with my level of participation in the activity	◯	◯	◯	◯
I played a more dominant role towards completing the activity	◯	◯	◯	◯
My partner was vital for me to complete the activity effectively	◯	◯	◯	◯
I am satisfied with my participation in the activity	◯	◯	◯	◯
I didn't feel encouraged to work with my partner	◯	◯	◯	◯
My partner and I seemed to have different goals while completing the activity	◯	◯	◯	◯
I achieved better results working with my partner	◯	◯	◯	◯
My partner and I had trouble communicating effectively	◯	◯	◯	◯
My partner and I worked well together	◯	◯	◯	◯
My partner and I communicated effectively	◯	◯	◯	◯
I would of achieved better results by not working with a partner	◯	◯	◯	◯
My partner was holding me back from completing the levels effectively	◯	◯	◯	◯
My partner and I had a shared goal while completing the activity	◯	◯	◯	◯
My partner played a more dominant role than me towards completing the activity	◯	◯	◯	◯
I felt encouraged to work with my partner	◯	◯	◯	◯
My partner and I didn't work well together	◯	◯	◯	◯

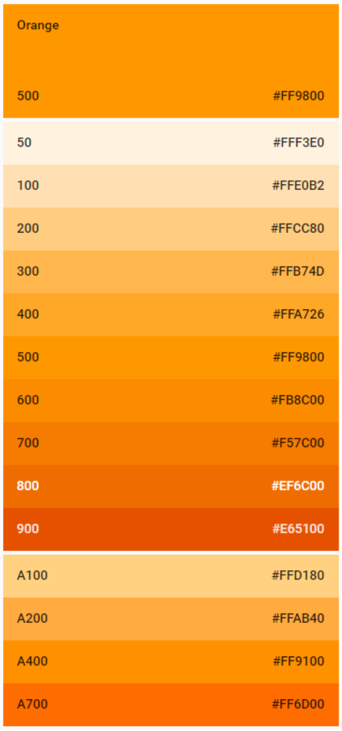
Was there anything in particular you enjoyed about the group of activities you just completed?

Was there anything in particular you experienced difficulties with during the last group of activities?

Appendix F Sequence Diagram of Normal Play with No Forced Collaboration



Appendix G Example Colour Swatch from the Material Design Specification



[14]

[14] Further swatches available from https://www.google.com/design/spec/style/color.html

Appendix H Likert-like Questionnaire Responses

The following tables are the responses from the Likert-like questionnaire that all participants were asked to fill out after completing each mode of the application. Values present are keyed like so; -2 = Strongly Disagree, -1 = Disagree, 1 = Agree, 2 = Strongly Agree.

Group 1	**P1**		**P2**	
Question Id	**Collab**	**No Collab**	**Collab**	**No Collab**
PQ1	1	1	2	1
NQ1	-1	-1	-1	1
PQ2	2	2	1	1
NQ2	-1	-2	-1	-1
PQ3	1	1	1	1
NQ3	-1	1	-2	-1
PQ4	1	2	-1	-1
NQ4	1	1	-1	1
PQ5	1	1	1	1
NQ5	-1	-1	-1	-1
PQ6	1	1	1	1
NQ6	-1	1	-2	-1
PQ7	1	1	1	1
NQ7	-1	-1	-1	-1
PQ8	2	-2	2	1
NQ8	-1	-1	-1	-1

Group 2	P3		P4	
Question Id	Collab	No Collab	Collab	No Collab
PQ1	2	2	2	-1
NQ1	-2	-1	-1	1
PQ2	2	2	1	1
NQ2	-2	-2	1	1
PQ3	2	2	2	-1
NQ3	-2	-2	-2	2
PQ4	-2	-2	1	-1
NQ4	-2	-2	-2	-1
PQ5	2	2	1	-1
NQ5	-2	-2	-1	-1
PQ6	2	2	2	-1
NQ6	-2	-2	-2	-1
PQ7	2	2	2	-1
NQ7	-2	-2	-1	-1
PQ8	2	-1	2	-2
NQ8	-2	-2	-2	2

