Single Display Gaming: Examining Collaborative Games for Multi-User Tabletops

Sebastian Pape, Laura Dietz, Peter Tandler

Fraunhofer Integrated Publication and Information Systems Institute (IPSI)
Dolivostrasse 15, D-64293 Darmstadt
{sebastian.pape, laura.dietz, peter.tandler}@ipsi.fraunhofer.de
http://www.ipsi.fraunhofer.de/concert/

Abstract. Playing games has always been a joint activity. New hardware developments such as multi-user touch-sensitive tabletop surfaces offer new possibilities to support games with information technology. This paper analyzes the properties of games that are relevant for choosing an appropriate hardware setup, e.g. the need for public or personal data for different players. These properties lead us to a categorization of games. Based on these results, possible hardware configurations for different game categories are discussed. Four sample games are presented by which the influence of the hardware setup on the game has been studied. The paper concludes with the lessons learned from our classification of games and their weaknesses and the next steps we are going to make.

1 Introduction

Tabletop games and computer games both enjoy a major popularity. While computer-support enables (for instance) the handling of complex rules, tabletop games benefit [3] from the fun of a co-located face-to-face situation [4]. In the context of our research on distributed collaboration, we have developed distributed computer games. As new hardware devices become available that allow multiple co-located users to interact simultaneously, such as the SMART DViT [5] and the MERL DiamondTouch [1], [6], we started investigating the potential of these devices to seamlessly combine distributed computer games with tabletop games.

While modifying the existing games we observed some important differences between the games which lead us to a more general classification of board and card games. Different properties of games require a different hardware setup in order to be fun to play. In this paper, we propose a simple and intuitive classification of games and investigate its consequences on the hardware needed. We end with a description of our games and the adaptations we had to make.

2 Classification of Games

In this section we name distinguishing properties for board and card games. We give examples for each class and talk about cases where our distinction might be inadequate. Although plenty of game classifications exist (e.g. [2], [12]), we could not find one that suited our needs.

Criteria

We distinguish between competitive and co-operative play, turn-taking and simultaneous games and games with and without secret information.

By *competitive* play we mean that every player[1] needs to play egoistically to win the game. All actions should only suit the players' purpose to win the game or at least to gain an advantage. On the contrary, in *co-operative* games all players share the same goal which they can achieve more efficiently when playing together than by themselves. It is quite obvious that competitive games can usually be won by only one player while in co-operative games all players win or lose together.

The next criterion focuses on the game's turn-taking or simultaneous character. We say a game has *turn-taking* character, when there is one special player or party who can do things the others are not allowed to and this attribute is passed on to the other players after a while. In *simultaneous* games all players perform their actions at the same time. We differentiate between games with synchronized phases and those with jumbling character. Synchronized phases means that the game is divided into rounds, and while a round lasts on, all players play simultaneously. When the round ends, there is a synchronization of the players' actions which will usually be some sort of evaluation. By jumbling character we mean that there is no special point of synchronization during the game, as is generally the case if the only evaluation occurs at the end of the game. While turn-taking in competitive games is quite natural it is mostly used as a restriction to the players in co-operative games.

For the last characteristic we examine if some players have information others don't have. It is quite obvious, that in competitive situations *secret information* is never a disadvantage. In co-operative situations players may be burdened with keeping information secret, which usually does not contribute to the common goal.

Examples

Competitive games. In Table 1 we enumerate examples for each of the four competitive cases. One could object that Bingo is played on secret sheets. But since a player cannot influence the next draw, he does not benefit in any way if all sheets are accessible by him. Thus we classify Bingo as a non-secret game. Since there is only one evaluation at the end of the game, Bingo is also an example for a game with jumbling character.

[1] Notice that the concept of player is meant abstract here and also includes groups if they can be understood as one player in the sense of the game.

As another example we refer to "best of five" rock-paper-scissors game, which is a typical synchronized-phases game, since both players act simultaneous and have the same possibilities and the end of a round is only used for synchronization.

