A Pattern Language for Interactive Tabletops in Collaborative Workspaces

CHRISTIAN REMY, MALTE WEISS, MARTINA ZIEFLE, and JAN BORCHERS, RWTH Aachen University

In this paper, we present a Human-Computer Interaction (HCI) design pattern language that bundles existing knowledge on tabletop design and offers solutions to recurring problems. Our patterns enable not only developers, designers, and domain experts to improve their existing systems and facilitate the design process of new systems, we also encourage novice users to comprehend the variety of tabletop research and commercial products in this domain. We consider our language as a starting point to create a sustainable body of knowledge that will be extended and refined by the community.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Standardization*

General Terms: Design, Human Factors, Standardization, Languages

Additional Key Words and Phrases: HCI design patterns, guidelines, interactive tabletops

ACM Reference Format:

Remy, C., Weiss, M., Ziefle, M., and Borchers, J. 2010. A Pattern Language for Interactive Tabletops. jn 1, 1, Article 1 (July 2010), 48 pages.

1. INTRODUCTION

Interactive tabletops offer a variety of benefits over desktop computers. They allow direct manipulation of on-screen objects and provide a natural setting for collaborative work. As building multi-touch devices becomes more feasible, researchers start to focus on interactive tabletops by creating many new technologies and exploring novel interaction techniques that reach beyond the well-known desktop metaphors. These findings are now more and more expanding into the commercial market. New consumer devices are released that can be used in everyday life, such as the Microsoft Surface, the SMART Table, or the Apple iPad. However, it becomes increasingly hard for interactive tabletop designers to inform their design decisions from existing findings in this field. There are hardly any guidelines to help designers during the process of building a tabletop and developing applications for it. This might finally slow down the process of innovation and refinement.

In this paper, we present an HCI design pattern language to overcome this lack of information. The patterns are gathered by identifying recurring problems with approved solutions in tabletop design and compiling these into an established format of HCI design patterns. This allows not only designers to justify their design decisions using other proven solutions, it also gives practitioners an insight into the current state of the research field and its solutions to problems they encounter. Thereby, the format of HCI design patterns offers a more suitable format

Author's address: Christian Remy, RWTH Aachen University, Media Computing Group, 52056 Aachen, Germany, remy@cs.rwth-aachen.de; Malte Weiss, RWTH Aachen University, Media Computing Group, 52056 Aachen, Germany, weiss@cs.rwth-aachen.de; Martina Ziefle, RWTH Aachen University, eHealth Group, 52062 Aachen, Germany, ziefle@humtec.rwth-aachen.de; Jan Borchers, RWTH Aachen University, Media Computing Group, 52056 Aachen, Germany, borchers@cs.rwth-aachen.de

than, e.g., guidelines. HCI domain experts agreed that the format of design patterns is in particular suitable to teach usability and deliver knowledge of the HCI design process[1].

As aforementioned, this pattern language addresses not only domain experts, but also novice users. For example, occasional tabletop users can learn more about these systems without having to invest hours in reading scientific papers, or customers who want to order a personalized tabletop can acquire tabletop knowledge to have a common vocabulary when explaining their needs and demands to the designers and developers.

2. FORMAT

The format of the patterns follows the HCI design pattern definition developed at the CHI 2000 workshop[2] and later similar events. It is heavily oriented towards the format of Alexander's architecture pattern language[3] and Borchers' HCI design patterns for interactive exhibits[4]. In the following, we give a brief overview of our layout decisions.

Each pattern *name*, written in small caps, is preceded by a unique *ID* and followed by a *ranking* from zero to two asterisks. A higher ranking, i.e., more asterisks, indicate a pattern that offers more likely the best possible solution for the encountered problem. An *illustration* serves as sensitizing example and–together with the *context*–concludes the introduction of the pattern. While the title gives the reader a short idea of the pattern's solution, the context serves as orientation if this pattern is applicable for the specific design scenario.

The body of each pattern is separated from its header and footer by a small horizontal line, similar to the three diamonds in Alexander's and Borchers' language. Printed in bold, the *problem statement* and the *solution* enframe the extensive *problem description*. In this part, we explain the problem and its forces in detail, and mention examples to show that the solution has been applied to the real world. Since tabletops are not as widespread as other electronic devices, the forces are not always as obvious as in more conventional application domains, e.g., desktop user interfaces or mobile device interaction. The examples help to clarify the forces, therefore it is not always possible to clearly distinct these parts of the problem description. This inductive approach also enables novice readers to get the general concept and make sure they completely understand the solution statement. The sketched *diagram* closes the body of the pattern and gives a general impression of the solution without being too specific or showing unnecessary details.

Separated with another horizontal bar, the *references* part names further patterns from the pattern language to consider. This part is similar to the context, but contains links into the other direction, i.e., every pattern mentioned in a context points to this pattern in its references part as well and vice versa. The final part in our patterns is a short bibliography of the examples cited in the pattern body. We decided to put the references at the end of every pattern for specific reasons. First, this enables readers to take out single patterns without losing the references to applied examples mentioned in the pattern. Second, even when browsing the complete pattern language, the named sources are easily visible. This reduces reading time since it avoids unnecessary switching back and forth between a pattern and the bibliography.

[1] J.O. Borchers and J.C. Thomas, Patterns: What's In It For HCI?. *Ext. Abstr. CHI '01*, pp. 225–226.
[2] R. Griffiths, L. Pemberton, J. Borchers, and A. Stork, Pattern Languages for Interaction Design: Building Momentum. *Ext. Abstr. CHI '00*, p. 363.
[3] C. Alexander, S. Ishikawa, and M. Silverstein, A Pattern Language: Towns, Buildings, Construction. *Oxford University Press*, 1977.
[4] J. Borchers, A Pattern Approach to Interaction Design. *John Wiley & Sons*, 2001.

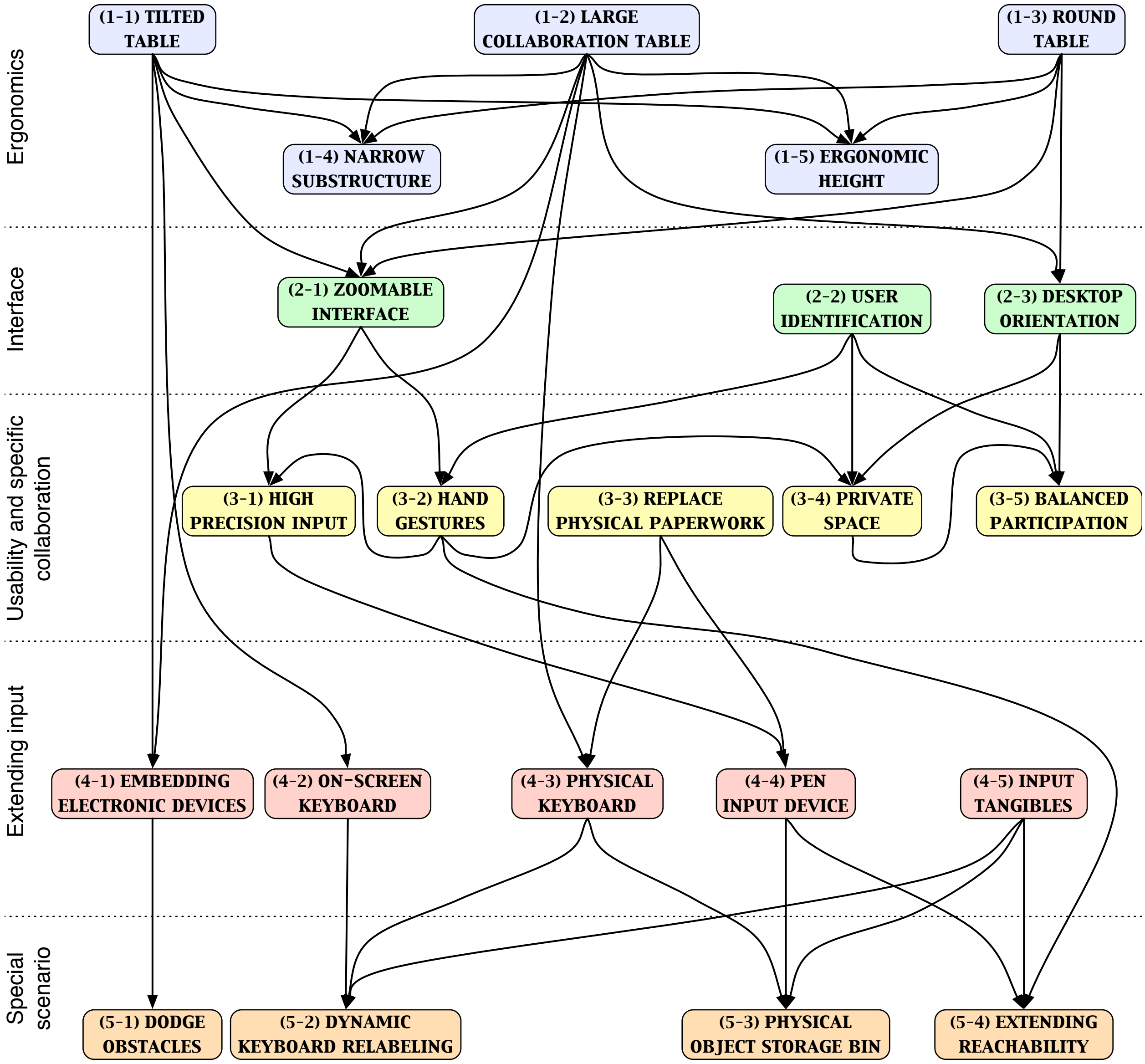


Fig. 1. Tabletop pattern language graph

3. STRUCTURE

The structure of our pattern language is illustrated in Figure 1. The patterns are categorized in five different levels, with a distinct color for each of them. The first digit of each pattern ID denotes one particular category. The categories are

(1-x) ergonomics patterns (blue),

(2-x) interface patterns (green),

- **(3-x)** usability and specific collaboration patterns (yellow),
- **(4-x)** extending input patterns (red), and
- **(5-x)** special scenario patterns (orange).

The higher a pattern is located in the graph the larger its scale of application is, and the earlier it is applicable during tabletop design. Patterns on the bottom address more specific tabletop problems that often depend on a particular scenario.

4. FUTURE WORK

We consider our language as a start for the tabletop community to create a comprehensive set of guidelines by gathering the collective knowledge of research and commercial products. Hence, there is much potential to extend our language. We want to encourage tabletop design experts to participate in the process of identifying additional patterns as well as improving existing ones. Furthermore, we can imagine other pattern languages that cover different aspects of interactive tabletops, such as tracking technologies, software frameworks, or applications and scenarios especially expedient for interactive tabletops. We hope that by connecting this interdisciplinary know-how and gathering the collective experience, the pattern language helps to share expert knowledge, aids novices to familiarize with the domain, and brings the community closer together.

5. ACKNOWLEDGEMENTS

This work was funded in part by the German B-IT Foundation. We would like to thank Christian Kohls for his great advices and helpful suggestions to improve our paper during the shepherding process, which significantly enhanced the quality of the submission. Furthermore, thanks to Mike Kremer for drawing the diagram sketches, Simon Jakubowski for taking the photographs for the pattern illustrations, and Simon Völker for his support and ideas regarding the pattern content.

(1-1) TILTED TABLE *

Fig. 2. A flexible, tiltable tabletop display. Courtesy by Leitner et al. [1].

... you are in the early stages of designing a tabletop that will only be used by a single user. Tasks do not include many large objects on a table, and the user will make extensive use of direct touch, e.g., for layout or design scenarios.

Working for a long period of time on a multi-touch wall can be exhausting. On the other hand, staring down on a horizontal surface can induce neck pain.

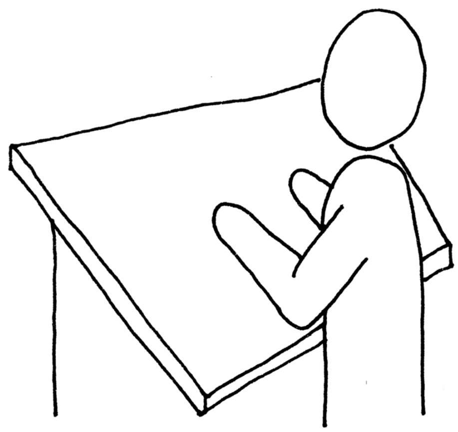
There are many arguments for both horizontal and vertical displays. A board is usually used for presenting your drawings to an audience, e.g., in a classroom or lecture hall. But also for a list of milestones, design guidelines, or project notes users put a board on their wall, even in a single-user environment. One reason why users do this, instead of putting the information on a sheet of paper on their desk, is that you cannot accidentally occlude them with books, cups, notes, or other obstacles on your table. It is also more comfortable to read information at eye level instead of looking down all the time. On the other hand, in some situations a vertical surface is not suitable for some tasks. Consider text input, which can be done in different ways. For example, a pen is the usual device for whiteboards, and mimics the input metaphor of chalk on blackboards. This kind of input originally did not allow to put your hand on the surface, as it would smudge the chalk. It is also an uncomfortable position for your hand while writing on a wall, opposed to a sheet of paper on a horizontal surface. Another, more preferable text input method is a keyboard, like an (4-2) ON-SCREEN KEYBOARD or a (4-3) PHYSICAL KEYBOARD. Both are almost unusable in a vertical alignment, especially the physical keyboard, which first would have to be mounted on the surface. An observation of knowledge workers from the tabletop domain by Morris et al. [2] stated informal guidelines about the setup of interactive surfaces, including the advice to support a display that can be tilted.

A study named Tilted Tabletops [3] examined the effect on collaboration tasks for groups at tabletops with different angles. The experiment with 78 participants revealed that users in general preferred the tilted setup over horizontally or vertically aligned surfaces. Quite astonishing in this study seems the fact that higher tilt angles (i.e., more upright positions) were preferred over a light tilt even for writing tasks. This maybe due to similarities with whiteboards where users put down short handwritten notes. However, it should be kept in mind that higher angles prohibit placing physical objects on the tabletop, which was not considered in this specific user test. An adjustable tilt angle solves this conflict; users that prefer higher angles can turn the tabletop if there are no objects placed on the table.

In single user setups [4] and remote collaboration tasks [5], tilted displays clearly outperform horizontal and vertical multi-touch displays. For collaborative displays, this can sometimes result in awkward arrangements for the users. This can be avoided by choosing a (1-2) LARGE COLLABORATION TABLE, such that every user can look on the tilted table from a comfortable angle. Additionally, the tilting support should be optional and not be mounted with a fixed angle. This offers users also an alternation of the setup to avoid neck pain from looking down on a horizontal surface or arm fatigue on a multi-touch wall, when the users have to use the display for a lot of text input.

Therefore:

Use a tilted table in an angle most comfortable to the user. If possible, make the angle flexible to account for better collaboration or placement of physical objects.



Since physical objects would probably fall down from the tilted table, proper use of an (4-2) ON-SCREEN KEYBOARD should be considered. Your tabletop will probably be used by a single user, therefore you might offer personalization of screen data by (4-1) EMBEDDING ELECTRONIC DEVICES. Another thing to look into are a (1-4) NARROW SUBSTRUCTURE and an (1-5) ERGONOMIC HEIGHT, which are easier to achieve on a tilted table even with bottom-projection due to the specific angles for cameras and projectors located behind the surface...

