

From DigiQuilt to DigiTile: Adapting Educational Technology to a Multi-Touch Table

Jochen Rick and Yvonne Rogers
The Open University
Department of Maths, Computing, and Technology
Walton Hall, Milton Keynes, MK7 6AA, UK
{j.rick, y.rogers}@open.ac.uk

Abstract

To realize the potential of multi-touch tables, interaction designers need to create meaningful applications for them in real-world contexts. One convenient shortcut towards that end is adapting a meaningful application from another interface paradigm. In this paper, we detail the process of adapting DigiQuilt, a single-user desktop educational technology, to DigiTile, a collaborative multi-touch application. With this case study, we concretely demonstrate the utility of adapting and how previous research and theory can inform that process. In particular, we show how learning theory (1) motivated the transition from the desktop to the multi-touch table, (2) guided the design process, and (3) informed the evaluation.

1 Introduction

Each new interface requires studies of how the technology can best be appropriated. Early work tends to focus on technical innovations and basic usability. “Demo apps” that demonstrate the features and potential of the new UI are created. Increasingly, more sophisticated applications are developed to solve real world problems in ecologically-valid settings.

Interaction designers are tasked with creating these applications. Do they start from scratch? Or, can they adapt existing applications, developed for other user interfaces, to the new hardware? Here, we make the case for the latter. We detail our own success in adapting DigiQuilt, a single-user desktop application, into DigiTile, a multi-user multi-touch tablet application.¹

¹We chose to change the name for the multi-touch version for two reasons. First, it makes it easier to distinguish between the two in writing. Second, quilting is seen differently in the U.S., where DigiQuilt was conceived, and the U.K., where DigiTile is used.

2 Adapting Applications

Adapting is one of the best and most natural ways we have to understand novel interfaces. A novel interface is a new medium. As with any new medium, it is difficult to predict its affordances. One of the best practices for understanding a new medium is *remediation*—the practice of taking techniques and practices that work in one domain and applying them to the new domain [1]. Many new media are created and explored in this matter. For instance, early sound recording was based on a parallel to written text; Edison envisioned his phonograph as writing sounds [4]. By applying the practices of the better understood medium, a better understanding of the new medium can be achieved. With experience and innovation, new uses particularly suited to the new medium emerge. So, while Edison envisioned a machine that could both read and write sounds, recorded sound technology came into prominence with the application of pre-recorded musical records. Based on these principles, a particularly fruitful way to investigate new user interfaces is to apply ideas from already existing applications to the new interface.

Adapting an application also has several practical advantages. First, the plethora of existing applications provide a good source of inspiration for design. Even if only one percent of all desktop applications would make for viable multi-touch table applications, there would still be plenty of worthy candidates, given the large amount of desktop applications. Much of good design is appropriating previous good ideas [11]. A successful application embodies several working ideas. Adaptation allows the designer to build on these.

Second, adapting can shorten development time by leveraging on the previous design work. A speedy development cycle can be critical to research on novel interfaces. It can be difficult for application-based research to keep up with new hardware innovations; by the time an application has

been built for specific hardware, it is out of date. There is only a limited amount of time when research results can have an important impact. If the novel interface becomes commercially viable, work on the novel interface can move out of the research domain before significant results can be achieved. Adapting an existing application can cut that development time down significantly, because the designer of the new application can appropriate the design work put into the existing application.

Third, the existing application and the adapted application can inform each other. For novel user interfaces, a comparison to a similar application for an established interface can provide researchers a point to compare and contrast. Thus, adapting can also aid in evaluating the new interface.

While adapting has its advantages, there are also challenges that come with adapting to a new interface. Depending on the nature of the application, overcoming these challenges can range from easy to difficult. For the easy case, consider using a tabletop for baggage screening. In baggage screening, the user is trying to identify dangerous objects, such as guns, from an x-ray image. Having multiple users view the image from different angles can help increase accuracy [3]. To adapt this task to a multi-touch table is trivial: The display just needs to be placed horizontally. This works well, because the interaction (clearing the bag or signaling for a hand search) does not change much (if at all). For applications where the graphical user interface does not need to change, middle-ware that maps touch input into desktop commands, such as mouse movements, can speed up the adaption process [28].