Group 3	P5		P6	
Question Id	Collab	No Collab	Collab	No Collab
PQ1	2	1	2	2
NQ1	-2	-1	-2	-2
PQ2	2	1	2	2
NQ2	-2	-1	-2	-2
PQ3	1	1	2	1
NQ3	-2	-1	-2	-2
PQ4	1	1	-1	-2
NQ4	-2	-1	-2	1
PQ5	1	1	2	1
NQ5	-1	-1	-2	-1
PQ6	1	1	2	1
NQ6	-1	-1	-2	-1
PQ7	1	1	2	2
NQ7	-2	-1	-2	-1
PQ8	2	-1	2	2
NQ8	-2	-1	-2	-1

Group 4	**P7**		**P8**	
Question Id	**Collab**	**No Collab**	**Collab**	**No Collab**
PQ1	2	1	2	2
NQ1	-2	-1	-2	-1
PQ2	2	2	2	2
NQ2	-2	-2	-2	-2
PQ3	2	-1	1	1
NQ3	2	2	-2	1
PQ4	-2	2	-1	-2
NQ4	-2	-1	2	2
PQ5	2	1	2	2
NQ5	-2	-1	-2	-2
PQ6	2	1	2	2
NQ6	-2	-1	-2	-2
PQ7	2	1	2	2
NQ7	-2	-1	-2	2
PQ8	2	-2	2	2
NQ8	-2	-1	-2	-2

Appendix I Aggregate Likert-like Response Scores

The following tables show the distribution of responses that were received for the Likert-like questionnaire that participants filled out after completing each game mode.

	Forced Collaboration Mode			
Question	**Strongly Disagree**	**Disagree**	**Agree**	**Strongly Agree**
PQ1	0.00%	0.00%	12.50%	87.50%
NQ1	62.50%	37.50%	0.00%	0.00%
PQ2	0.00%	0.00%	25.00%	75.00%
NQ2	62.50%	25.00%	12.50%	0.00%
PQ3	0.00%	0.00%	50.00%	50.00%
NQ3	75.00%	12.50%	0.00%	12.50%
PQ4	25.00%	37.50%	37.50%	0.00%
NQ4	62.50%	12.50%	12.50%	12.50%
PQ5	0.00%	0.00%	50.00%	50.00%
NQ5	50.00%	50.00%	0.00%	0.00%
PQ6	0.00%	0.00%	37.50%	62.50%
NQ6	75.00%	25.00%	0.00%	0.00%
PQ7	0.00%	0.00%	37.50%	62.50%
NQ7	62.50%	37.50%	0.00%	0.00%
PQ8	0.00%	0.00%	0.00%	100.00%
NQ8	75.00%	25.00%	0.00%	0.00%

	Non-collaborative Mode			
Question	**Strongly Disagree**	**Disagree**	**Agree**	**Strongly Agree**
PQ1	0.00%	12.50%	50.00%	37.50%
NQ1	12.50%	62.50%	25.00%	0.00%
PQ2	0.00%	0.00%	37.50%	62.50%
NQ2	62.50%	25.00%	12.50%	0.00%
PQ3	0.00%	25.00%	62.50%	12.50%
NQ3	25.00%	25.00%	25.00%	25.00%
PQ4	37.50%	25.00%	12.50%	25.00%
NQ4	12.50%	37.50%	37.50%	12.50%
PQ5	0.00%	12.50%	62.50%	25.00%
NQ5	25.00%	75.00%	0.00%	0.00%
PQ6	0.00%	12.50%	62.50%	25.00%
NQ6	25.00%	62.50%	12.50%	0.00%
PQ7	0.00%	12.50%	50.00%	37.50%
NQ7	12.50%	75.00%	0.00%	12.50%
PQ8	37.50%	25.00%	12.50%	25.00%
NQ8	25.00%	62.50%	0.00%	12.50%

Appendix J Participant Perception of Question Sets

The below table shows how positive or negative the participant's perception of the subject matter of the question set is, that is, how much the participants agreed with the statements presented to them.

Question Set	Forced Collaboration	Non-collaboration
Q1	88%	44%
Q2	78%	75%
Q3	72%	16%
Q4	13%	-9%
Q5	75%	56%
Q6	84%	50%
Q7	81%	47%
Q8	94%	13%