References

[1] J. Leitner, J. Powell, P. Brandl, T. Seifried, M. Haller, B. Dorray, P. To, Flux: A Tilting Multi-Touch and Pen Based Surface. *Proc. CHI '09*, pp. 3211–3216.
[2] M. Morris, B.A.J. Brush, and B. Meyers, A Field Study of Knowledge Workers' Use of Interactive Horizontal Displays. *Proc. TABLETOP '08*, pp. 105–112.
[3] C. Muller-Tomfelde, A. Wessels, and C. Schremmer, Tilted Tabletops: In Between Horizontal and Vertical Workspaces. *Proc. TABLETOP '08*, pp. 49–56.
[4] D. Wigdor, G. Perm, K. Ryall, A. Esenther, and C. Shen, Living with a Tabletop: Analysis and Observations of Long Term Office Use of a Multi-Touch Table. *Proc. TABLETOP '07*, pp. 60–67.
[5] H. Ishii and M. Kobayashi, ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. *Proc. CHI '92*, pp. 525–532.

(1-2) LARGE COLLABORATION TABLE *

Fig. 3. A group of students playing a game on a large interactive tabletop.

... your tabletop will be used in a collaborative workspace where a large number of users interacts with the surface simultaneously. You have yet to decide which hardware you will use, but you already know the estimated amount of users and possibly also the projected tasks.

Every user wants sufficient space for comfortable interaction, but you do not want to arise interaction issues like objects being out of reach. The tabletop should also work as a ubiquitous furniture and not as an object spatially dividing the participants and invoking a large distance.

If a tabletop is too small, it raises many problems like insufficient screen space, restricting wide-ranging gestures, and visibility issues as the number of collaborators increases. For example, in a meeting, everyone needs enough space to take down notes, or on an exhibit tabletop a large number of users should be able to explore the media together. On the other hand, if the table is sized too big, the practitioners cannot reach the entire area and you are facing reachability issues. You can always solve these problems with other techniques, such as (3-2) HAND GESTURES, but you will lose your ubiquitous feeling that the tabletop just supports the discussion. It will more become the center of the meeting, and technical details should not become the topic of a meeting in general, unless it was the original intention. Large tables can also create a distance feeling among users, such that they feel spatially divided by it and they behave more passive during the discussion.

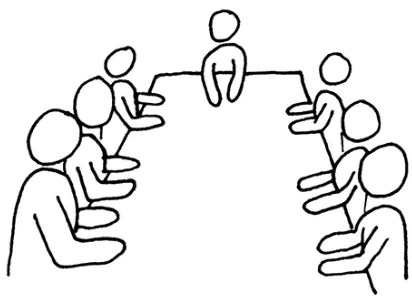
Multi-touch tables can have many different sizes, from devices sized as small as a tablet to huge walls, as large as the users are able to reach or even slightly beyond. In most cases, your choice is obviously not easy to undo later on, so it should be well-justified. Elaborate the projected number of users, prototype basic tasks with the minimum and maximum number of practitioners to get the best size for your table. For the same number of people, different tasks may require different table sizes; e.g., working with four people on one list of items requires a smaller table compared to four people working on four separate graphical tasks. Combining multiple tables can also suffice the need for space [1].

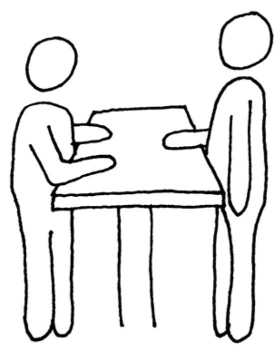
Inkpen et al. [2] conducted an extensive study about how the tabletop setup affects the collaboration, with display size one of four factors that they evaluated. They conclude that a display should be small enough to fit into the user's view, and large enough to give each user sufficient space to interact with one another. In comparison to the larger display this setup resulted in an even more distribution of touched areas.

A design space of tabletop hardware, evaluated by Grossman and Wigdor [3], also stresses the importance of the display size selected for the tabletop. In a similar taxonomy, focusing rather on the collaboration aspect on multi-user display devices than the hardware, Terrenghi et al. [4] emphasize the impact on the display's size on collaboration. While not limited to interactive tabletops, they list different sizes of devices and the implications for the interaction between users and the device that should be considered in designing such systems.

Therefore:

Build a tabletop that offers enough space for all users and their expected tasks. The size should be equal to a traditional non-interactive table where users would work in a similar scenario.





The larger the table, the more functionality is needed for a reasonable display and tracking quality, but you should still consider a (1-4) NARROW SUBSTRUCTURE and an (1-5) ERGONOMIC HEIGHT. Large areas on the tabletop can offer a (3-4) PRIVATE SPACE for users, and you can also take advantage of your available space by putting additional input objects on the table, e.g., (4-1) EMBEDDING ELECTRONIC DEVICES like cell phones or laptops, or a (4-3) PHYSICAL KEYBOARD, when you expect a lot of text input...

References

[1] K. Ryall, C. Forlines, and C. Shen, Exploring the Effects of Group Size and Table Size on Interactions with Tabletop Shared-Display Groupware. *Proc. CSCW '04*, pp. 284–293.
[2] K. Inkpen, K. Hawkey, M. Kellar, and R. Mandryk, Exploring Display Factors that Influence Co-Located Collaboration: Angle, Size, Number, and User Arrangement. *Proc. HCI '05*.
[3] T. Grossman and D. Wigdor, Going Deeper: a Taxonomy of 3D on the Tabletop. *Proc. TABLETOP '07*, pp. 137–144.
[4] L. Terrenghi, A. Quigley, and A. Dix, A taxonomy for and analysis of multi-person-display ecosystems. *Personal Ubiquitous Computation*, vol. 13, 2009, pp. 583–598.

(1-3) ROUND TABLE *

Fig. 4. The reacTable. Courtesy by Jordá et al. [1].

... you just started building the concept for your tabletop and are looking into basic form factor decisions. Your tabletop will not be used for single-user tasks like office work or text reviews, but rather serve as, e.g., the center of a team meeting or as an interactive tabletop for an exhibit. All users at the table are supposed to interact on the same level, without privileging a specific user.

In a collaborative environment, many participants work around a table simultaneously, but everybody wants to have the windows and the table aligned towards herself.

On a rectangular tabletop there are often positions for collaborators where they feel uncomfortable, especially if there are more than four users. For example, sometimes participants join the discussion and have to sit at a corner or at the edge of the table from where it is hard to reach the other side of the table. A circular tabletop overcomes this issue, since the table looks the same from every position. Every collaborator has the same amount of table space in their area of reach, and if the windows are aligned and distributed in a circular way, no one has a superior look on the desktop (see (2-3) DESKTOP ORIENTATION).

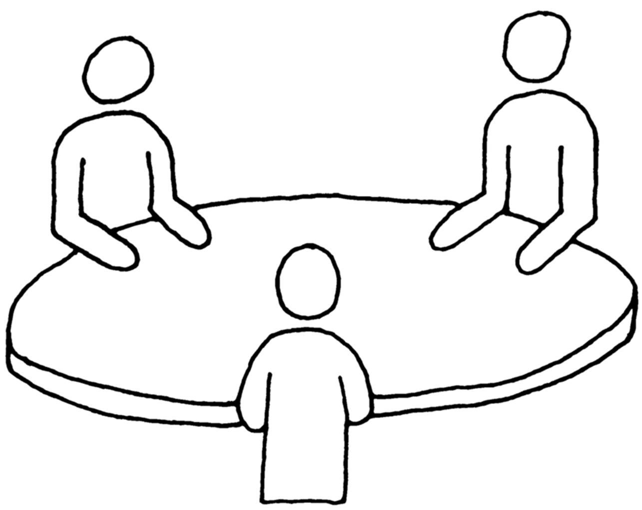
One of the best-known circular tabletops is the reacTable [1]. Its main application is a music demonstration where users can place tangibles on the table and interact with them to emit unique sounds. The reacTable encourages participants to mix different sounds created with their tangible objects in hand, thus creating a concert of ambient music. Due to the round table, every collaborator is at the same level and no one is implicitly acting as a principal musician or orchestra leader.

In the same area of producing music, the SoundScape renderer built by Bredies et al. [2] is also a round tabletop, though the interaction is completely different. The SoundScape renderer is located in the center of a circular room, such that the display mimics the spatial setup of the environment. Users can then directly manipulate sound sources indicated by spots on the table, arrange them as they want and change their volume independently. A small set of (3-2) HAND GESTURES allows for more input variety.

Another example for a round table is given by Koike et al. [3], with the addition that it is also rotatable, which was the main focus of this research. The possibility to easily rotate the tabletop by putting wheels beneath the table is another advantage of a round table; in their work, a round rotary surface was mounted on a larger rectangular table, which was sufficient for their purpose of user tests. Since some of the advantages are not true anymore in this setup, e.g., there are still corners that may decrease the reach distance for some users, this is not a good example for a round table, unless you want to switch between round and rectangular environments for some reasons in your project. Another disadvantage that is clearly visible in this solution applies to most round tabletop setups: unless you are using a special projector, a non-rectangular image is simply created by not displaying the pixels in the corners of the screen. Hence, a round tabletop is either reduced in screen real-estate due to less pixels or it is more expensive, when using multiple projectors or a special device that displays a round image.

Therefore:

Build a round table or a rectangular table with round corners.



When building the hardware, choose an (1-5) ERGONOMIC HEIGHT such that users will sit or stand comfortable at your table. Implement a (2-3) DESKTOP ORIENTATION algorithm for your displayed widgets to ensure the effect of your table shape can fully unfold and benefit the users...

References

[1] S. Jorda, G. Geiger, M. Alonso, and M. Kaltenbrunner, The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces. *Proc. TEI '07*, pp. 139–146.
[2] K. Bredies, N.A. Mann, J. Ahrens, M. Geier, S. Spors, and M. Nischt, The Multi-Touch SoundScape Renderer. *Ext. Abstr. AVI '08*, pp. 466–469.
[3] H. Koike, S. Kajiwara, K. Fukuchi, and Y. Sato, Information Layout and Interaction on Virtual and Real Rotary Tables. *Proc. TABLETOP '07*, pp. 95–102.

(1-4) Narrow Substructure **

Fig. 5. An exhibit tabletop (*Aachener Frieden*) with a tapered substructure to avoid visitors hitting the structure with their knees.

... you are designing a system for a multi-user environment, either a (1-2) Large Collaboration Table or a (1-3) Round Table that allows to sit or stand comfortably. For a single-user environment, you might have chosen a (1-1) Tilted Table.

Tables with top-projection have the occlusion problem, while a tabletop with a projector beneath the table does not leave much space for the legs of the user.

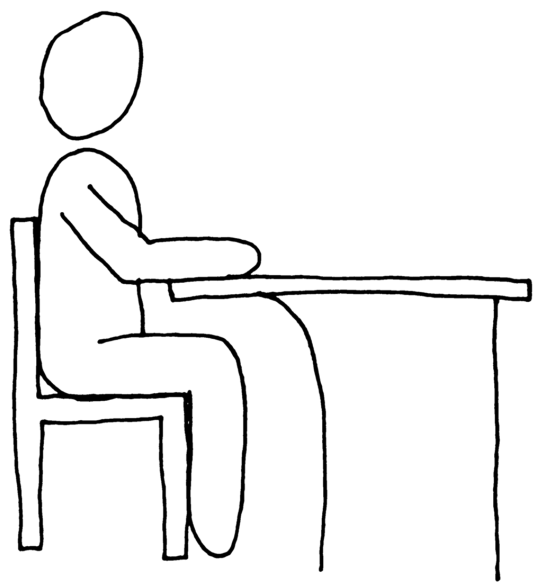
Multi-touch detection can be realized in many different ways. Both the detection of input and the display output might need devices that have to be placed in a reasonable distance on top or beneath the table. While top projection raises occlusion problems, which are especially distracting in multi-user environments, projectors and cameras beneath the table take a lot of space. Whenever users sit down at a table, there should be enough space for their legs. But also while standing at tabletops, this is an important issue, as users should not hit the substructure with their knees. A plasma or LCD display with touch sensing capability offers a surface without either of these problems; however, these displays are very expensive, fragile, and easily damaged, especially when mounted horizontally on a tabletop.

You can solve these problems without big changes in your hardware design which you created by using a mirror, as proposed by Masoodian et al. [1] in their paper for an environment to share documents on a tabletop. Projectors with a high projection angle, e.g., short-throw projectors, further improve the legroom since they enable even higher angles. Also consider user tests with prototypes to find out which substructure construction allows which range for the user's arms on the table. These prototypes can be really low-cost, made of wood or cheap metal, where you let the user grasp objects on the table and see if they touch the substructure with their knees. As long as users can reach the whole surface without hitting the substructure with their knees, the tabletop is small enough.

Some basic guidelines for legroom can be found in several standards published by authorities. For example, according to the ISO 9241-5 [2], the depth should be 60 cm to 80 cm in 12 cm height, measured from the floor, and 20 cm in 67 cm height, respectively. The latter requirement can be acceptable at 20 cm in 62 cm height, if the structure does not meet the first requirements. However, always keep in mind the scenario you will use the table in; e.g., children need smaller tables than adults, changing your blueprint considerably. Additionally, in a multi-user environment legroom might not be an issue as big as in a single-user task, since on every table users try to avoid hitting the legs of other users, thus keeping a larger distance than in a scenario where they are working at a table alone.

Therefore:

Make preliminary user tests with your hardware to check if the users have enough space for their legs and can sit comfortably at your tabletop. At all edges where you expect users to sit or stand, use hardware that allows for a narrow substructure, such as mirrors or special cameras and projectors.



This is a basic pattern with no further references within this language.

References

[1] M. Masoodian, S. McKoy, and B. Rogers, Hands-on Sharing: Collaborative Document Manipulation on a Tabletop Display Using Bare Hands. *Proc. CHINZ '07*, pp. 25–31.

[2] ISO 9241-5, Ergonomic requirements for office work with visual display terminals: Workstation layout and postural requirements

(1-5) ERGONOMIC HEIGHT *

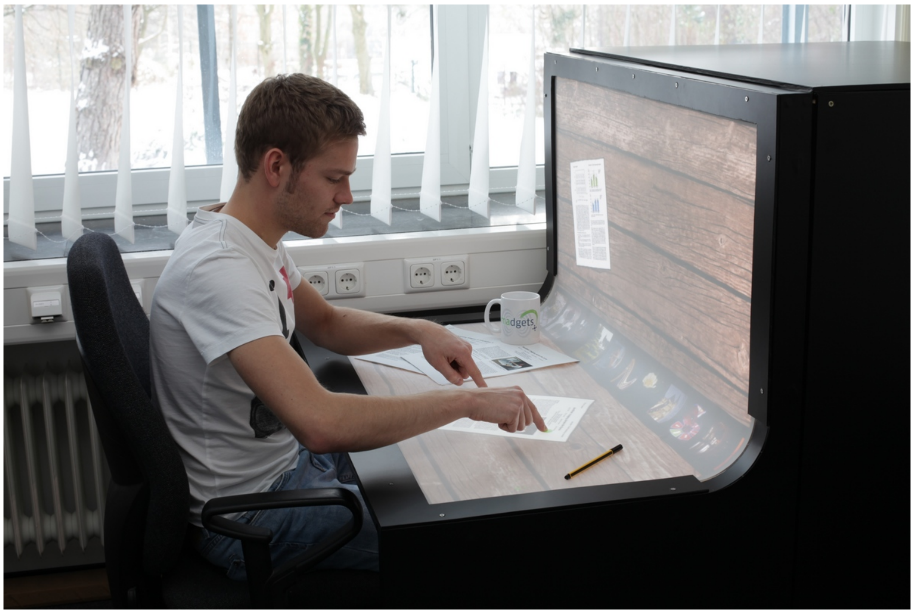


Fig. 6. BendDesk [1] was created according to the most comfortable height for users sitting at it.