Where some applications are easy to adapt, others require serious design and development work. While remediation works to a certain extent, the remediated practice is usually different to the original practice [1]. Changing the interfaces changes the activity. For these adaptations, guiding theory of the application domain and previous research results can inform the process. First, these sources can provide *motivation* for adapting a specific application. Second, they can inform the concrete *design*. Third, they can help develop criteria for *evaluation*. In this paper, we concretely demonstrate how learning theory and previous research informed these three phases of adapting DigiQuilt to a multi-touch table.

3 From DigiQuilt to DigiTile: A Case Study

DigiTile was created for the ShareIT project,² which aims to further the understanding of novel interfaces for supporting collaboration in real-world situations. One path for supporting collaboration with computers is *single display groupware* [26], where multiple people can simultaneously interact with a shared display. This user paradigm

²<http://shareitproject.org>

is being supported by a number of new technologies. One such technology is multi-touch tabletop computers. Tabletop computers have a wide design space, especially when part of a larger ubiquitous computing infrastructure [19].

Collaboration tables [23], tabletop computers that allow multiple people to interact, are still a fairly novel user interface. Hardware innovations are still being published on a regular basis [6, 7, for example]. Yet, a reasonable hardware basis exists for conducting HCI research. For example, MERL's DiamondTouch [2] has been actively researched since 2001. Software toolkits, such as DiamondSpin [24] and reacTIVision [9], help developers overcome some of the more vexing problems of developing software for these platforms. With this hardware and software basis, research on tabletop computing can focus on creating applications. Yet, designing applications from scratch can be a difficult and time-consuming task.

To shortcut this process, we adapted an existing application, DigiQuilt, to study the potential of multi-touch tabletops to support collaboration in an authentic setting. In particular, DigiTile was designed to support learning, a task where collaboration can often be beneficial. Designing computer-based learning environments is an established research field with established criteria [25] for success. While multi-touch table technology is relatively new, there have also been published research on users using multi-touch tabletops for learning [14, 16, for example]. Finally, there is the previous research on DigiQuilt. DigiTile drew on all of these areas.

This case study first introduces DigiQuilt as an interesting educational technology. Next, it motivates adapting DigiQuilt to a multi-touch table. Then, it details the significant design decision in creating DigiTile and the rationale behind those decisions. Finally, the plans for evaluating DigiTile are discussed.

3.1 DigiQuilt

DigiQuilt is a *construction kit* [17] for learning about math and art by designing patchwork quilt blocks [13]. It is based on the the instructional theory of *constructionism*, which holds that people learn particularly well when designing personally-meaningfully public artifacts [15]. In constructionism, learners are motivated to construct the artifact, because it is meaningful to themselves and because it can be shared with others. In DigiQuilt, third and fourth graders assemble colored pieces into a square quilt block (Figure 1).

Clicking a color button in the palette (1), turns the pieces at (2) that color. Clicking on a piece creates a copy of that piece, which can be dragged into the quilt block (3) or into the work area (4). In the work area, pieces can be rotated and assembled into more complex pieces. The

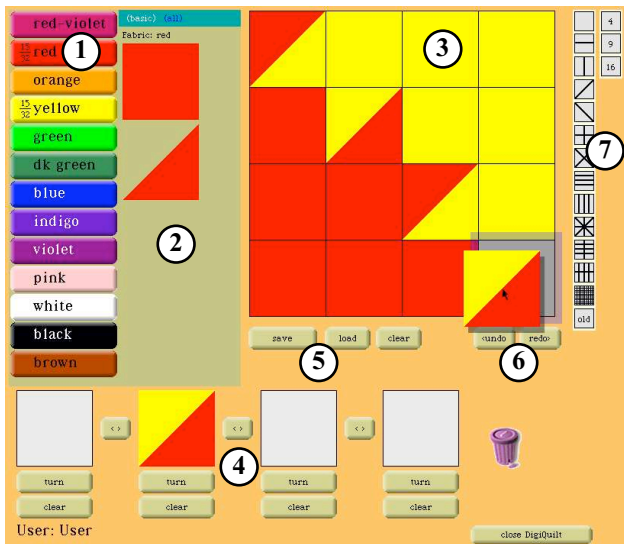


Figure 1. Using DigiQuilt to create a quilt block design (actual 3rd grader's design)

complex pieces can then be copied and transferred into the quilt block. The software keeps a complete history of the quilt block, which learners can navigate (6) to revert unwanted changes. Once finished, the design can be saved (5). Saved designs can be opened and edited. Designs can be printed out to be displayed in the classroom, used as wrapping paper, etc. In user studies, children enjoyed creating designs with DigiQuilt and proudly showed them to their classmates, teacher, and family [12].