Table 1. Examples for competitive games according to our categorization.

	With secrets	**Without secrets**
Turn taking	Skat, Scotland Yard, Battle Ship	Memory, Chess, Tic-Tac-Toe
Simultaneous	Rock-Paper-Scissors	Set, Bingo

Co-operative games. Since jigsaw puzzle is the most famous co-operative game, we describe some variants of puzzles. Usually jigsaw is played simultaneously without secrets, because this way one has the best chances to achieve the goal efficiently. In a "with secret" puzzle variant, all pieces of the puzzle are subdivided among the players, and each player must not see the pieces of his team-mates. If only one player at a time is allowed to put a piece, we get a puzzle with turn-taking characteristics. To achieve the remaining case, we combine both variants.

Hybrid games. Some games may have multiple contradictory characteristics so they cannot be uniquely categorized. Canasta has competitive as well as co-operative character, when playing in pairs. The pairs compete with each other while the two players of each pair co-operate. One cannot simply regard the pair as one player, since each pair member is not allowed to tell his partner about his cards.

3 Hardware

In this section we give an overview about useful devices for co-located gaming and examine how the classes of section 2 can be implemented.

Classification and Examples

We classify hardware according to two dimensions: one versus multiple outputs and sequential versus simultaneous inputs.

By *multiple outputs* we talk about situations, where some displays should not be accessed by all players. Otherwise separate displays are only used to increase space or resolution. Due to their prevalence and size, Personal Digital Assistants (PDA) come in handy.

PDAs can also be used to implement *simultaneous inputs*. But there are also Single-Display devices like the SMART DViT [1] and the MERL DiamondTouch [6] that can cope with simultaneous inputs. Even more interesting is the capability of *tracking and distinguishing users* which allows us to associate actions with a certain user.

If the device cannot track users, software based identification may be implemented, for instance via gesture recognition or restricting a user's operations on a certain region on the interface. This method can be used only in special cases, or it limits the freedom of a user's interaction, thus hardware tracking should be preferred in general.

Mapping the Classes to Hardware

It is quite obvious that advanced hardware is more expensive than conventional one. If we want to implement a game that belongs to a determined category, which hardware provides the best trade-off?

When every player needs to have access to secret information, we have to equip him with his own display. Handheld devices like PDAs may be combined with a Single-Display device and fulfill this requirement in a convenient way, although the input capabilities might not be needed [3], [4], [8].

While the aspect of secrets makes primarily use of output issues, the use of turn-taking versus simultaneous games mainly focuses on input issues. Games with *turn-taking* can be implemented using floor-control mechanisms. Since people usually perform one action after another, devices with single-input capabilities are sufficient for this situation. *Simultaneous* games need an interface that can handle multiple players entering data in parallel.

Competitive games need knowledge about which player performs an action, e.g. in order to update the score correctly. In games with floor control this can natively be implemented by keeping track of the floor-owner. In simultaneous games either the hardware or the software must be capable of distinguishing the players. When playing *co-operatively*, player identification is only needed to control rules of the game, e.g. to monitor restrictions.

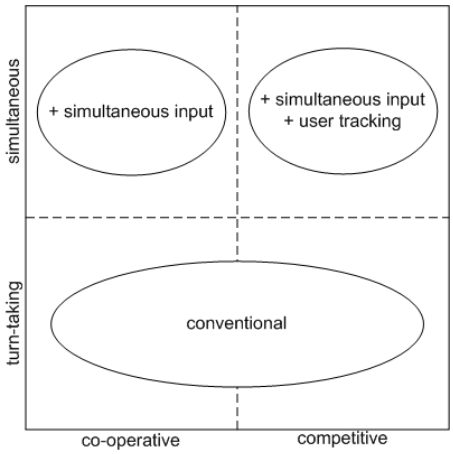

Fig. 1. Input hardware needed for each class.