... you have started to design your tabletop and you are in the very beginning of choosing the hardware specifications. Depending on the target audience and tasks, your system has a different setup, e.g., a (1-2) LARGE COLLABORATION TABLE or a (1-3) ROUND TABLE for a multi-user scenario, or a (1-1) TILTED TABLE for a single-user environment. You also know whether the users are sitting or standing at the table, and whether you are designing for children or adults.

The tabletop requires tracking technology to detect, e.g., finger touches or objects on the surface. With all the technology underneath the table, it may be too high for users to interact comfortably.

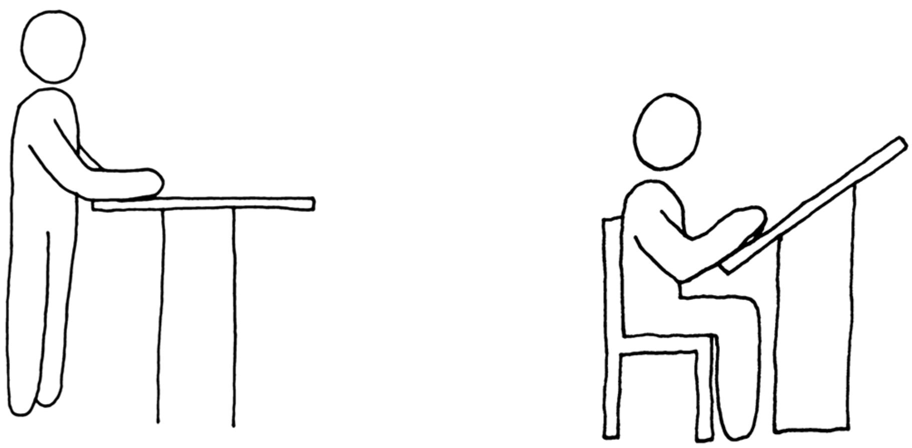
While designing your table, you might have several restrictions to the ergonomic design. If you have a rather large table with projector and cameras beneath it, you need a reasonable size for the projected image to fit the table measurements. On the contrary, if your table is too small, then you do not have a lot of space next to all the elements in the substructure, and while keeping a (1-4) NARROW SUBSTRUCTURE, you may tend to stack things up as much as possible. In both situations the height of your final tabletop has to be higher. But this possibly raises another issue depending on your scenario, e.g., it might be impossible for children sitting at your table to interact properly. On the contrary, if the tabletop is used in an exhibit with users walking around, it needs to be much higher than just the height dedicated by your technical limitations. Before building the basic hardware, you have to consider the application of your tabletop, e.g., by observing users in a scenario on a prototype table.

One of the drawbacks of direct touch input on tabletops is the fatigue of the users' arms. However, the intensity of exhaustion that users feel when using touch interaction is connected to the ergonomics of the table. Compare this to a desktop PC, where most users usually sit on a chair at a table with the keyboard, mouse, and display. To make this situation more comfortable, users arrange these devices such that their arms are placed in a natural way and it is not exhausting to work on a PC. Additionally, the chair is often height adjustable and tables have a uniform standard height. Of course you can choose this height for your tabletop, too. ISO 9241-5 [2] recommends 72 cm for ergonomic work space furniture, but keep in mind that this is for a scenario where the user sits at your table. The recommended height for standing at a work place with a monitor is 118 cm, however this is probably too high for touch interaction and needs further investigation.

In a long-term study, where a tabletop was designed for one particular user who did all his digital work on the multi-touch table, Wigdor et al. [3] built a table that let the user feel comfortable in both sitting and standing position. A good way to avoid that your construction will be too high or too low is a preliminary user test with a prototype, which can easily be done by taking an office table and chair, adjust its height and simulate some touchscreen tasks with pieces of paper that have to be sorted. Be sure that you check for both sitting and standing scenario, in case of sitting on a chair the chair should be the same that is used later on (e.g., height adjustable) and your table should have the same (1-4) NARROW SUBSTRUCTURE as your finished tabletop has to allow for sufficient legroom.

Therefore:

Consider the target user group and the working environment for your tabletop carefully and choose a suitable table height. If the users are children or they are intended to sit at the table, choose a small table height. To get the best results, build early prototypes and emulate interaction tasks.



This is a basic pattern with no further references within this language.

References

[1] M. Weiss, S. Voelker, C. Sutter, and J. Borchers, BendDesk: Dragging Across the Curve. *Proc. ITS '10*, pp 1–10.
[2] ISO 9241-5, Ergonomic requirements for office work with visual display terminals: Workstation layout and postural requirements
[3] D. Wigdor, G. Perm, K. Ryall, A. Esenther, and C. Shen, Living with a Tabletop: Analysis and Observations of Long Term Office Use of a Multi-Touch Table. *Proc. TABLETOP '07*, pp. 60–67.

(2-1) Zoomable Interface *

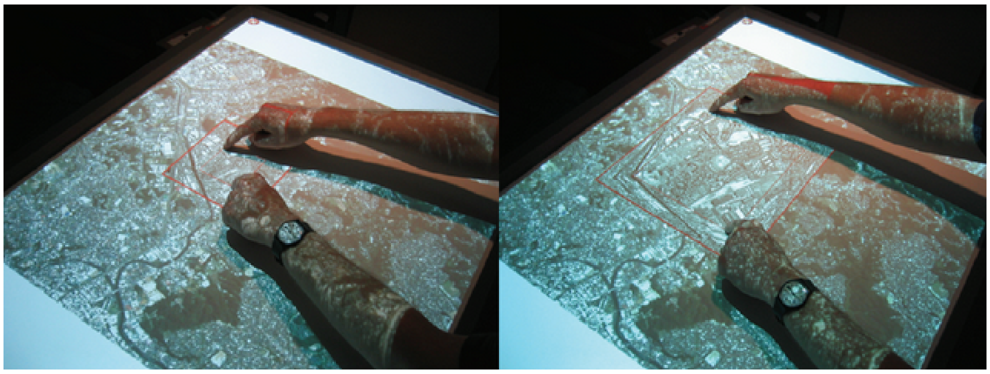


Fig. 7. User performing actions with the DTLens toolkit. Courtesy by Forlines and Shen [3].

... although your tabletop has a suitable size, you sometimes may encounter space issues. For example, when a large number of participants is working simultaneously on creating a large mind map out of brainstorming results, or on a (1-1) Tilted Table in a single-user environment.

A tabletop should be as large as possible such that not every interaction is limited by the physical edges of the table. But sometimes the desktop elements being displayed on the tabletop are so space-demanding that it is impossible to show them all at once.

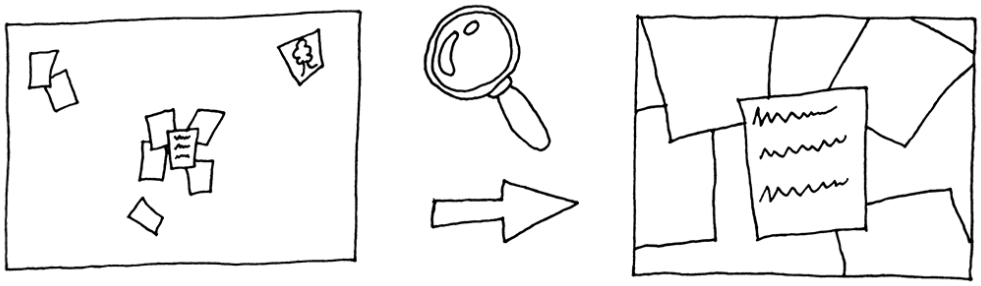
The idea of zoomable interfaces has often been mentioned in research, with one early concrete example proposed by Perlin and Fox [1] in their "Pad" interface. They strife to make interaction more natural by using geographic analogies for navigation. The interface is one flat two-dimensional space, where the user can zoom in and out to see more details or get a broader overview, respectively. Raskin built up on this approach and described ZoomWorld [2], imagining a novel interface without file names and other descriptors, just content where the user can browse intuitively using her real-world experience on navigating through large amounts of data.

With multi-touch tables and their natural input mechanism, it is possible to create interfaces that mimic the behavior of Pad and ZoomWorld. For example, the DTLens toolkit by Forlines and Shen [3] allows users to zoom on large data files, e.g., geological maps or pictures of space with high resolution. The two-handed zoom gesture is an example of the advantages that tabletops offer over traditional desktop computers for zoomable interfaces to realize these ideas that have been around for a long time.

The most sophisticated example of a zoomable interface can be experienced on a multi-touch device that is much smaller than a tabletop: Apple's iPhone [4]. The built-in web browser allows to visit websites that are designed to be viewed on large desktop displays. The user can zoom out to see the website as a whole, and with zooming into the content it is possible to click on tiny links or enter text in small text input areas. When building your tabletop think about adopting these ideas in order to have sufficient space at the end.

Therefore:

Let the user not only zoom into the content of your applications, but also into the interface elements itself.



To ease navigating through your tabletop environment and its widgets, (3-2) HAND GESTURES are commonly used and especially applicable to your zoomable interface. Also consider implementing (3-1) HIGH PRECISION INPUT methods in your tabletop software framework...

References

[1] K. Perlin and D. Fox, Pad: An Alternative Approach to the Computer Interface. *Proc. SIGGRAPH '93*, pp. 57–64.
[2] J. Raskin, The Humane Interface: New Directions for Designing Interactive Systems, Addison-Wesley Professional, 2000.
[3] C. Forlines and C. Shen, DTLens: Multi-User Tabletop Spatial Data Exploration. *Ext. Abstr. UIST '05*, pp. 119–122.
[4] Apple Inc., iPhone, http://www.apple.com/iphone/

(2-2) User Identification *

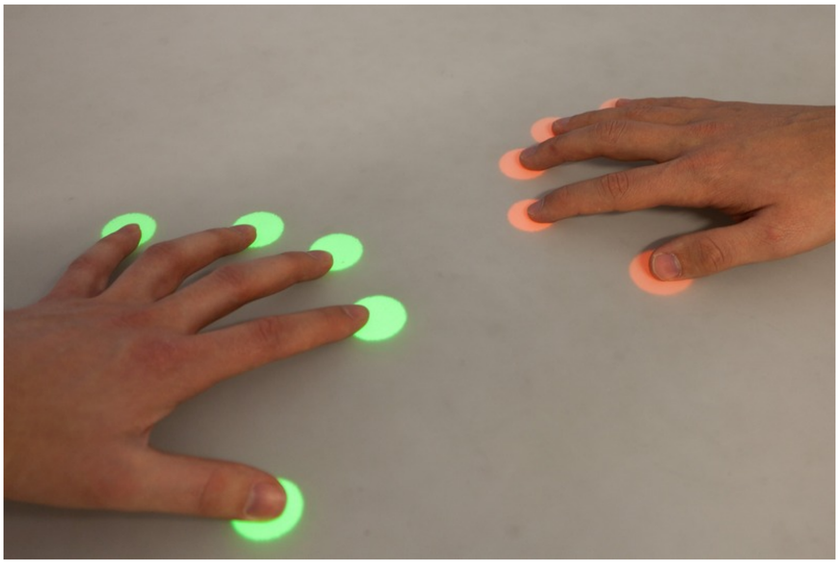


Fig. 8. Distinct colors for different users.

... your tabletop will be used in a setup where multiple users work simultaneously at a tabletop. Either on different tasks at the same time or together on a task where every user produces a semantically distinct input. For example, in a game where users own certain virtual objects, or in a meeting where they put down personal notes.

In a collaborative environment, multiple users work at the tabletop together. But often it is critical for the system to know which user produced which input.

When multiple users are working simultaneously at a tabletop, using only their bare hands for direct touch input, it is not easily possible to tell which input belongs to which user. Electronic devices, such as mice, keyboards, and pens can have unique identifiers to be separated from each other. Unless your system uses an incredible high input resolution to detect finger prints - which is unlikely to be realized on a large tabletop surface with the current state of technology - there is no simple way to assign each finger to a person. You could put a small fingerprint scanner in one of the corners of your table, but then you would need overhead cameras to keep track of the fingers after they are lifted from the table. A similar result could also be achieved by identifying users via an electronic device they are carrying, e.g., their cell phone or a dongle.

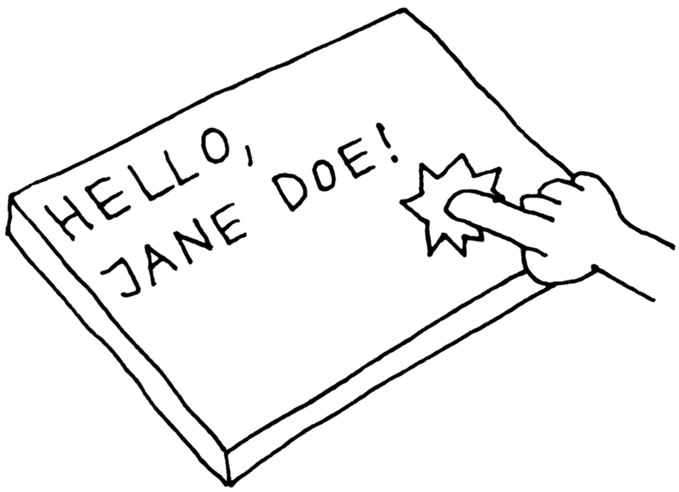
The DiamondTouch [1] is a capacitive tabletop, where an array of antennas is used to transmit small electrical charges from a transmitter beneath the table through the user's finger, further to their body, and to the chair and finally to the receiver. This enables user identification since the system can tell from which chair the input was received. However, obviously the imagination of sitting on an electric chair can produce a very uncomfortable feeling for the users. It is also a difficult and time-consuming procedure to connect all the wires, and the connection is lost as soon as users get up or even change their seats.

Mohamed et al. [2] use gestures from pen input to identify users. Their recognition algorithm is able to tell on which side of the table the user is located, only by pressure and angle information and without the need for overhead camera detection. The downside of this method is obviously that you need a pen, which eliminates the advantage of direct touch. Consider a tabletop game, where user identification is crucial, but you do not necessarily have enough pens and direct touch input is possibly preferred, e.g., in a card game.

Recent research came up with an idea that gets close to the initial fingerprint idea, but does not rely on high resolution: Schmidt et al. [3] implemented a hand shape detection mechanism. With high robustness and an error rate of only 0.5% this approach is a realistic alternative that does not depend on techniques as capacitive touch or pen input like the aforementioned solutions.

Therefore:

Let the table distinct the input from different users. Identify the user either by hardware or via software mechanisms and assign the interface elements to the user that he or she is working with.



With identification of users, (3-4) PRIVATE SPACE and (3-5) BALANCED PARTICIPATION can be supported more easily. Also, personalized (3-2) HAND GESTURES offer a distinct system response for different users with various additional options...

References

[1] P. Dietz and D. Leigh, DiamondTouch: A Multi-User Touch Technology. *Proc. UIST '01*, pp. 219–226.
[2] K.A. Mohamed, S. Haag, J. Peltason, F. Dal-Ri, and T. Ottmann, Disoriented Pen-Gestures for Identifying Users around the Tabletop without Cameras and Motion Sensors. *Proc. TABLETOP '06*, pp. 43–52.
[3] D. Schmidt, M.K. Chong, and H. Gellersen, HandsDown: Hand-contour-based User Identification for Interactive Surfaces. *Proc. NordiCHI '10*, pp. 432–441.

(2-3) Desktop Orientation **

Fig. 9. Windows on a rotary table aligned to the border with the DiamondSpin toolkit. Courtesy by Shen et al. [1].