In addition to being aesthetically pleasing, the patch work quilts also lend themselves to mathematical analysis. The designs embody fraction concepts and are often symmetric. For instance, the design in Figure 1 is half red and half yellow; it is also diagonally symmetric. To highlight the learning goals, learners are given increasingly difficult challenges to accomplish, such as creating a design that is half red or creating a design that is horizontally symmetric. To help learners reflect on fractions, the fraction of the whole that a certain color covers is displayed next to that color (Figure 1, Point 1). To help learners reflect on symmetry, different grids (Figure 1, Point 7) can be overlaid on the quilt. These tools help learners reflect on the targeted concepts.

3.2 Motivation

The previous section introduced DigiQuilt as a successful and compelling single-user desktop application; however, that by itself is not enough to justify adapting it to a multiple-user tabletop application. Adapting to a new interface requires examining both the *application* and at the

implementation level. Could this application benefit significantly from the move to the new interface? If not, why adapt. How hard is it to implement the transition to the new interface? Even if the application would be compelling, implementation issues might make it impractical.

3.2.1 Application

The primary reason to adapt DigiQuilt to a multi-touch table is *the potential to support collaborative learning*. Collaboration is an established method for enhancing learning. Single display groupware has been shown to better support collocated collaborative learning than taking turns on single-user application [8].

One relevant theory of collaborative learning is Roschelle's [21] theory of *convergent conceptual change*—when two learners work together with a *reflective tool* (a tool that responds to user input to reflect the embedded domain concepts), they tend to converge on an understanding that is better than either would achieve independently. When learners work together on a challenge, they naturally articulate how they would solve the challenge, based on their understanding. If their strategies clash, it triggers them to justify their understanding. The reflective tool allows the learners to demonstrate or test their understanding. Thus, an understanding can be confirmed or rejected. As they work on the challenge together, learners' conceptual understandings do not just converge with each other, but also with the domain concepts embodied in the tool.

DigiQuilt is a reflective tool, giving feedback when changes are made (e.g., the fraction updates when a new piece is placed). So, we expect that two learners working with DigiQuilt to engage in convergent conceptual change. There is also practical evidence that collaboration might be warranted: DigiQuilt users often make comments on others' designs and sometimes want to take over the controls to show something [12].

Roschelle's model of collaboration is particularly compelling, because it shows how collaboration between equals can benefit both partners. On the flip side, if collaborators are not evenly skilled, as is often the case in a classroom setting, a more appropriate model of collaboration may be that of Vygotsky's [29], where a more skilled person helps a less skilled person gain competence. No matter which model of collaboration (even or uneven) is appropriate to a specific pair, we would still expect the collaboration to benefit learning.

3.2.2 Implementation

While implementation is a challenge when actually adapting, it is worth considering beforehand. If there is not a good fit between the source application and the new interface, it may be difficult to successfully adapt. The interface

may prove awkward or the application may lose its compelling nature in the process.

DigiQuilt's existing interface is a good match for a multi-touch table. It is a fairly simple application with a small number of user actions. The primary action is to move pieces to the work area and quilt block. While this is easy enough with a mouse, simply dragging the shapes with a finger on a multi-touch table could be an even better physical mapping. Early prototypes of DigiQuilt were carried out with physical shapes on paper. While this physical approach had significant drawbacks,³ moving pieces by hand was an intuitive interface [12]. Besides dragging pieces, the main interactions in DigiQuilt are pushing buttons, something that translates directly to a multi-touch table. DigiQuilt does make use of the keyboard to name designs. This small amount of typing can be achieved with a virtual on-screen keyboard (see Figure 4).

3.3 Design: To DigiTile

As outlined above, the basic interactions of DigiQuilt are well supported by a multi-touch table, but there are still important design decisions to be made in adapting. In this section, we detail how learning theory and previous research helped us negotiate these design challenges. We organize the subsections around design features well documented in the literature on multi-touch tables—group / table size, orientation, and collaboration.

3.3.1 Group / Table Size

A first problem to consider is *group size* and *table size* [22]. Based on the theory of convergent conceptual change, two is the ideal number of users: Two can challenge and help each other without spending undue amount of time coordinating.