4 Experiences Applying the Classification

In this chapter we describe our experiences applying our classification to games we originally developed for the distributed scenario.

4.1 Game Description, Hardware Setup and Goal

The *Coast Puzzle* [10] is a jigsaw game that allows multiple players at different computers to puzzle jointly. In the basic version all players share the same view and can arrange pieces of the puzzle simultaneously. In the extended version all players share the same arrangement of the puzzle pieces, but every player sees a different picture.

Set is a card game, where twelve cards are laid out open. Each card contains one, two or three matching objects, in one of three colors, shapes, and shadings. A set consists of three cards where every attribute is either equal or different on all cards. All players try to spot a set. The first successful player gets the matching cards and restocks the free places from the deck.

MemoSet [7] is a computer supported variant of Set where twenty cards are laid out disguised. Each player is allowed to select and deselect cards to a maximum of three selected cards (fig. 2). Selected cards are turned around and are visible to all players. All other rules stay untouched.

Fig. 2. Session with our MemoSet implementation on the MERL DiamondTouch.

All described games are programmed in Smalltalk and make use of the COAST groupware framework [7]. COAST eases the implementation of distributed applications by providing shared data models. Thus the developer does not have to care about synchronization, change notification and data distribution.

The hardware which was to our disposal was a SMART DViT and a MERL DiamondTouch. Both devices can handle simultaneous input, while only the DiamondTouch is able to distinguish users.

With the objective of playing those games conveniently on our hardware we improved the original software. Since the games are usually played on a tabletop we embedded touch-sensitive devices in a desk, creating a computer augmented tabletop.

4.2 Analysis of the Coast Puzzle and MemoSet

According to our classification the basic variant of the Coast puzzle is a co-operative, simultaneous game without secrets. Thus an adaptation of the software using the SMART DViT is sufficient. In the extended variant no useful public information is left which could be displayed without revealing one player's secrets. Therefore there is no benefit of using a tabletop for this variant.

On the contrary, Set and MemoSet are competitive games with also simultaneous and no-secret character. So we make use of the MERL DiamondTouch.

4.3 Implementation

Due to the fact that each of the boards has a different (but similar) API, we built a wrapper API to hide the implementation details from the application.

Implementing Multi-User-Tabletop-Support for the basic variant of the Coast-Puzzle is straight forward, since only modifications to the event handling are needed. On the other hand, the needed modifications for MemoSet are fundamental. As we already noticed, identification of the players is necessary, so in contrast to the distributed scenario we have to cope with multiple players on the same computer. Thus a change to our model was inevitable. We also added a turn-taking variant of MemoSet which allowed us to use hardware not capable of player identification.

4.4 Evaluation of Our Game Implementations

When doing some test-games it turned out that playing co-located on one board together was a lot more fun than playing distributed or co-located on many machines. Playing on a "board" was also more intuitive and provided the look and feel of traditional games, so collisions during the interactions occurred seldom. We also noticed that software-emulation for distinguishing players by artificial turn-taking turned out to be annoying compared to the original hardware capability. So, if the adequate hardware is available it should always be the first choice.

5 Conclusions and Research Vision

Our classification turned out to be quite useful, although there were some games that could not be classified satisfactorily. A further improvement of our classification will

include hardware recommendations for hybrids and include classes that cannot be applied to the tabletop (like the extended Coast Puzzle).

As one of the next steps, we compare UI requirements of games in the distributed, single-user and co-located scenario. We want to know which awareness features are necessary and how UI widgets have to be adapted.

The same will be done for the technical aspects: How to we have to adapt existing groupware frameworks to support transitions between distributed, co-located, SDG and single-user settings? How can this be achieved with as few modifications to the client applications as possible?

It would also be interesting to further investigate the scenario with one public tabletop and additional PDAs for each player as well as including physical objects, which was proved to be important on user acceptance [11]. Other interesting research developments might be improved software simulation for simultaneous games without annoying effects to the players.