... multiple users will interact with your (1-2) Large Collaboration Table simultaneously, which might be a (1-3) Round Table that does not favor anybody. As you are developing the software framework, you still keep equilibrium of participation among the several users in mind, knowing that it is not possible for all users to look at the screen from the same direction.

Multiple users interact with multiple desktop widgets from different points of view. Viewing content, especially reading text upside down or even sideways can be hard and makes the user uncomfortable while using the tabletop.

On vertical displays such as traditional desktop computers, there is always a clear direction for all widgets, immediately visible for text being displayed. In a collaborative environment, participants are usually standing or sitting around multi-touch tables, hence everyone is looking on the table from a different angle. For example, imagine a task where four users sort small note widgets with short text passages on it. It is very hard to read upside down or even sideways, thus the user with the most text elements oriented towards her is in advantage. However, the system often does not know which user is looking at which elements on the surface. In addition, aligning everything to one side of the screen as on a traditional PC gives one user an unfair advantage over the others and does neglect the possibilities of technology in today's systems, where dynamic and independent orientation can easily be implemented.

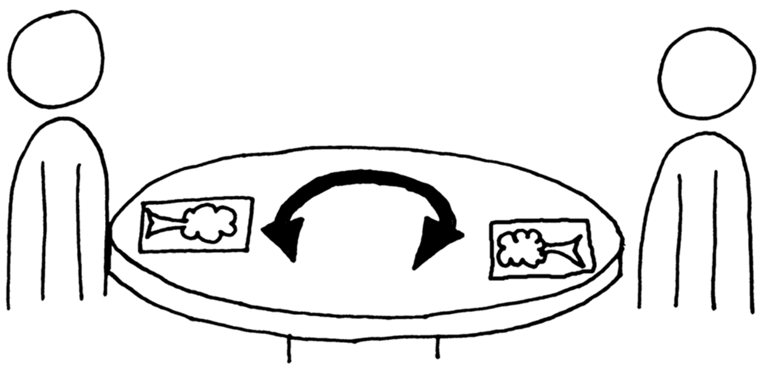
The DiamondSpin toolkit [1] enhances a DiamondTouch table with a circular workspace projection. It offers an easy way to orientate and align desktop elements according to the viewpoint of the users, thus improving the interaction experience of all collaborators. It is also possible to rotate the whole table, e.g., to switch tasks between two users, or orient all windows towards one single user, a technique they call "magnetizing". However, in their implementation there was no independent orientation of widgets, i.e., it is not possible to orient an element towards a side of the table that is farther away than an other element.

Another similar approach was done by Koike et al. [2], focusing on browsing large amounts of data, in their specific example huge picture collections. The pictures are being displayed in a circle, thus aligning all elements automatically to the nearest edge.

Both examples share the same disadvantage: Automatic orientation is not always what users may want. Kruger et al. [3] second this in their extensive user study that investigates the orientation issue. Results clearly suggest that viewpoint correction has to be resolved dependent on the tasks and user's demands. A recent technique which allows manual orientation regardless of the user's position at the table was proposed by Dragicevic and Shi [4]. With the help of (3-2) HAND GESTURES, users can create their own workspace, automatically including and aligning a set of specified documents selected beforehand.

Therefore:

Enable free rotation of interface elements on a basic system level in your tabletop setup. Support automatic orientation if you have knowledge of the user's position at the table, but also include manual orientation functionality.



With interface elements oriented towards the users, it is possible to ensure (3-5) BALANCED PARTICIPATION among the participants, e.g., in a meeting or on an exhibit tabletop. Also think about implementing (3-4) PRIVATE SPACE, inherited by the orientation of screen elements in this area, to let users protect their documents, e.g., private notes or other separated work...

References

[1] C. Shen, F.D. Vernier, C. Forlines, and M. Ringel, DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. *Proc. CHI '04*, pp. 167–174.

[2] H. Koike, S. Kajiwara, K. Fukuchi, and Y. Sato, Information Layout and Interaction on Virtual and Real Rotary Tables. *Proc. TABLETOP '07*, pp. 95–102.

[3] R. Kruger, S. Carpendale, S.D. Scott, and S. Greenberg, Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication. *Computer Supported Cooperative Work (CSCW)*, vol. 13, 2004, pp. 501–537.

[4] P. Dragicevic and Y. Shi, Visualizing and Manipulating Automatic Document Orientation Methods Using Vector Fields. *Ext. Abstr. ITS '09*, pp. 65–68.

(3-1) HIGH PRECISION INPUT *

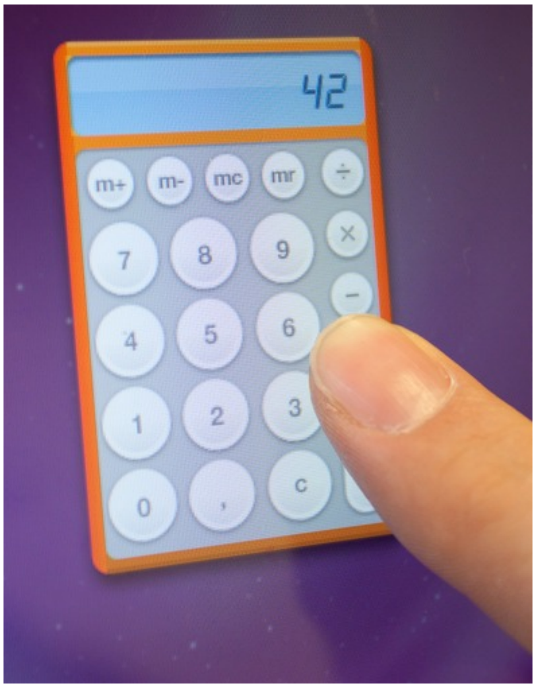


Fig. 10. Example for high precision needed: too small buttons.

...your tabletop is dealing with large amounts of data, and you may have implemented (3-2) HAND GESTURES or a (2-1) ZOOMABLE INTERFACE already to cope with the problems of navigating through the content. Although your interface is designed properly for multi-touch purposes, sometimes users may encounter difficulties when trying to hit buttons or content.

The touch interaction needs to be precise and avoid mistakes, but it is hard to get the exact pixel that was hit by a certain finger.

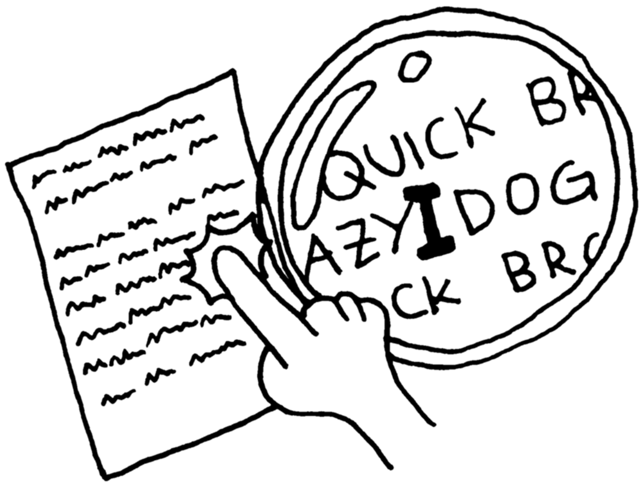
Direct touch and its natural behavior is one of the biggest advantages of multi-touch tabletops. But although the resolution is high and the tracking mechanism is quite exact, it is hard to tell which exact point the user hit, since the shape of a finger covers a larger area of the surface than just one pixel. In tabletop research, this is often referred to as the "fat finger problem". Although to a different extent, it applies to almost every task on a tabletop. In a scenario where you want to point out a location or a route on a geographical map, you want to hit certain locations like roads, towns, or sights as exactly as possible. You can always zoom in and out, but sometimes this is not possible in your task, e.g., when you want to draw a line alongside a long road on a map, you probably do not want to zoom in, because you might not see your target anymore on the surface. A similar problem applies to text input, where you want to hit the small space between two letters, and by zooming in you will lose the context of the surrounding text when it gets off the screen. The Apple iPhone [1] overcomes this issue by displaying a small magnifying glass above the finger, such that the user is able to tell exactly which letter he just hit. A more general example is depicted in the sample illustration above, where a user encountered problems to hit the desktop interface elements that were build for a traditional PC and are now too small to hit on an interactive tabletop.

Benko et al. [2] elaborate on this situation and offer five different techniques to overcome this problem. User studies shows that the best of their solutions is a dual finger stretch technique, where one finger is used as the selector and a second finger is placed nearby the selector, moving away and thereby zooming in to the area, making it easier to aim for the first finger. Four other techniques were presented: two with a cursor speed control (one with a zoom-like gesture, one with a pie-menu) and two solutions which did not perform very well in the user tests, being the dual finger midpoint technique (where the midpoint of two fingers is the cursor) and the offset technique (where the cursor has a slight offset to the real touch input point).

Another more recent paper by Olwal et al. [3] introduces two different (3-2) HAND GESTURES to solve the precision problem, rubbing and tapping, which zoom into the interface. They combined these gestures in several distinct ways, e.g., one finger to zoom in with the rubbing gesture and the other finger doing the selection. In conclusion the tapping gesture as zoom-in with the finger release point being the selection performed well, offering a very short and easy method to increase precision with only one finger. Nevertheless, the user tests also suggest that an additional click to confirm the selection is more robust without sacrificing too much time.

Therefore:

Offer input methods to point to exact coordinates on the tabletop. Use gestures, tangibles, or precision-enhancing algorithms to allow the user for precisely selecting specific points on the surface.



High precision input extends the variety of applications on your tabletop. One other way to achieve this, is a (4-4) PEN INPUT DEVICE...

References

[1] Apple Inc., iPhone, http://www.apple.com/iphone/
[2] H. Benko, A. Wilson, and P. Baudisch, Precise Selection Techniques for Multi-Touch Screens. *Proc. CHI ’06*.
[3] A. Olwal, S. Feiner, and S. Heyman, Rubbing and Tapping for Precise and Rapid Selection on Touch-Screen Displays. *Proc. CHI ’08*, pp. 295–304.

(3-2) HAND GESTURES **

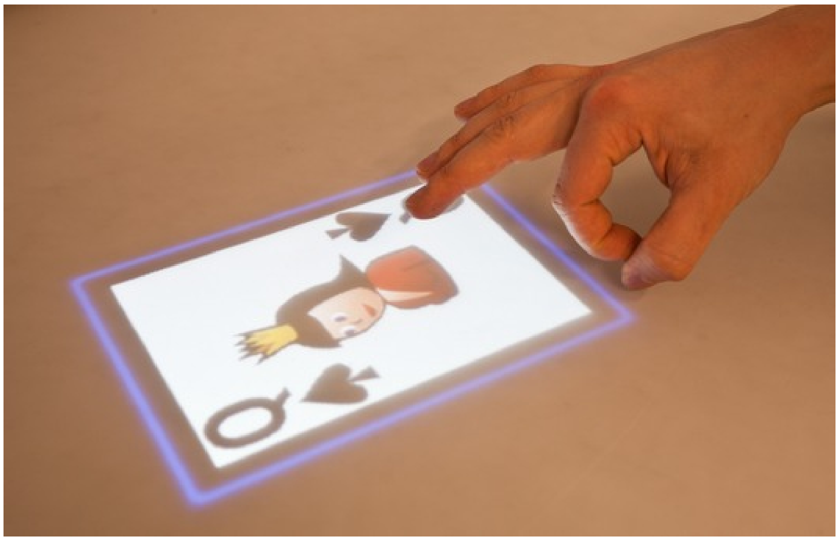


Fig. 11. Pushing away a card on a tabletop with a flicking gesture.

... the tabletop software framework is being developed, including possibly (2-2) USER IDENTIFICATION and a (2-1) ZOOMABLE INTERFACE. You are considering the users' tasks and realize that there is a variety of different commands beyond your limited number of buttons or similar interface elements to issue these commands.

There can only be a certain number of buttons on the screen without overcrowding the surface and making it unusable. But as the applications grow in functionality, so does the set of input commands.

While there are hardly any gestures commonly known on desktop computers, multi-touch devices with direct input afford the development and use of gestures. The most well-known gesture is probably the two-finger zoom gesture, which was not invented, but introduced to a broad public by the Apple iPhone [1]. This zoom gesture is often used on tabletops as well, but since tables are larger devices, users tend to use two hands instead of just two fingers from one hand. This example shows that gestures are extremely useful, they can extend tabletop interaction in innovative ways, and they are easy to learn due to their natural behavior. But there are also some drawbacks when introducing new gestures. Without visual hints, most users will not use them, unless they were told to do so. However, the absence of visibility is one of the advantages about gestures, they offer invisible input for diverse commands without the need for screen real estate.

The former mentioned zoom gestures were also used in a paper by Tse et al. [2], but extended by speech input and even other gestures, application-specific. For example, when browsing on a geographic map, one finger is used to move the map, the complete hand issues a 3D tilt up, whereas a five finger touch results in a 3D tilt down. In the other task, a strategy game, there are actions often necessary like selecting a number of units. While putting two sides of the hand down on the table, the space in between is marked and the units in the area are selected.

Wu et al. [3] elaborate on the development of new gestures with design guidelines for the process of defining gestures. They say it is important to make sure the gesture registration is distinct from any other input done, the recognition should be variable and not too selective to allow errors in reproducing the gesture, and gestures should be short, easy, and partly reused, to ease the learning process for the users. In some user tests they demonstrated their findings with a few gestures they came up with, e.g., a wiping gesture for handwriting. Similar gestures are supported by the SMART Board [4], where the size of the touch point determines whether to draw (pen-sized touch), move (finger-sized touch), or erase (palm-sized touch).

To get an overview of useful gestures that have been invented so far, take a look at the work by Wobbrock et al. [5], who created a taxonomy for gestures. They also conducted a user study with user-defined gestures, where in all 1080 gestures were evaluated for 27 different commands, leading to interesting results. When choosing gestures for your application, consider these important aspects: gestures should offer a natural mapping to real-world interaction, or even better if gestures have been used before in other applications for issuing the same command. If the gesture is a newly invented one, provide small hints without being annoying; however, if too many hints are necessary, the gesture is clearly unintuitive and should be replaced by another. Most important, gestures should be consistent among different systems. By adopting established gestures for your tasks, you can reduce the learning effort, enhance usability, and contribute to build up new standards.

Therefore:

Extend the number of input commands by implementing gesture detection. The gestures should be robust, unambiguous, and easy to learn, or even better, already be commonly used, like the two-finger zoom. Do not overuse gestures though, and offer novice users hints for unfamiliar gestures, e.g., when using the application for the first time.

Gestures can be used for (5-4) EXTENDING REACHABILITY or invoking (3-4) PRIVATE SPACE. There are also examples of gestures to offer (3-1) HIGH PRECISION INPUT...

References

[1] Apple Inc., iPhone, http://www.apple.com/iphone/
[2] E. Tse, C. Shen, S. Greenberg, and C. Forlines, Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop. *Proc. AVI '06*, pp. 336–343.
[3] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan, Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. *Proc. TABLETOP '06*, pp. 185–192.
[4] Smart Technologies, SMARTBoard,
http://www.smarttech.com/
[5] J.O. Wobbrock, M.R. Morris, and A.D. Wilson, User-Defined Gestures for Surface Computing. *Proc. CHI '09*, pp. 1083–1092.

(3-3) REPLACE PHYSICAL PAPERWORK *

Fig. 12. The Apple iPad [3] may become one step towards a future where physical paper is obsolete.

... you are building a tabletop that will be used primarily in office scenarios like team meetings or as an extension for everyday's paperwork tasks. Its focus is mainly actively reading text, which involves editing or taking notes.