We wanted the learners to collaborate on one design, so the table needed to be small enough to allow intended users (10–12 year-olds) to both reach every part of a central tile. The smaller DiamondTouch table (32 inch diagonal) easily allowed for these young users to reach almost all of the table.

³While a physical interface was intuitive, it failed in other ways. First, it was hard to properly constrain the physical world: Users could easily overlap tangible pieces in ways that did not allow for fractional analysis. Second, physical pieces do not allow for several useful operations easily implemented in software: assembling simple pieces into a complex piece (such as an L shape), creating copies of these complex pieces, swapping colors, reverting to an earlier design point, etc. Third, with a large number of colors and shapes, a physical implementation would require an overwhelming amount of different pieces.

3.3.2 Orientation

Unlike a desktop display, different users need not have the same perspective on a tabletop display. They could sit across from each other, around the corner, or next to each other. This makes it difficult to maintain the desktop orientation, where eyes up equals the top of the screen and eyes down means the bottom of the screen [27]. Developing software without a dominant orientation can be tricky [24].

For DigiTile, different positions would change how learners view the tile. If seated around a corner, horizontal symmetry for one partner would be vertical symmetry for the other. If seated across from each other, the tile would appear upside down. For more abstract designs, this would not be a problem; however, many DigiQuilt designs are based on real-life objects which have a preferred orientation (e.g., a stick figure). While adults tend to favor working across from each other, children often prefer working next to each other [23]. So, it made sense to position the learners next to each other to share the same orientation. This configuration (dyads next to each other) allows DigiTile to maintain a desktop-style orientation (Figure 2).

3.3.3 Software Implementation

DigiTile is implemented in Squeak [5], a multimedia-enabled cross-platform open-source Smalltalk. A small Java application reads DiamondTouch events and broadcasts them to Squeak via OSC (Open Sound Control). Squeak then handles the events. While it was not designed for creating applications for a multi-touch table, it was relatively easy to adapt Squeak for this purpose. For instance, all Squeak interfaces can be rotated.

More pressing for DigiTile, Squeak allows for multiple concurrent mouse pointers, which can be a challenge for some UI toolkits [23]. Mapping touch input to mouse input is not entirely straightforward, since hovering and clicking do not work quite the same as for a mouse. In our mapping, any touch was treated as a mouse down condition; further movements were treated as mouse drag movements. This simple mapping matched well with DigiQuilt's interface, since its main interaction techniques were dragging tile pieces and pressing buttons. Other common mouse functions, such as pressing secondary buttons and scrolling, were not necessary.

DigiQuilt itself was implemented in Squeak. While the source code for DigiQuilt was available, we choose to implement DigiTile from the ground up. Like most software which has been slowly evolved over an extended time by several developers, DigiQuilt's code base had serious problems. The size of the pieces was hard coded. The model and view were tightly coupled. The file format was optimized for size, not flexibility. We took this opportunity to rewrite the application with a cleaner design, being espe-

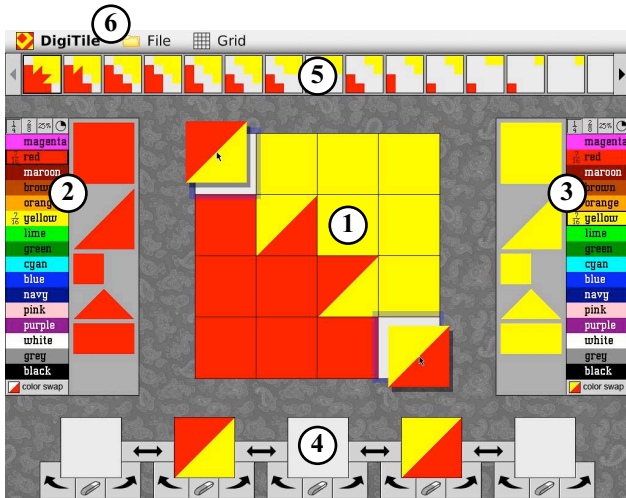


Figure 2. Using DigiTile to create a tile

cially careful to decouple model and view; thus, different views for multi-touch application and the desktop application can share the same underlying model.