6 Acknowledgements

We would like to thank all colleagues for helpful discussions. We especially thank Wolfgang Hinrich for the implementation of the wrapper API and Nils Knappmeier for connecting it. We also like to thank our co-operation partners MERL and SMART for providing the hardware.

7 References

[1] Dietz, P.H.; Leigh, D.L., “DiamondTouch: A Multi-User Touch Technology”, ACM Symposium on User Interface Software and Technology (UIST’01), pp. 219-226, 2001.

[2] Lundgren, S., Björk, S., “Game Mechanics: Describing Computer-Augmented Games in Terms of Interaction”, Proceedings of TIDSE 2003 (Technologies for Interactive Storytelling and Digital Entertainment), Darmstadt, Germany 24-26 March 2003. http://civ.idc.cs.chalmers.se/publications/

[3] Magerkurth, C., Stenzel, R., “Computer augmented board games”, Journal for interactive and communicational media – i-com, 03/2003, pp. 6-14

[4] Mandryk, R., Inkpen, K., “False prophets: Exploring hybrid board/video games”. Extended Proceedings of CHI’02, pp. 640-641. ACM Press, 2002.

[5] Martin, D. A., Morrison, G., Sanoy, C., McCharles, R., “Simultaneous Multiple-Input Touch Display”, UbiComp'02 Workshop on "Collaboration with Interactive Walls and Tables", Sep. 29 2002.

[6] Mitsubishi Electric Research Laboratories, “DiamondTouch Hardware", http://www.merl.com/projects/DiamondTouch/

[7] Pape, S., “MemoSet” http://www.memoset.pallas.net/

[8] Schoemaker, G. B. D., Inkpen, K. M., “Single Display Privacyware: Augmenting Public Displays with Private Information”. Proceedings of CHI’01, pp. 522-529, 2001. http://citeseer.nj.nec.com/444794.html

[9] Schuckmann, C., Schümmer, J., Seitz, P., ”Modeling Collaboration using Shared Objects.” Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group'99.), pp. 189-198, New York, ACM, 1999.

[10] Schümmer, T., Schümmer, J., “COAST: The COAST-Puzzle”, http://www.opencoast.org/download/puzzle.pdf
[11] Scott, S.D., Grant, K. D., Mandryk, R. L., "System Guidelines for Co-located, Collaborative Work on a Tabletop Display" in Proceedings of the 18th European Conference on Computer Supported Cooperative Work (ECSCW'03), Kluwer Academic Publishers, Amsterdam, NL, September 14-18 2003.
[12] “Game classification” From: Wikipedia. http://en.wikipedia.org/wiki/Game_classification

8 About the Authors

Sebastian Pape studied mathematics at Darmstadt University of Technology and still studies computer science there. He is a student member of Fraunhofer IPSI since 1998 and since 2004 he also works at the Department of Computer Science, Telecooperation of Darmstadt University of Technology. His main research interests are CSCW, distributed applications and networking issues, he is further interested in computer security and cryptography. In 2002 he developed MemoSet and spends some of his spare time with board and card games as well as computer games.

Laura Dietz is research associate in the CONCERT division at Fraunhofer IPSI since 2002. She studied computer science at Frankfurt University. Her diploma thesis was about distributed co-operative text authoring. She currently works on the DIGITAL MODERATION project. Her research interests are technical as well as usability aspects of co-located and distributed synchronous groupware and how the outcomes of different groupware domains (e.g. CSCW, CSCL, co-operative gaming) can be applied to each other.

Peter Tandler is the head of the CSCW research group of the Concert division at Fraunhofer IPSI. His research interests are within the areas of CSCW, integration of virtual and physical environments, new forms of human- and team-computer-interaction for roomware components. He is currently working on his Ph.D. in the context of application models and software infrastructure for roomware environments. He studied computer science at the Technical University of Darmstadt, Germany, with education and psychology as additional subjects.