Paper offers many advantages, and many users prefer books over digital content. However, books and paper sheets occlude the tabletop surface, which limits the available interaction area, and for many tasks digital files are superior to physical paper.

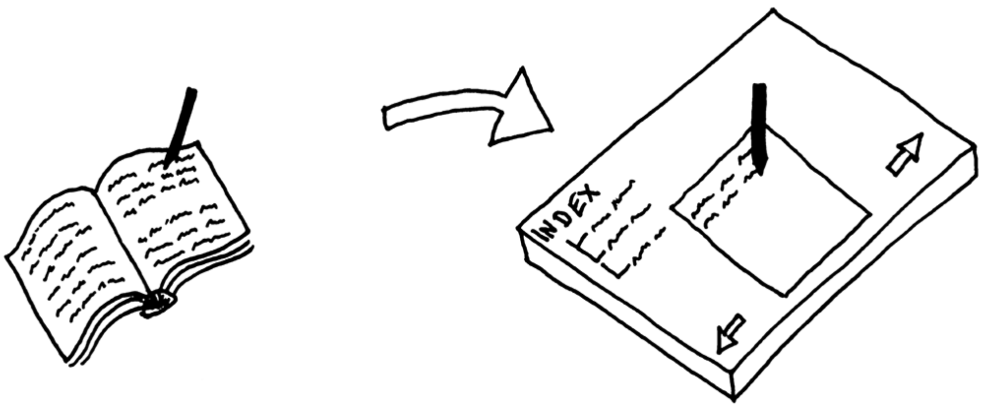
There are many arguments for keeping physical paperwork, but ebooks and other digital texts are becoming more and more popular. We are receiving a lot of information in a digital form, e.g., scientific papers, social networks, or online news services. However, the current implementations to read text on desktop computers are still not user-friendly enough to convince users to throw away their books. On the other hand, on tabletops you do not have a large area for books and sheets as on your desk for your desktop computer.

The goal should be not just to replace paper, but to make use of the advantages of tabletops over paper and desktop computers [1]. For example, when searching a particular text passage, you flip the pages of a book or look into the index. On a desktop computer, you either look the word up in the index, too, or use the search function. On a tabletop, you can use intuitive gestures to mimic the flipping pages functionality, as well as have the search and index functions from desktop computers. Especially making annotations and highlighting text is more natural with direct input than relative input like the mouse. You can even add a pen to your tabletop, to make it work the exact same way like annotating a physical book. The advantage then again is that you can search later on for all the highlighted text passages, which is not easily possible in a book unless you mark the pages in another way, e.g., sticky notes. As also noted by Sellen and Harper [2], it is important to change the work practice before users replace paper with digital reading material. For example, support the extraction of small text parts or creation of notes that can be arranged in a mind map or a list.

With the Apple iPad [3], the press and many publishers hope to bridge the gap between physical paper and digital text. With its low weight and size, it aims to mimic the look and feel of a book, while offering many of the advantages of today's technology. Desktop computers were not able to replace paper; one reason for that might be that collaboration is limited to remote, due to the single-user centered system and spatial arrangement. Tabletops however support multiple users that can work simultaneously on multiple files. Additionally, sharing files is even easier than on the iPad, which you have to hand over to someone like a book or connect to another device for sending data. On a tabletop, you can just flick the document to the other user with a touch, and edit it collaboratively on the huge work space, as Masoodian et al. [4] proposed: They modified a desktop text processing application to allow for collaborative editing, where multiple users could simultaneously work on the same document using personalized (3-2) HAND GESTURES.

Therefore:

Implement paperwork applications that support tasks by mapping real-world interactions to digital text, such as flipping pages, marking pages, and visible feedback of the available pages left over. Make use of the advantages of digital text, e.g., collaborative editing, quick sharing, undo, fast index search, and dynamic layouts.



If you expect a lot of text input, you may put a (4-3) PHYSICAL KEYBOARD on the table. With a (4-4) PEN INPUT DEVICE, you can allow users to make handwritten annotations...

References

[1] L. Terrenghi, D. Kirk, A. Sellen, and S. Izadi, Affordances for Manipulation of Physical Versus Digital Media on Interactive Surfaces. *Proc. CHI '07*, pp. 1157–1166.
[2] A.J. Sellen and R.H.R. Harper, The Myth of the Paperless Office, MIT Press, 2003.
[3] Apple Inc., iPad, http://www.apple.com/ipad/
[4] M. Masoodian, S. McKoy, and B. Rogers, Hands-on Sharing: Collaborative Document Manipulation on a Tabletop Display Using Bare Hands. *Proc. CHINZ '07*, pp. 25–31.

(3-4) PRIVATE SPACE *

Fig. 13. Children at a tabletop reaching into the private space of each other. Courtesy by Zhang and Takatsuka [1].

... you have multiple users working on their own private data, e.g., personal documents or in a game. With techniques such as (2-3) DESKTOP ORIENTATION or (2-2) USER IDENTIFICATION the users' data is personalized, and (3-2) HAND GESTURES extend the input variety on your tabletop, e.g., a (1-2) LARGE COLLABORATION TABLE, but you are still facing privacy issues while sharing data among users.

Users want to use the whole table, but sometimes privacy becomes an issue when users want to interact with displayed elements without sharing them with other users.

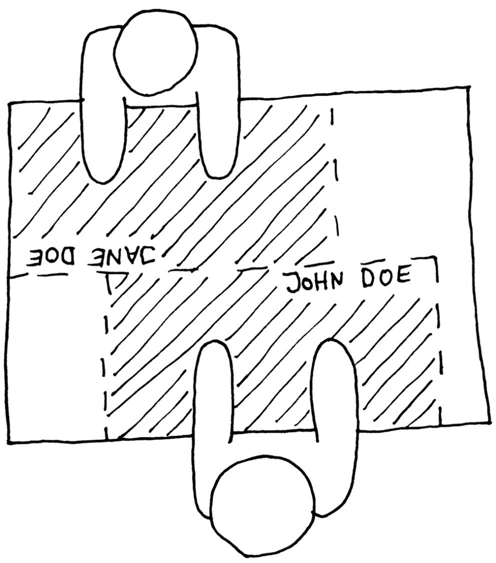
In collaborative workspaces users have personalized data, e.g., their folder of documents, cards in a game, or piles of photos. When multiple users bring their data to the same tabletop where they are working collaboratively, they do this for a reason. They want to share documents, work together on a task or just play a game. But while all this data is available to all the users, it raises the problem of undesired access in specific situations. The situation also avoids undesired access, e.g., that someone steals a document from someone else. Consider a scenario where two users are sorting pictures, one user applies a zoom gesture to enlarge a picture to look at some details, and accidentally the whole screen is covered by that single picture, hindering the others from interacting with the tabletop.

Tse et al. [2] explored the interaction interferences of co-located collaborators in everyday's work tasks. They proposed a set of informal design guidelines for application designers to account for territoriality in multi-user interaction. If the tasks are semantically separated, they should be spatially separated from each other as well. Pop-up windows and other widgets should not be centered on the screen like on usual desktop systems, but stay inside the input area of the user who is working on that particular task. They also clearly advice to create a private space as well as a group space to distinguish between the different modes of tasks. The private space should be close to the physical position of the user, while users should be able to move their documents around during their work.

A study of territoriality based on a trading-card game experience by Pinelle et al. [3] revealed that automatic protection mechanisms are preferred by the users. They implemented a user control level that changes accordingly to the distance of the participants: If the user's input focus is farther away from their home area, the control level is lower and other users can steal items from the users private space as long as their control level is higher. If a user is working in her private space, the control level is at maximum so that no other user can steal or even distract the user during her work. In this particular application, the automated mechanism was preferred by the users and also showed better results in the quantitative study than the user-controlled mechanism. However, depending on the specific task of your application, a combined privacy implementation of both automatic and user-controlled may be more suitable.

Therefore:

Offer an area of private space that allows users to protect their data against undesired access. This can be implicitly or explicitly implemented, by having a lock/unlock button or a designated area that does not allow interaction from other users than the documents' owner.



With users having control over data, make sure that there is a (3-5) BALANCED PARTICIPATION among the collaborators and no one can block others from interacting with the tabletop, controlling the discussion...

References

[1] X. Zhang and M. Takatsuka, Put That There NOW: Group Dynamics of Tabletop Interaction under Time Pressure. *Proc. TABLETOP '07*, pp. 37–43.
[2] E. Tse, J. Histon, S.D. Scott, and S. Greenberg, Avoiding Interference: How People Use Spatial Separation and Partitioning in SDG Workspaces. *Proc. CSCW '04*, pp. 252–261.
[3] D. Pinelle, M. Barjawi, M. Nacenta, and R. Mandryk, An Evaluation of Coordination Techniques for Protecting Objects and Territories in Tabletop Groupware. *Proc. CHI '09*, pp. 2129–2138.

(3-5) Balanced Participation *

Fig. 14. The SIDES game is a turn-based game for children on a tabletop. After each turn, the user presses a 'turn taking' button to give the turn to the next player. Courtesy by Piper et al. [1].

... you designed a tabletop for collaborative tasks like a game or an exhibit, where users can interact not only with the tabletop, but also with other participants. The input can be distinguished by (2-2) User Identification and users have their own (3-4) Private Space, with elements aligned towards them using (2-3) Desktop Orientation.

In a collaborative environment, multiple users should contribute to the interaction. Occasionally, some users take the lead and hinder other, more calm users take part in the discussion, thus possibly creating unbalanced results.

There are many different situations for multiple users in a collaborative workspace: working simultaneously or in distinct turns, interacting on the same element or completely different desktop widgets, contributing to the same task or doing another project on the same table. In most situations, each collaborator should get roughly the same amount of interaction time at the tabletop. Imagine a brainstorming session or a discussion about new design ideas. Usually, some people tend to take the lead in these situations, no matter whether unintentionally or on purpose. Other users stay in the background, they speak up only when being asked or if there is a break in a conversation, which does not happen very often, e.g., in brainstorming sessions.

An insightful experiment was conducted in Helsinki, where a large multi-touch wall was installed in a central city location [2]. Several observations were made how distinct multi-user interaction can be, and we can outline how the setup of the multi-touch display can influence the turn taking. For example, although the surface was large enough that several users could interact simultaneously, users lined up at some point and started interacting one after each other. In another situation, one user used the zoom gesture to enlarge a photo to the whole screen.

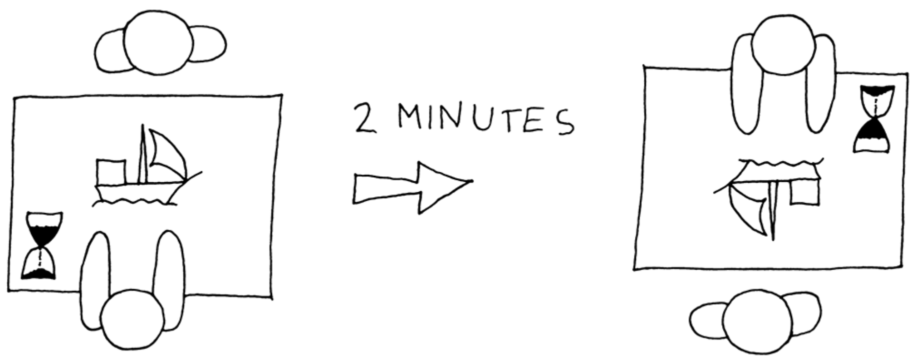
Another user, who was exploring the wall at the same time, was distracted by this photo on his side of the wall, immediately stopped the interaction, and left the scene. Both examples outline how important it is to make use of the whole table as a space to invite users. In the first example, some invitation screen could have encouraged users to start interacting simultaneously, while in the second example the zoom gesture should not be able to overlap other interaction areas that are active at this time.

Piper et al. [1] implemented an interactive tabletop game to improve social skills for adolescents suffering Asperger's Syndrome. Four students had to build a path of lily pads to help a virtual frog to cross a sea by collaboratively lining up the pads in turn. They discovered that explicit "turn-taking" buttons were necessary to balance participation among the students, since a voting system in an earlier prototype lead to discussions dominated by more active children.

Marshall et al. [3] evaluated how mouse versus finger input and single-touch versus multi-touch influenced the equity of participation. Their findings indicate that touch interfaces and especially multi-touch interfaces lead to more equity in interaction. However, the verbal participation showed no significant change, although the subjective perception of the users contradicted to these points, as they perceived more equity. A possible explanation might be that even though the amount of verbal participation does not increase quite much, the contribution of more silent collaborators can be huge. Nevertheless, the study also suggests that a larger table might lead to different results, hence make sure to choose a suitable table size for your tabletop setup.

Therefore:

Set up the table such that every collaborator can contribute in the same way to the task. Your software should not allow anyone to occupy the table and hinder interaction by the others. If applicable to your task, try to balance the participation over all users.



This is a basic pattern with no further references within this language.

References

[1] A. M. Piper, E. O'Brien, M.R. Morris, and T. Winograd, SIDES: a cooperative tabletop computer game for social skills development. *Proc. CSCW '06*, pp. 1–10.
[2] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko, It's Mine, Don't Touch!: Interactions at a Large Multi-Touch Display in a City Centre. *Proc. CHI '08*, pp. 1285–1294.
[3] P. Marshall, E. Hornecker, R. Morris, S.N. Dalton, and Y. Rogers, When the Fingers do the Talking: A Study of Group Participation with Varying Constraints to a Tabletop Interface. *Proc. TABLETOP '08*, pp. 33–40.

(4-1) EMBEDDING ELECTRONIC DEVICES *

Fig. 15. PhoneTouch connects a tabletop with a cell phone. Courtesy by Schmidt et al. [3].

... your tabletop is built and you have implemented a basic software framework. But you realize that users need their data while working on the interactive tabletop, or you even want the users to use their data, e.g., in a meeting at a (1-2) LARGE COLLABORATION TABLE, in an exhibit, or if your (1-1) TILTED TABLE is used as an information device in public.

Tabletops offer large areas to work on personal data or share it with others, e.g., to work collaboratively on it, but those files are often on electronic devices such as cell phones or notebooks.

We carry a lot of data with us, like documents on our laptop, pictures, or contact data on our cell phone, some more data on a pen drive or other devices. When you work with a tabletop, you deposit these devices somewhere; maybe in a (5-3) PHYSICAL OBJECT STORAGE BIN. Then you grasp a keyboard to type some text, or a digital pen to paint something, or some tangibles for other special input. But one might ask: why not combine all these actions to enhance your tabletop experience? Instead of putting your laptop or your cell phone away, place it on the tabletop and use it as input device as well as use the data on it. Former devices which were recognized as obstacles and hindered proper interaction, should rather be recognized as tools in order to improve the user's experience.

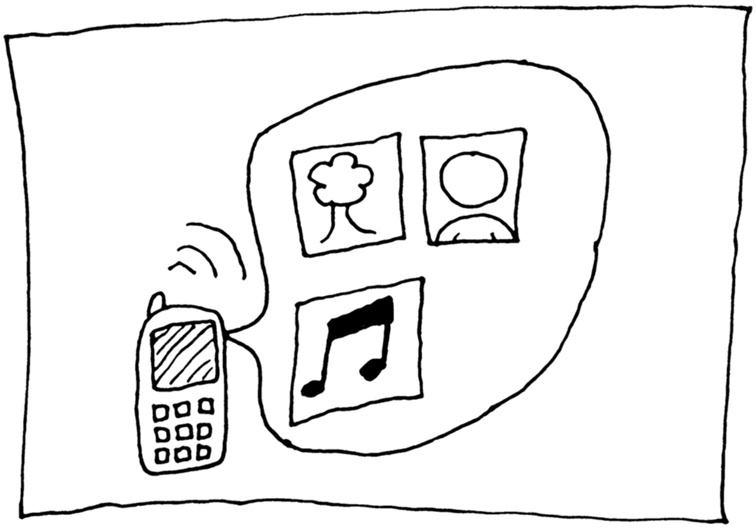
There are numerous examples for this, e.g., with your laptop you could create a small synchronized text input area on a (1-2) LARGE COLLABORATION TABLE to take notes, or an information screen that tells the user the way to a sight and store the direction on her cell phone. Other scenarios could be, e.g., a tabletop game where participants can save their game on a pen drive, or a meeting where workers bring in sales figures from different company departments to compare them on the tabletop and collaboratively create one single chart for a presentation.