3.3.4 Collaboration

As with DigiQuilt, the main focus of DigiTile (Figure 2) is the quilt block (1). Since the learners are positioned next to each other, we placed a palette area on both the left (2) and right (3) side. Practically, each learner has their own palette to use. This positioning allows learners to feel like they have specific *ownership* over part of the interface. To encourage *sharing*, we created a shared work area (4). We intentionally used an odd number of work zones, so that one work zone would be in the middle, implying no left or right ownership. Like in DigiQuilt, the palette buttons are labeled with the name of the color. In classroom use of DigiQuilt, the labelling proved to be useful in allowing children to talk about their designs [12]. Similarly, we believe that these labels can help aide dialogue (e.g., students can easily differentiate between lime and green).

After some user testing, it was obvious that dropping a piece accidentally was fairly common in the multi-touch case. In addition, partners often disapproved of a specific change. For these reasons, navigating the history was more important for DigiTile than DigiQuilt. We developed a graphical history (5) that allowed users to easily back up to a previous design. In contrast, DigiQuilt only provided undo and redo buttons (Figure 1, Point 6).

While Squeak supports *concurrency* at the implementation level, concurrency at the application level is not always a good idea. For instance, saving an application that is concurrently being worked on is problematically ambiguous. To solve this problem, we implemented a menu bar (6).

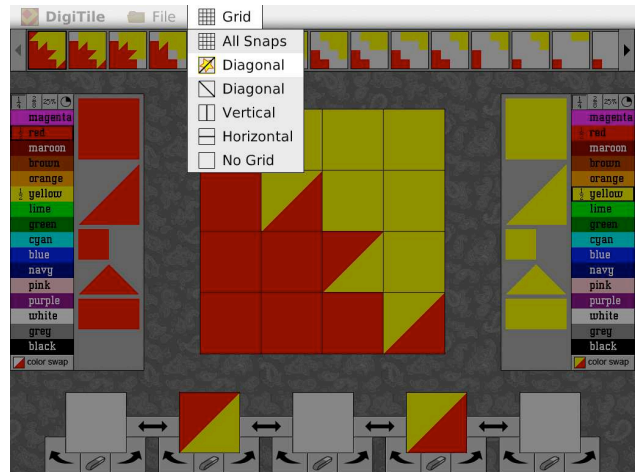


Figure 3. Selecting a grid to test for symmetry



Figure 4. Saving a tile design

When a menu is pressed, the rest of the interface becomes unavailable, signified graphically by darkening the application area (Figure 3). Other elements of the interface were relatively insensitive to concurrent interaction. Even if two tiles were dropped simultaneously, they could be processed separately. The only practical difference being that two history positions would have been created, rather than one.

Using the menu, users launch file functions (new, open, save, etc.) and can apply different grids. Different interfaces appear for saving (Figure 4) and opening (Figure 5) designs. The grid menu enables feedback on symmetry. When a line of symmetry is selected, the corresponding icon appears next to the “Grid” menu label; that icon shows a star when that line of symmetry is valid (see the diagonal icon in Figure 3).

Because of collaboration, the learning process in Digi-



Figure 5. Opening an existing tile

Tile is different than the learning process in DigiQuilt. To effect convergent conceptual change, DigiTile use should encourage different perspectives for different users. To further this, we provide different ways for the two learners to analyze the quilt. DigiQuilt displays the fraction that is covered by a color on the respective colored button. DigiTile allows the user to change the representation from reduced fraction to least-common-divisor fraction to percentage to visual pie chart. One learner could display his or her palette with percentages, while the other displays reduced fractions. This allows for us to create challenges like “a tile that is $\frac{1}{2}$ red and 50% yellow.” This challenge allows the learners to discover that $\frac{1}{2}$ and 50% are the same. Using *multiple-linked representations* to represent a mathematical concept leads to deeper math understanding [10]. This strategy would not work as well in the single-user DigiQuilt.

3.4 Evaluating DigiTile

Since the focus of this paper is the design process, we do not have room to detail study results. Instead, we show how learning theory and previous work informed the design of the evaluation. Previous research on DigiQuilt has already engaged many of the questions of whether tiling is a useful medium to engage with mathematics and art. This allows us to concentrate on questions that arise from adapting the interface to a multi-touch table. As HCI researchers, we are interested in how the multi-touch table interface can (better) support collaborative authoring.