Rekimoto et al. [1] suggest to connect your electronic devices to the tabletop environment to create a "spatially continuous work space". The laptop keyboard can be used as text input device, documents are shared, and the laptop display compensates for the occluded part of the tabletop surface. The devices are recognized via a camera mounted on top of the surface that searches for unique pattern markers attached. An application example shows a large area map on the tabletop and upon pointing to sights they are displayed on the laptop LCD screen.

Eight years later, connecting these devices has become incredibly easy, since almost every portable device supports WiFi and Bluetooth. Based upon the latter, Wilson and Sarin proposed BlueTable [2], a tabletop system which uses object detection via a camera on top of the surface just like Rekimoto, but now detection by shape was used and without the need to attach markers to it. When a known device like a cell phone is recognized, a connection is automatically established via Bluetooth handshaking protocol. If it is successful, recent files on the cell phone like pictures are automatically displayed around the phone on the surface. Schmidt et al. [3] extend on this idea by supporting not only file transfer, but also interaction with the tabletop via the phone.

Therefore:

Ease the transfer of data between the tabletop and other electronic devices. Support seamless connectivity to share data and expand the tabletop's input variety.



To avoid that these devices occlude important information on the screen, techniques such as (5-1) DODGE OBSTACLES are helpful...

References

[1] J. Rekimoto and M. Saitoh, Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. *Proc. CHI '99*, pp. 378–385.

[2] A.D. Wilson and R. Sarin, BlueTable: Connecting Wireless Mobile Devices on Interactive Surfaces Using Vision-Based Handshaking. *Proc. GI '07*, pp. 119–125.

[3] D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen, PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. *Proc. UIST '10*, pp. 13–16.

(4-2) On-Screen Keyboard *

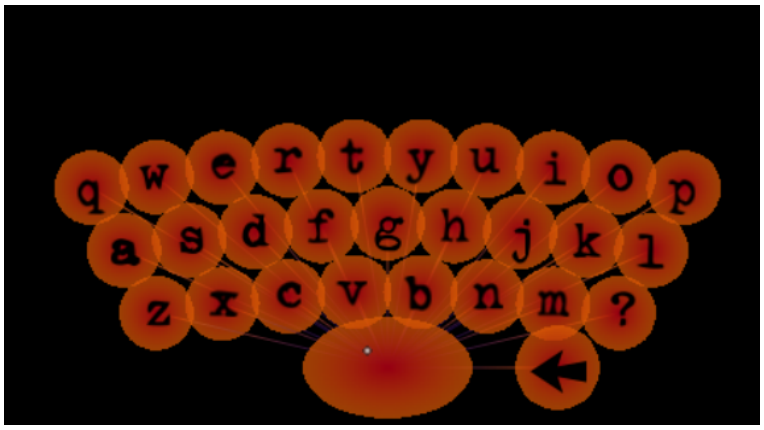


Fig. 16. BubbleType on-screen keyboard for text input on a tabletop. Courtesy by Hinrichs et al. [3].

... your tabletop is used in scenarios like interactive games, at exhibits, or in meetings where it serves more as a presentation table than for a lot of paperwork, and thus only needs limited amount of text input. In these occasions your screen space becomes limited, with cups, notes, and electronic devices like cell phones on the table, or it is even a (1-1) Tilted Table, where you cannot put objects on.

A physical keyboard may offer a familiar text input environment, but it clutters the space with its occlusive appearance. On a very limited amount of available screen space, you have to enable text input sometimes in order to store textual information.

Text entry is necessary in almost every imaginary scenario for an interactive tabletop. In exhibits you may want to allow comments, in meetings participants want to take notes, and in a game you want to put your name in before you start playing. But as Hinrichs et al. [1] elaborated, there are situations in which the realization of text input becomes a difficult problem. For example, if your space is limited, as on a rather small device, or in the case you have a (1-1) Tilted Table or even a vertical wall instead of a horizontal tabletop surface, you cannot place input objects on the surface. Also, if your task demands only for small amount of input such as short annotations, it is a tedious waste of time to get a (4-3) Physical Keyboard, typing in your text, and then putting it away again. Sometimes even a keyboard is not necessary, you may even allow hand-written annotations, which can easily be extended by drawings to illustrate the note.

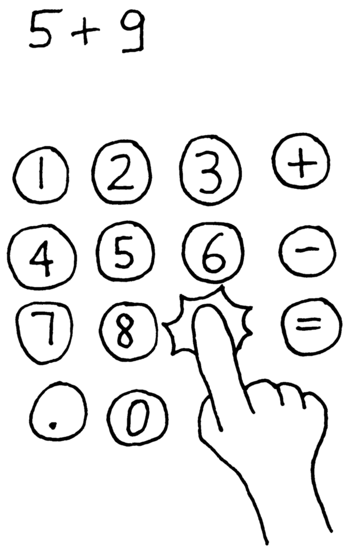
These interfaces do not need to mimic the layout of a physical keyboard, instead they should make proper use of the dynamic behavior by displaying only necessary input elements. For example, if the input focus is on a numeric-only text input area, such as a calculator application. Therefore, a small number keypad is sufficient. On the other hand, those number keypads as well as the function keys are rarely used in a pure text-writing environment and can be hidden. Other examples are, e.g., graphical software where only hotkeys are needed, applications that limit input to function keys, or only copy, paste, edit keys in a photo viewing program.

Another particular aspect of on-screen keyboards is the ability to try completely different new layouts, even those that are not easily realizable on physical devices. User tests that compare these new layouts with traditional ones could lead to completely new input devices. Hirche et al. [2] introduced a keyboard with 14 keys (one for each finger except the little fingers and forefingers, which has two keys), where a small shift downwards or upwards for each finger hit another letter. The order of letters on the keys was arranged according to the frequency of use in the English language.

However, keep in mind that these layouts are only useful if users are faster than on the default QWERTY layout. Changing this is not recommended in scenarios such as exhibits or for other tabletop use cases where users expect familiar keyboards. Hinrichs et al. [3] proposed such different layouts, e.g., keys in a circle around the hand, or a single line of keys. But BubbleType, the final suggestion of their paper, is again a traditional QWERTY keyboard layout. It makes use of the tabletop display by offering a prediction system: After typing a specific letter, the next most likely letter is highlighted to decrease the possibility of a typing error.

Therefore:

Implement an on-screen keyboard that uses the advantages of the tabletop's direct touch nature. The keys should be easy to hit, offer visual and auditory feedback to ease the interaction. If the input is limited to numbers or some other subset of keys, display only these to save space.



An on-screen keyboard offers superior, previously unknown possibilities to the user's working experience, such as (5-2) DYNAMIC KEYBOARD RELABELING, which makes excellent use of the tabletop display for your tasks...

References

[1] U. Hinrichs, M. Hancock, C. Collins, and S. Carpendale, Examination of Text-Entry Methods for Tabletop Displays. *Proc. TABLETOP '07*, pp. 105–112.
[2] J. Hirche, P. Bomark, M. Bauer, and P. Solyga, Adaptive Interface for Text Input on Large-Scale Interactive Surfaces. *Ext. Abstr. TABLETOP '08*, pp. 153–156.
[3] U. Hinrichs, H. Schmidt, T. Isenberga, M.S. Hancock, and S. Carpendale, BubbleType: Enabling Text Entry within a Walk-Up Tabletop Installation. Report 2008-893-06, Department of Computer Science, University of Calgary, 2008.

(4-3) PHYSICAL KEYBOARD *

Fig. 17. Wireless keyboards on a tabletop.

... you have built a (1-2) LARGE COLLABORATION TABLE for an office environment, e.g., to (3-3) REPLACE PHYSICAL PAPERWORK in an administration, or as a meeting tabletop device. You expect a lot of text input and you are looking for solutions such that users will not lose time taking notes compared to someone using a laptop or a sheet of paper.

Users often need text input on a tabletop. However, a displayed virtual keyboard on the tabletop lacks tactile feedback of the keys.

Text input is necessary in many different use cases. On an interactive tabletop in an exhibit, you probably do not need text, maybe only for putting in the user's name or a comment; an (4-2) ON-SCREEN KEYBOARD is sufficient for this scenario. However, for large text passages, e.g., in a meeting where one person is putting brainstorming notes into a mind map, or in administration office tasks with a lot of paperwork, such a virtual keyboard has many downsides. It lacks haptic feedback, i.e., the user does not feel where she is typing and whether she hits the center of a certain key or in between two keys, thus producing more typing errors.

Hinrichs et al. [1] examined different approaches of text input and identified their benefits and drawbacks. A physical keyboard is one of the fastest input devices for text, it is familiar from the traditional desktop setup, and novice-user friendly. However, due to its size it reduces the visible screen space on the tabletop. Additionally, the transition between an external input device and the touch surface can interrupt the user's task focus. Nevertheless, depending on the setup the authors rate a physical keyboard as best input method for most systems.

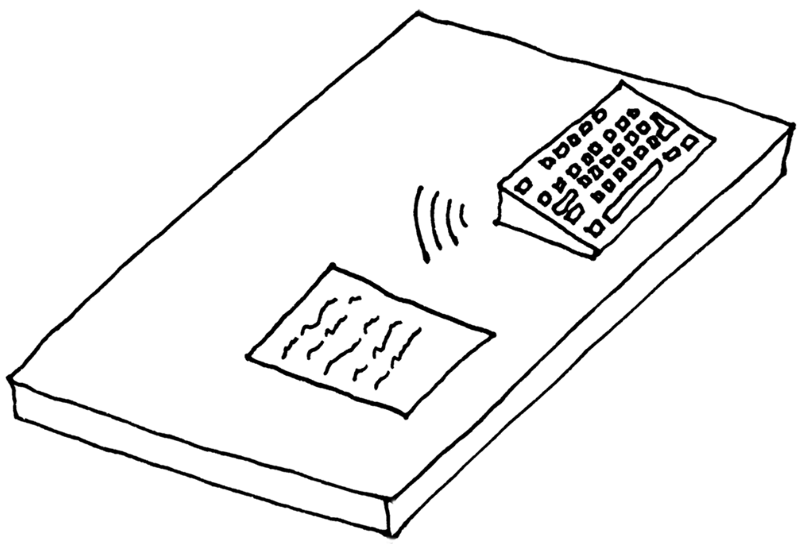
Summarizing their tabletop research experience and drawing conclusions from their observations, Ryall et al. [2] also suggest that a wireless keyboard is the best input solution for text. They point out that typical tabletop tasks should minimize the need for text input and focus on other, more tabletop-suited tasks.

In a more recent paper, Hartmann et al. [3] used wireless keyboards on a tabletop. To overcome the occlusion problem from the keyboard-occupied space, the tabletop was larger than other usual tabletop systems. This introduced another common tabletop problem: users could not reach the whole table. They added wireless mice to solve this problem, additionally to the keyboards. Although this seems to degenerate the tabletop to a simple horizontal output display, since mouse and keyboard input seems to make the direct touch input obsolete, informal user tests showed that users still switched between physical device input and direct touch input. This once more emphasizes the observation that physical input devices do not replace, but supplement touch input.

While we propose in this pattern to offer a keyboard for text input, the use of mice should be avoided by choosing the right table size. If it is too small to offer an additional keyboard without sacrificing too much space, other solutions such as (4-5) INPUT TANGIBLES may be more appropriate.

Therefore:

Put a wireless keyboard on the tabletop to allow the user a familiar way of text input with haptic feedback. The keyboard should be small compared to the table size and easy to move around to minimize occlusion issues.



Buy a special keyboard, e.g., with small displays or transparent keys, or modify your available keyboard to support (5-2) DYNAMIC KEYBOARD RELABELING. After finishing your tasks, the keyboard wastes a lot of space on the screen, so put it in a (5-3) PHYSICAL OBJECT STORAGE BIN...

References

[1] U. Hinrichs, M. Hancock, S. Carpendale, and C. Collins, Examination of Text-Entry Methods for Tabletop Displays. *Proc. TABLETOP '07*, pp. 105–112.
[2] K. Ryall, C. Forlines, C. Shen, and K. and Everitt, Experiences with and Observations of Direct-Touch Tabletops. *Proc. TABLETOP '06*, pp. 89–96.
[3] B. Hartmann, M.R. Morris, H. Benko, and A.D. Wilson, Augmenting Interactive Tables with Mice and Keyboards. *Ext. Abstr. UIST '09*, pp. 149–152.

(4-4) PEN INPUT DEVICE **

Fig. 18. A digital pen creates digital hand-written input.

... you have built a tabletop for educational purposes or office tasks, e.g., for presentations or to design discussions in a team, or to (3-3) REPLACE PHYSICAL PAPERWORK with the use of (3-1) HIGH PRECISION INPUT. Now you think about further input devices you can add to your setup for these specific tasks.

Tabletops offer natural input with direct touch, but for some tasks more precision or multi-functional input devices with different modes and states would be useful.

A digital pen combines two advantages of tabletop interaction: while it mimics the usual input of drawing or writing with a pen on a piece of paper, it also carries the option to embed micro-technology for further interaction possibilities. A well-known pen for tabletops is the *ANOTO* [1] pen. A small camera inside the pen identifies the position on the surface, which is covered by a so called *ANOTO pattern*, a paper with small printed dots on it that store unique location information. The pen's camera tracks these positions while the user is writing and sends the position of information via Bluetooth to the tabletop system.

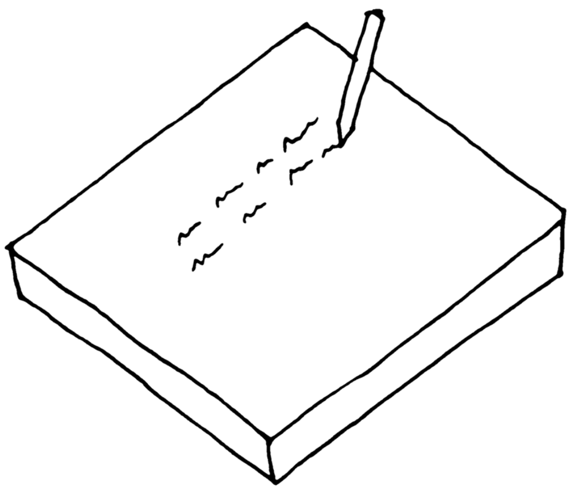
The uPen by Bi et al. [2] aims are combining a laser pointer with a pen. A camera detects the laser spot to get the current pointing position, thus the pen does not even have to touch the surface. It supports simultaneous multi-user input, (2-2) USER IDENTIFICATION, and mouse emulation by two buttons mounted on it, mapped to right-click and left-click. Ortholumen [3] uses a similar technique, although it uses light from LEDs instead of a laser and a polarization filter on the tracking camera for higher sensitivity and precision. The camera, in their implementation located beneath the table, detects not only the position pointed on the surface, but also its shape and size. From that information it calculates the angle and height in which the device is pointing. And as a result, adding more degrees of freedom to the input.

Pen input can be used in many other ways to extend the input. Mohamed et al. [4] combined pen input with gestures to detect on which edge of the table a user is located. Their algorithm needs no camera, it can track the user just by pressure and angle of the pen, with vanishingly little error rate in their user tests.

Pen input does not only replace the input, additionally it extends touch input. In an extensive user study, Brandl et al. [5] discovered that bi-manual input with both direct touch and pen simultaneously is superior to bi-manual direct touch input in some situations. The user performed different tasks, one particular example was a drawing application with a drawing canvas on the right and a huge settings and toolbox pane on the left. Selecting menus and creating a free-hand drawing are two distinct semantic tasks. These are easier to separate for our brain if they are executed in a more distinguished way, such as one by direct touch and the other with a digital pen. For different application tasks they used three setups, one with two pens, one with bi-manual touch, and one with touch and pen combined. The results suggest that "speed, accuracy, and user preference" of the pen and touch setup were superior to the others.

Therefore:

Support a digital pen as an additional input device. The pen should at least be easily trackable by your system without any errors, and it can be enhanced with special functionality to enrich the input variety, e.g., for drawing applications or text input.



With novel interaction techniques, the pen can be used for many different tasks, such as (5-4) EXTENDING REACHABILITY. When the pen is not used, it can be placed in a (5-3) PHYSICAL OBJECT STORAGE BIN so it does not get lost...

References

[1] Anoto, http://www.anoto.com/
[2] X. Bi, Y. Shi, and X. Chen, uPen: A Smart Pen-liked Device for Facilitating Interaction on Large Displays. *Proc. TABLETOP '06*, pp. 160–168.
[3] T. Piazza and M. Fjeld, Ortholumen: Using Light for Direct Tabletop Input. *Ext. Abstr. TABLETOP '07*, pp. 193–196.
[4] K.A. Mohamed, S. Haag, J. Peltason, F. Dal-Ri, and T. Ottmann, Disoriented Pen-Gestures for Identifying Users around the Tabletop without Cameras and Motion Sensors. *Proc. TABLETOP '06*, pp. 43–52.
[5] P. Brandl, C. Forlines, D. Wigdor, M. Haller, and C. Shen, Combining and Measuring the Benefits of Bimanual Pen and Direct-Touch Interaction on Horizontal Interfaces. *Proc. AVI '08*, pp. 154–161.

(4-5) Input Tangibles **

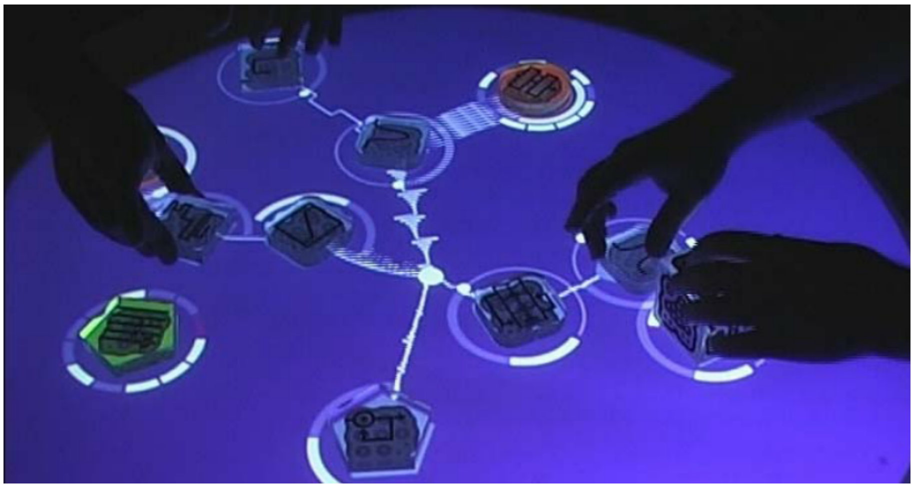


Fig. 19. The Reacable Live! [5] tabletop with tangibles. Courtesy by Nacho Alegre [6].

... if you build a tabletop for environments like exhibits or for gaming purposes, you probably want novel and unfamiliar input devices. Also, if your tabletop is used in a learning environment, traditional desktop PC input devices might not suffice your demands and the user can benefit from additional devices, e.g., to allow simultaneous input from multiple users.

The direct touch input of a tabletop does not always suffice the needs for intuitive interaction and tactile feedback. External devices like keyboards draw the user's attention away from the tabletop and hinder fluent interaction.

Ishii [1] illuminates the advantages of tangible user interfaces (TUIs), and Fishkin [2] proposed a taxonomy of TUIs. There are many different kinds of TUIs, but they all share the same solutions to common problems, such as the lack of tactile feedback or the limited input due to the flat tabletop surface. While they add tactile feedback, they do not separate visually from the tabletop since they are directly mounted on the surface. Additionally, they allow direct control of the tangible objects' virtual representation by manipulating a physical widget.

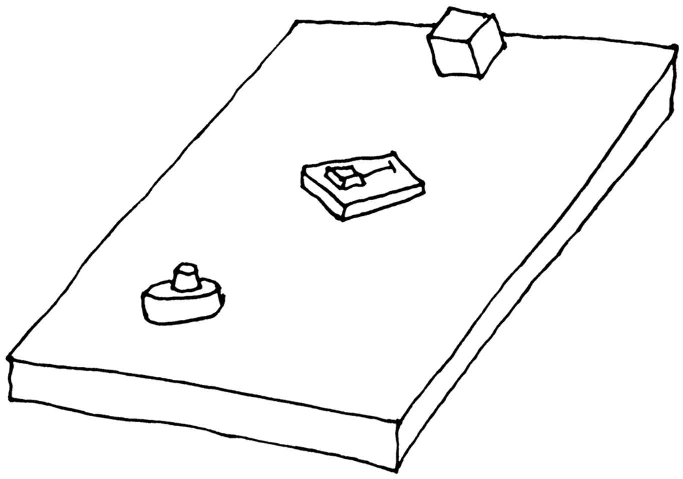
An example of the possibilities of tangibles is Photohelix [3], a knob that can be twisted to scroll through a circular arrangement of pictures. The system design supports bimanual photo editing with the Photohelix, such as holding a knob in the left hand and an ANOTO pen in the right hand. Specific gestures for particular viewing and editing options facilitate handling of large collections.

SLAP widgets [4] combine the advantages of the tangibles' idea with the tabletop output projected on the screen. The transparent objects allow input metaphors such as rotating a knob mounted on the screen, while the table is still visible beneath it and thus the interaction result is immediately visible without drawing off the user's attention from the screen. This also adds new possibilities for text input. The SLAP keyboard offers a customizable keyboard layout: when pressing the shift or the control key, the keyboard layout changes to reflect the projected key or action when pressed.

Another example is the reacTable [5], where different tangible widgets represent different syntheziser instruments on the table. By moving, rotating, and turning those objects the user can change aspects like volume, tone, pitch, and various other settings. Multiple tangibles can be arranged relatively to each other to generate synergy effects. During interaction the user gets continuous and immediate feedback, both visual and acoustic.

Therefore:

Offer tangibles to enhance interaction and increase input variety. Keep the tangibles unobtrusive and provide clear and intuitive transitions towards traditional direct touch tabletop input so that users do not feel distracted by the additional devices.



Tangibles can have displays or just be transparent to support (5-2) DYNAMIC KEYBOARD RELABELING. Especially when you have a large number of tangibles, a (5-3) PHYSICAL OBJECT STORAGE BIN can keep the tabletop tidy and your objects do not get lost...

References

[1] H. Ishii, Tangible Bits: Beyond Pixels. *Proc. TEI '08*, pp. xv-xxv.
[2] K.P. Fishkin, A Taxonomy for and Analysis of Tangible Interfaces. *Personal Ubiquitous Computing*, vol. 8, 2004, pp. 347–358.
[3] O. Hilliges, D. Baur, and A. Butz, Photohelix: Browsing, Sorting and Sharing Digital Photo Collections. *Proc. TABLETOP '07*, pp. 87–94.
[4] M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. Hollan, and J. Borchers, SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. *Proc. CHI '09*, pp. 481–490.
[5] Reactable Live!, http://www.reactable.com/
[6] Nacho Alegre, http://www.nachoalegre.com/

(5-1) DODGE OBSTACLES *

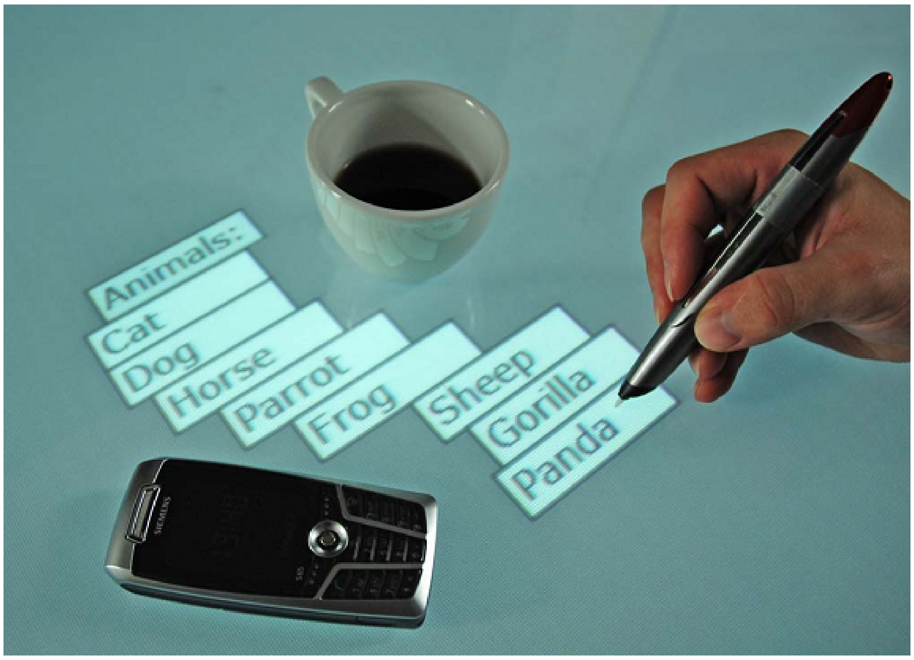


Fig. 20. User-drawn path menus alongside a coffee cup on a tabletop. Courtesy by Leithinger and Haller [2].

... your tabletop system is used in everyday working areas like meetings, where users occasionally put certain devices on the tabletop. (4-1) EMBEDDING ELECTRONIC DEVICES can offer new input as well as data connectivity, if these devices are capable of being combined with your system. But you also have to deal with objects like keys, cups, notes, or books, which occlude valuable screen space.

The tabletop size is well-adjusted to the content displayed, so screen space is valuable and should not be wasted. But users carry everyday objects and may put them on the table during work, either unintentionally or on purpose.

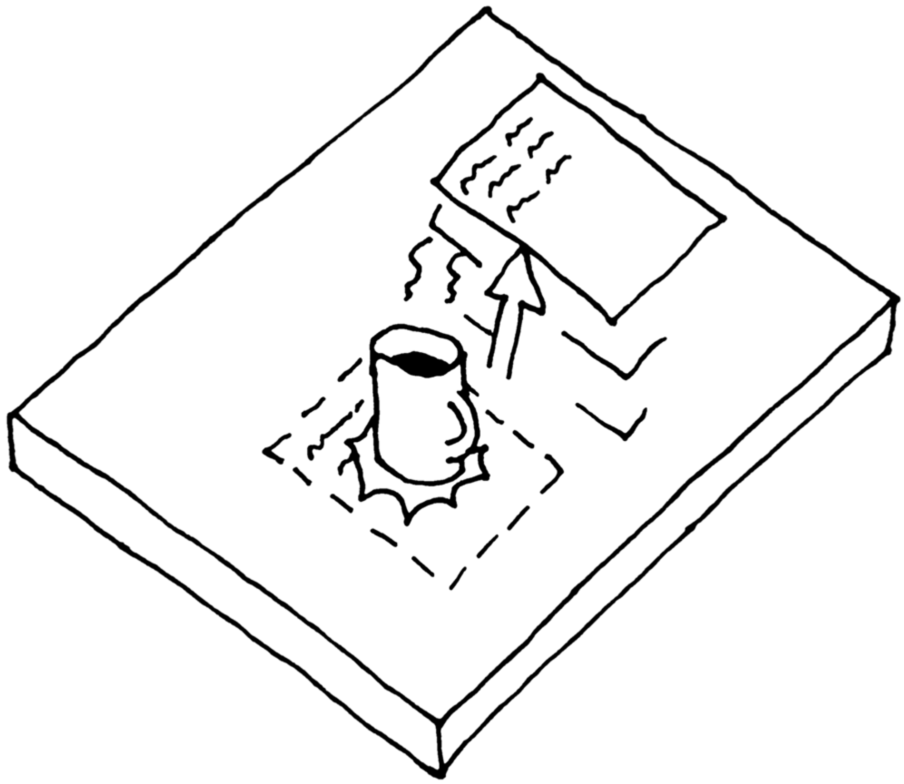
Cotting and Gross [1] introduced a display system where the tabletop detects obstacles on the table and displays data accordingly (Figure 24). Widgets are not displayed as rectangular windows like on traditional desktop UIs, but as *display bubbles* around the obstacles. Every bubble area is surrounded by a boundary area to ensure that the bubbles are not too close to any obstacles and distract the user's view.

In a more specific approach, Leithinger and Haller [2] invented user-drawn path menus on a pen-input driven tabletop (see Figure 23). When the user requests a context menu, it is not drawn immediately but the first item appears on a pen tip. The system waits for the user to draw a line with the pen and creates the menu alongside the path. The authors implemented the following four different menu creation styles: The fan out menu expands all items simultaneously. The card deck menu with a delayed appearance sequence corresponding to the pen stroke. The pearl string menu works like the card deck but with inversed order. And finally the trail menu which is similar to the pearl string menu, but if the pen is moved after the input, then the menu follows its trail.

Olwal and Wilson [3] go one step further. They do not only avoid obstacles, but integrate them into the working area. RFID markers and a camera detect the objects and then contextual information is displayed, surrounding the item. These objects are more than just obstacles, they work almost as (4-5) INPUT TANGIBLES, although they are like passive everyday objects.

Therefore:

Embed the obstacles in your screen design. Detect those objects and let the information flow around, either automatically or manually by the user, and if you are aware of the context of those items, adjust the arrangement and content accordingly.



This is a basic pattern with no further references within this language.

References

[1] D. Cotting and M. Gross, Interactive Environment-Aware Display Bubbles. *Proc. UIST '06*, pp. 245–254.
[2] D. Leithinger and M. Haller, Improving Menu Interaction for Cluttered Tabletop Setups with User-Drawn Path Menus. *Proc. TABLETOP '07*, pp. 121–128.
[3] A. Olwal and A.D. Wilson, SurfaceFusion: Unobtrusive Tracking of Everyday Objects in Tangible User Interfaces. *Proc. GI '08*, pp. 235–242.