An initial user study was conducted with 10–12 year olds to investigate whether and how pairs of learners collaborate using DigiTile. In particular, we were interested in whether the kind of reflective dialogue, that Roschelle based convergent conceptual change on, occurred in this design task. After a brief familiarization phase, we assigned pairs

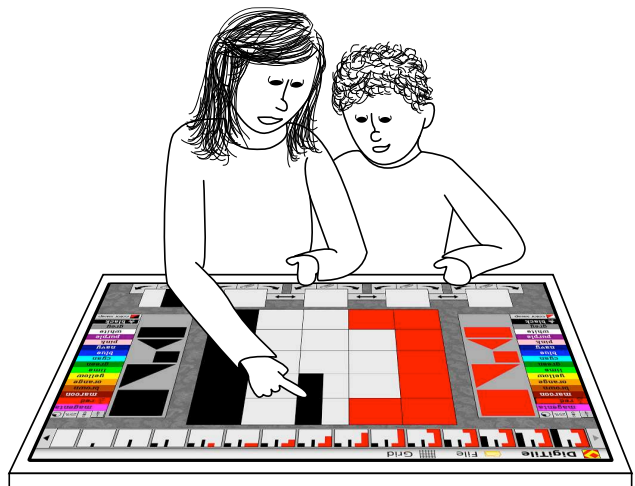


Figure 6. Working together on a difficult challenge

mathematical challenges. The goal was to see how pairs would collaborate when faced with a difficult challenge—one that required significant thought and work. Some challenges proved trivial and were accomplished in less than a minute with minimum planning and negotiation between the participants. We increased the difficulty until the task became challenging. At that point, the pairs had a harder time forming and trialing approaches. Because both participants could interact simultaneously, it was necessary for one participant to recruit the other in trialing an approach (Figure 6).

This initial study confirmed that DigiTile could encourage the kind of collaboration we were looking for. It also gave us feedback on how learners dealt with the different tools (graphical history, fraction representations, grid menu, etc.) in the full version of DigiTile. Even before this study, we believed a tailored version, leaving out features not necessary to the challenge at hand, would suit the learning task better. Yet, we left all features in as the study was largely formative, allowing us to see how intended users engaged the tools and how the interface could be improved.

In a follow-up study, we worked with DigiTile in a classroom, where the teacher felt that DigiTile’s focus on fractions would be appropriate. We worked with the teacher and an educational specialist to streamline DigiTile for that setting. To reduce complexity, we reduced the number of colors and shapes (Figure 7). Reducing the shapes meant that the fractions would no longer be as complex. With all the shapes available, it was fairly easy to place pieces to create denominators of 256 or 400 (larger than the learners were familiar with); in the simplified version, denominators are unlikely to reach above 64 (with a 4x4 grid) or 100 (with a

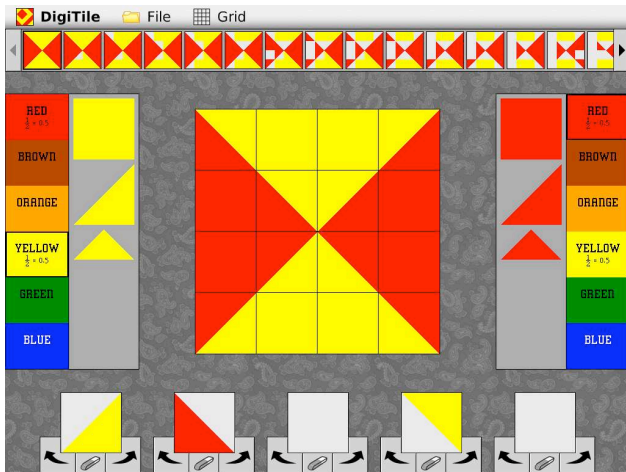


Figure 7. Using the simplified version for a training task

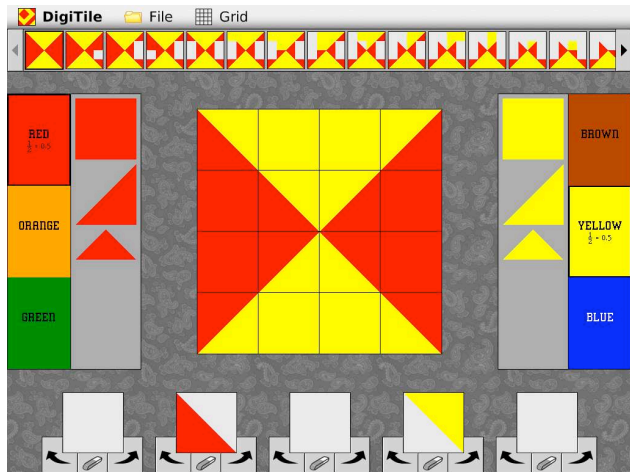


Figure 8. Using the split-palette version for a training task

5x5 grid). Reducing the number of colors removed colors, such as cyan, that children were not that familiar with. It also gave us more space to represent the fractions. As one of the learning goals in the curriculum was converting from fraction to decimal representation, we showed both the fraction and decimal representation simultaneously (underneath the color name).