(5-2) Dynamic Keyboard Relabeling **

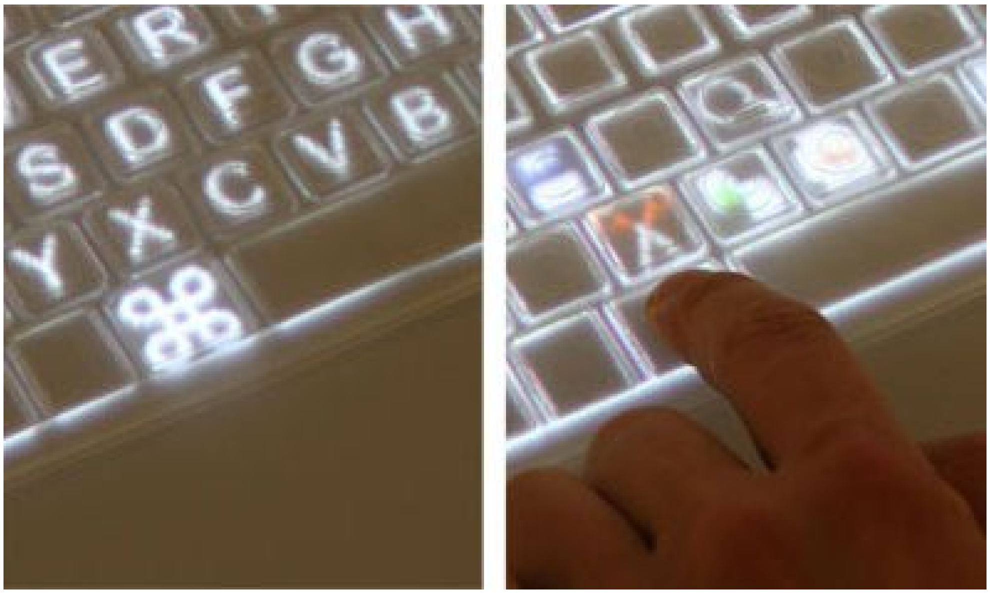


Fig. 21. SLAP keyboard relabeling [3].

... you offer tabletop text input by either implementing an (4-2) On-Screen Keyboard, putting a (4-3) Physical Keyboard on the table, or building your own (4-5) Input Tangibles. But for your tasks, you are searching for more possible ways to make use of the tabletop display or the objects on the surface, to create a more vivid tabletop environment.

Keyboards on tabletop should be flexible and unobtrusive, but there is a lack of the haptic feedback of quasi-mode keys like Shift or Control.

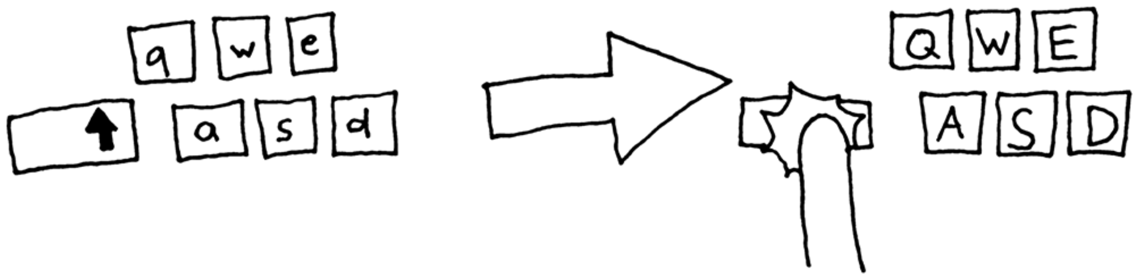
Tabletop keyboards always introduce new problems, dependent on the solution you choose: a physical keyboard is obtrusive, tangibles might be unintuitive, and on-screen keyboards lack haptic feedback. You can compensate all these issues by extending your chosen keyboard layout to change dynamically. This is easily implemented for an (4-2) On-Screen Keyboard, where the labels are just displayed anyway. Beyond the obvious capitals or control keys, there are further options, e.g., a combination of keys could be suggested to the user by highlighting the next key to be pressed. Bubbleqwerty [1] implemented this technique to avoid spelling mistakes, using a dictionary predicting the next most probable letter upon typing. Another advantage of keyboards with a changeable display is the possibility to highlight the pressed key. On smaller devices this can be a simple, but tremendous assistance for the user, consider the Apple iPhone [2], which enlarges the key pressed by the user. It is even possible to move the finger and look at the enlarged key to ensure that the right key was pressed.

An example for (4-5) Input Tangibles with dynamic relabeling of keys is the SLAP keyboard [3], which is a keyboard made of silicon. It overcomes the lack of haptic feedback, but it is less obtrusive than a (4-3) Physical Keyboard. Since it is transparent, the user can see the displayed items on the tabletop beneath the SLAP keyboard, but the user is able to feel the keys which she is typing. The application goes beyond the mapping for quasi-mode keys, e.g., for the function keys as depicted in the illustration. The user could choose her most familiar localized keyboard layout, e.g., an English, German, or French positioning of keys, or a completely user-defined custom layout.

Even when using a (4-3) PHYSICAL KEYBOARD, it is possible to change the keyboard layout, if the keys have small displays. A commercial available example is the Optimus Maximus keyboard [4], which consists of 113 OLEDs of 48x48 pixel size. While not being restricted to tabletops since it is a technology on its own, this is also one of its shortcomings: the tabletop display itself usually has no influence on what is being displayed on the keyboard.

Therefore:

Change the keyboard labels dynamically. Compensate for missing haptic feedback and offer alternative layouts.



This is a basic pattern with no further references within this language.

References

[1] U. Hinrichs, H. Schmidt, T. Isenberga, M.S. Hancock, and S. Carpendale, BubbleType: Enabling Text Entry within a Walk-Up Tabletop Installation, Report 2008-893-06, Department of Computer Science, University of Calgary, 2008.
[2] Apple Inc., iPhone, http://www.apple.com/iphone/
[3] M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. Hollan, and J. Borchers, SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. *Proc. CHI '09*, pp. 481–490.
[4] Art. Lebedev Studio, Optimus Maximus keyboard,
http://www.artlebedev.com/everything/optimus/

(5-3) Physical Object Storage Bin *

Fig. 22. An interactive SMART Board with storage for input devices [1].

... you have physical objects on your tabletop, e.g., a (4-4) Pen Input Device, a (4-3) Physical Keyboard, (4-5) Input Tangibles, a cell phone, or a laptop. These objects are used to extend your input variety in some occasions, but sometimes they just occlude the surface and you want to put them away.

To extend the input variety, electronic devices, tangibles, and other objects can be connected to the table. But those objects clutter the screen and when not being used, they easily get lost.

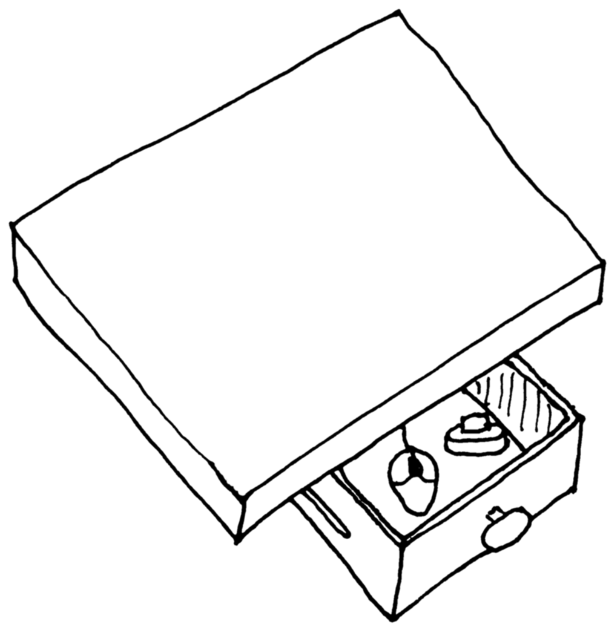
There are many physical objects to be used on tabletops, e.g., custom self-made tangibles, digital pens, or even desktop PC devices like a mouse or a keyboard. Even if you want your users to use nothing but their bare hands, you may have electronic devices or just everyday objects with you, e.g., a notebook, a cell phone, your keys, or just a glass of water. You do not want to place these objects on the table. A notebook wastes too much space, keys could damage the surface, and a cup of coffee may even be dangerous for the technology inside. On the other hand, you do not want to put these items far away, consider the notebook to configure the tabletop which has to stay close to it, or a cell phone or keys which you might forget if put too far away.

On every whiteboard or blackboard, there are storage bins for pens or pieces of chalk, respectively. It has become a ubiquitous element on whiteboards, which we use without thinking about - and everyone knows where to look for chalk or pens on such boards, just take a look at the bottom of it. Most tabletops usually do not offer this room for items. You can see people putting big cases on the table when putting tangibles on it, and without some specific place for pens you will often hear the question "Have you seen my pen? Where is it? Where did you put it?". To avoid this, create a storage bin beneath or next to the table. It should be clearly visible and not hidden, but not too big on an edge where users are supposed to sit, since it would increase their distance to the table unnecessarily. Likewise, a (1-4) NARROW SUBSTRUCTURE should be preserved. It also should suggest the user to put things in there. As shown in the illustration, the storage bin could match the shape of the objects, e.g., for pens, keyboards, mice, or even custom-shaped tangibles or everyday objects as keys, cell phones, or cups.

In scenarios where the user enters a lot of text, a (4-3) PHYSICAL KEYBOARD can be necessary. In their paper about text input on interactive tabletops, Hinrichs et al. [2] mention this scenario, and recommend a drawer for the keyboard. When the keyboard is not necessary because the text input task is finished, it does not clutter the surface and waste valuable space. On the other hand, as soon as you need it, the keyboard is quickly available again.

Therefore:

Mount a small item storage beneath or on the side of the tabletop to store unused physical objects. It should be large enough so that a keyboard or other large input devices fits in, but small enough to not annoy the user who has to sit in front of it.



This is a basic pattern with no further references within this language.

References

[1] Smart Technologies, SMARTBoard,
http://www.smarttech.com/
[2] U. Hinrichs, M. Hancock, C. Collins, and S. Carpendale, Examination of Text-Entry Methods for Tabletop Displays. *Proc. TABLETOP ’07*, pp. 105–112.

(5-4) Extending Reachability *

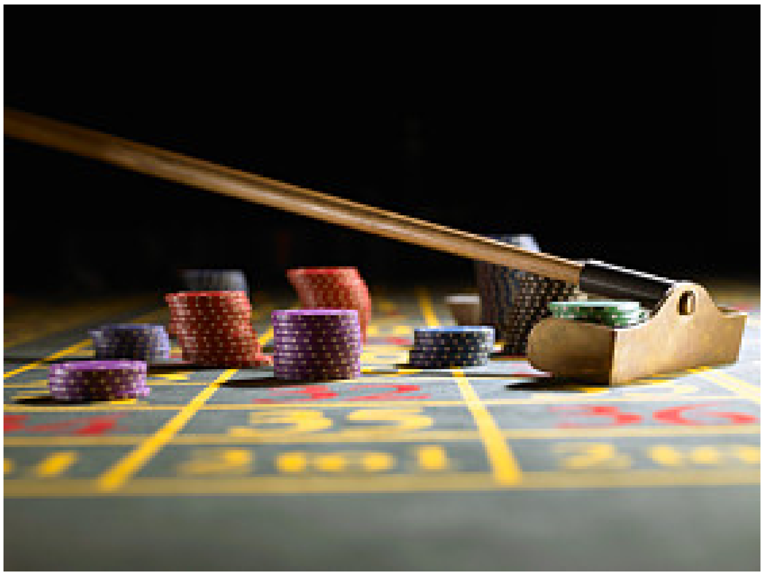


Fig. 23. A roulette rake mimics a physical example of reaching extension. Courtesy of Michael Blann/Digital Vision/Getty Images.

... you have built a tabletop system with a large surface that supports (3-2) Hand Gestures, and you possibly added (4-5) Input Tangibles or a (4-4) Pen Input Device. The tabletop offers a lot of space for a specific number of users, but sometimes there are not as many collaborators as expected, and there are unreachable areas on the table for some users.

Tabletops should be as large as possible, even if many users are working on it, it should offer enough space. But the larger the table, the further away are certain interface elements and other widgets on the table, and users cannot reach them anymore.

On traditional desktop computers, the area of reach is not an issue, since the mouse is a relative device and even if direct input is supported, e.g., on recent multi-touch enabled laptops, the surface is rather small. Tabletops are usually large surfaces, and especially if you create a collaborative environment with a rather huge display, there will likely be a scenario where users cannot reach elements they want to interact with. Many studies suggest that users then ask other participants to pass the elements, but as Zhang and Takatsuka [1] observed, this is not always the case. Their user tests included tasks with time pressure, an important aspect to be considered for meetings.

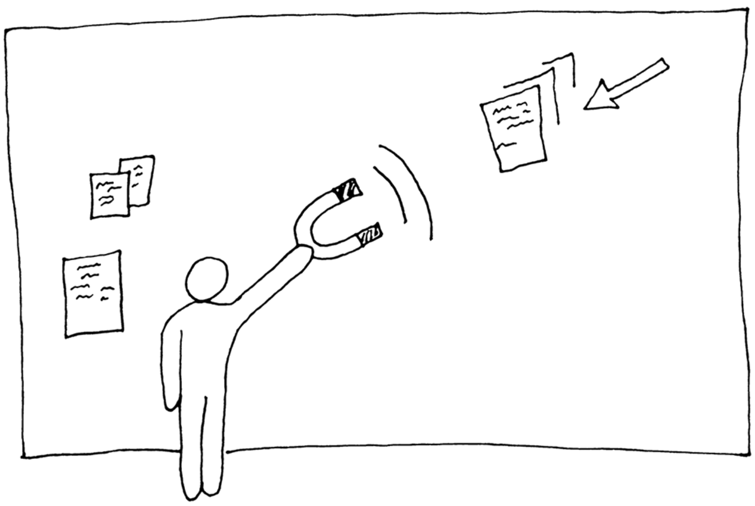
A study by Toney and Thomas [2] named important aspects for the reachability problem on tabletops. Different regions serve as different areas for work practice: The area close to the user is the working space, farther away but still reachable is the storage space for documents that will be used later on, and there is the area that is not reachable without any additional technique or walking around the table. They also mention the difference between sitting and standing, as the reach area obviously increases significantly when the users are standing at the table.

Nacenta et al. [3] compared six different reaching techniques on a tabletop in two different conditions, one with targets in user's hand's reach and one farther away and out of reach. The best technique was a radar, a small map that represents the whole table in miniature. Other solutions to be considered are sling shot and pantograph; in both approaches a pen stroke of a small distance is mapped to a longer movement alongside the stroke. The difference is that the sling shot gesture has an initial backward movement in the opposite direction, whereas the pantograph gesture has a pen stroke into the direction of the desired object.

A recently developed solution is the I-Grabber [4], which uses a bi-manual gesture for grasping unreachable objects. Two finger touching the table subsequently in 20 cm distance of each other initialize the I-Grabber, within direction to the second touch. By moving this finger farther away from the other, the I-Grabber extends its length by five times the finger movement. Releasing the second finger selects the object which is located beneath the hook in front of the I-Grabber at the same time.

Therefore:

Offer the user the ability to reach elements that are on the other side of the tabletop without having to get up from her place.



This is a basic pattern with no further references within this language.

References

[1] X. Zhang and M. Takatsuka, Put That There NOW: Group Dynamics of Tabletop Interaction under Time Pressure. *Proc. TABLETOP '07*, pp. 37–43.
[2] A. Toney and B.H. Thomas, Considering Reach in Tangible and Table Top Design. *Ext. Abstr. TABLETOP '06*, pp. 57–58.
[3] M.A. Nacenta, D. Aliakseyeu, S. Subramanian, and C. Gutwin, A Comparison of Techniques for Multi-Display Reaching. *Proc. SIGCHI '05*, pp. 371–380.
[4] M. Abednego, J. Lee, W. Moon, and J. Park, I-Grabber: Expanding Physical Reach in a Large-Display Tabletop Environment Through the Use of a Virtual Grabber. *Ext. Abstr. ITS '09*, pp. 61–64.