As researchers, we were still interested in how different features of the interface would affect the collaboration. One way to enforce collaboration is to allocated resources unevenly. So, we created another version of DigiTile that split the colors between the users (Figure 8). We also designed challenges, such as design a tile that is $\frac{1}{2}$ red and $\frac{1}{2}$ yellow, that take advantage of this split. In this second study, we will compare the collaboration of the simplified version with that of the split version. Is the split effective in promoting collaboration? Is there more reflective dialogue in one case? As of the writing, this study has been conducted, but the data has not been analyzed.

4 Conclusions

Creating meaningful applications for novel interfaces is an important challenge for HCI researchers. There are different approaches to solve this problem. For instance, the Equator Project applied creative exploration as the basis for development [20]. Here, we made the argument for adapting an existing application. With the DigiTile case study, we concretely demonstrated both the benefits and the challenges in implementing this approach.

While DigiTile is only one example of adapting to a multi-touch table, we believe many of the issues presented here are universal. Implementation issues of group / table

size, orientation, and collaboration are inherent in the design space of multi-touch tables. By giving a concrete example, we also demonstrate the specific benefits of adapting. For example, while code reuse may seem like a tempting benefit of adapting, we show the difficulties in realizing this. In contrast, design work and research on the source application was beneficial in motivation, design, and evaluation. DigiQuilt had an extensive design process, including paper prototypes and classroom-evaluated design iterations [18]. It took several years to evolve DigiQuilt; it took less than half a year to create DigiTile.

Adding collaboration changed how the application was used. Rather than seeing this as a problem, we see this as an opportunity to utilize the strengths of the new interface. Multi-touch tables are better suited to collocated collaboration than a desktop setup. As learning can benefit from collaboration, DigiTile is more than just a port of DigiQuilt. The learning scenario and the interface were changed to take advantage of the multi-touch interface. The representation of fractions and the role of the challenges were significantly changed to suit the new learning paradigm. By adapting DigiQuilt, we were able to create a novel application to support collaborative learning with a multi-touch table.

5 Acknowledgements

The work is part of the ShareIT project funded by the EPSRC, grant number EP/F017324/1. We would like to thank our other collaborators in that project for their helpful feedback: (alphabetically) Sheep Dalton, William Farr, Rowanne Fleck, Amanda Harris, Eva Hornecker, Paul Marshall, Richard Morris, Nadia Pantidi, and Nicola Yuill. Spe-

cial thanks go to K.K. Lamberty for supporting us in adapting DigiQuilt and Kim Bryant for help in creating the figures. We thank MERL for loaning us the DiamondTouch.

References

- [1] J. D. Bolter and R. Grusin. *Remediation: Understanding New Media*. The MIT Press, Cambridge, MA, 1999.
- [2] P. Dietz and D. Leigh. DiamondTouch: A multi-user touch technology. In *Proceedings of UIST '01*, pages 219–226, New York, 2001. ACM Press.
- [3] C. Forlines, C. Shen, D. Wigdor, and R. Balakrishnan. Exploring the effects of group size and display configuration on visual search. In *Proceedings of CSCW '06*, pages 11–20, New York, 2006. ACM Press.
- [4] L. Gitelman. *Scripts, Grooves, and Writing Machines: Representing Technology in the Edison Era*. Stanford University Press, Stanford, CA, 1999.
- [5] M. Guzdial and K. Rose, editors. *Squeak: Open Personal Computing and Multimedia*. Prentice Hall, Upper Saddle River, NJ, 2002.
- [6] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of UIST '05*, pages 115–118, New York, 2005. ACM Press.
- [7] S. Hodges, S. Izadi, A. Butler, A. Rustemi, and B. Buxton. ThinSight: Versatile multi-touch sensing for thin form-factor displays. In *Proceedings of UIST '07*, pages 259–268, New York, 2007. ACM Press.
- [8] K. M. Inkpen, W. ling Ho-Ching, O. Kuederle, S. D. Scott, and G. B. D. Shoemaker. "This is fun! We're all best friends and we're all playing": Supporting children's synchronous collaboration. In *Proceedings of CSCL '99*, pages 252–259, Mahwah, NJ, 1999. Lawrence Erlbaum Associates.
- [9] M. Kaltenbrunner and R. Bencina. reactivation: A computer-vision framework for table-based tangible interaction. In *Proceedings of TEI '07*, pages 69–74, New York, 2007. ACM Press.
- [10] J. Kaput. Linking representations in the symbolic systems of algebra. In S. Wagner and C. Kieran, editors, *Research agenda for mathematics education: Research issues in the learning and teaching of algebra*, pages 167–194. National Council of Teachers of Mathematics, Reston, VA, 1989.
- [11] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
- [12] K. K. Lamberty. *Getting and Keeping Children Engaged with a Constructionist Design Tool for Craft and Math*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, May 2007.
- [13] K. K. Lamberty and J. L. Kolodner. Exploring digital quilt design using manipulatives as a math learning tool. In P. Bell, R. Stevens, and T. Satwicz, editors, *Proceedings of ICLS 2002*, pages 552–553, Mahwah, NJ, 2002. Lawrence Erlbaum Associates.
- [14] M. R. Morris, A. M. Piper, A. Cassanego, and T. Winograd. Supporting cooperative language learning: Issues in interface design for an interactive table. Research Report 8, Department of Computer Science, Stanford University, Palo Alto, CA, 2005.
- [15] S. Papert. Situating constructionism. In I. Harel and S. Papert, editors, *Constructionism*, pages 1–13. Ablex, 1991.
- [16] A. M. Piper, E. O'Brien, M. R. Morris, and T. Winograd. SIDES: A cooperative tabletop computer game for social skills development. In *Proceedings of CSCW '06*, pages 1–10, New York, 2006. ACM Press.
- [17] M. Resnick, A. Bruckman, and F. Martin. Pianos not stereos: Creating computational construction kits. *Interactions*, 3(6):41–50, 1996.
- [18] J. Rick and K. K. Lamberty. Medium-based design: Extending a medium to create and exploratory learning environment. *Interactive Learning Environments*, 13(3):179–212, 2005.
- [19] Y. Rogers, Y.-K. Lim, and W. R. Hazlewood. Extending tabletops to support flexible collaborative interactions. In *Proceedings of TABLETOP '06*, pages 71–78, Washington, DC, 2006. IEEE Computer Society.
- [20] Y. Rogers, M. Scaife, E. Harris, T. Phelps, S. Price, H. Smith, H. Muller, C. Randell, A. Moss, I. Taylor, D. Stanton, C. O'Malley, G. Corke, and S. Gabrielli. Things aren't what they seem to be: innovation through technology inspiration. In *Proceedings of DIS 2002*, New York, 2002. ACM Press.
- [21] J. Roschelle. Learning by collaborating: Convergent conceptual change. In T. Koschman, editor, *CSCL: Theory and Practice of an Emerging Paradigm*, pages 209–248. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.
- [22] K. Ryall, C. Forlines, C. Shen, and M. R. Morris. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *Proceedings of CSCW '04*, New York, 2004. ACM Press.
- [23] S. D. Scott, K. D. Grant, and R. L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In K. Kuutti, E. H. Karsten, G. Fitzpatrick, P. Dourish, and K. Schmidt, editors, *Proceedings of ECSCW 2003*, pages 159–178, Netherlands, 2003. Kluwer.
- [24] C. Shen, F. D. Vernier, C. Forlines, and M. Ringel. DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proceedings of CHI '04*, pages 167–174, New York, 2004. ACM Press.
- [25] E. Soloway, M. Guzdial, and K. E. Hay. Learner-centered design: The challenge for HCI in the 21st century. *Interactions*, 1(2):36–48, April 1994.
- [26] J. Stewart, B. B. Bederson, and A. Druin. Single display groupware: A model for co-present collaboration. In *Proceedings of CHI '99*, pages 286–293, New York, 1999. ACM Press.
- [27] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *Proceedings of CHI 2006*, New York, 2006. ACM Press.
- [28] E. Tse, S. Greenberg, and C. Shen. GSI Demo: Multiuser gesture / speech interaction over digital tables by wrapping single user applications. In *Proceedings of ICMI '06*, pages 76–83, New York, 2006. ACM Press.
- [29] L. S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.