Collaborative Gaming in Augmented Reality

Zsolt Szalavári, Erik Eckstein, Michael Gervautz
Institute of Computer Graphics
Vienna University of Technology
Karlsplatz 13/186/2, A-1040 Vienna, Austria
szalavari|eckstein|gervautz@cg.tuwien.ac.at


*

1. ABSTRACT

We introduce a local collaborative augmented reality environment for home based entertainment. We propose a *setup* for multiple users with see-through head-mounted displays, allowing dedicated stereoscopic views and individualized interaction for each participant. This hardware arrangement does not hinder social communication, which is essential for playing games. To maintain a *high quality game experience* we use face-*snapping* for fast and precise direct object manipulation. We derive semantic actions from snap events to pre-defined *regions* in the virtual gaming space. Combined with the game regions, we introduce a *layering* concept allowing individual views onto the common data structure. This combination creates a *powerful automatic privacy mechanism*, which makes security management dispensable at runtime. We demonstrate our system with a board-game Mah-Jongg, that relies heavily on social communication and the need of private space.

1.1 Keywords

Augmented reality, interaction, CSCW, virtual gaming.

2. INTRODUCTION

Interactive gaming is becoming more and more one of the dominant application areas for computer graphics. The related industry is growing very fast both in the location based entertainment (LBE) and the PC-game domain. We focus on a computer-based gaming environment for home usage which is especially suitable for multi-player board-games. These type of games are driven by two major aspects: *Social communication* and the freedom to maintain individuality in a *private space*.

---

* This enhanced illustration shows the view of a player in the augmented gaming environment, while playing Mah-Jongg.

The social communication aspect can be clearly observed with non-computer based multi-player board-games like Mah-Jongg, Trivial Pursuit, etc. Computer based games, which gather around some classical computer-game idea (like Jump-and-Run games) often fail to support this type of communication and brought criticism to the computer games.

The other important aspect is privacy. Parts of the "gaming space" like the table in a board-game are public. Every user has the same visual access to this common information. Parallel to that the game has to maintain a private space for the player. This space provides security for individual strategic decisions.

These two aspects are weighted differently in different kind of games. In a game like Trivial Pursuit for example, the social communication has more weight than the private space. Privacy is only needed to hide answers from the other players. Whereas Mah-Jongg needs a private space for each user to hide tiles from others during the game.

We identified the *ability to display different information* to each participant, *unhindered social communication*, and *precise and fast interaction* with the game as the crucial factors for augmented gaming. We describe a technology setup which provides a good base for this requirements and present techniques to overcome limited precision in an augmented reality environment.

3. "SALON DE JEUX" - SETUP FOR A MULTI-USER AUGMENTED GAMING ENVIRONMENT

3.1 Related Work

Lot of expectation surrounds virtual reality (VR) technology to bring new solutions to gaming with computers. Observations shows results against these expectations, however the technology is appealing and everyone wants a personal experience [17]. Pausch et. al. came to the conclusion that it is possible to suspend disbelief. Yet to convince players is very hard, even with familiar content and story presented with high fidelity hardware.

Distributed multi-user VR systems like DIVE [7] try to incorporate communication for geographically distant users. Communication distribution is additional to application demands. Depending on the type of information that is communicated (avatar representation, text, audio or even video data), high network performance and intelligent distribution strategies have to support these tasks.

Local multi-user systems like CAVE [8] and Responsive Workbench [15] provide social communication naturally, since participants are co-located in one room. However these systems provide an experience of group immersion,

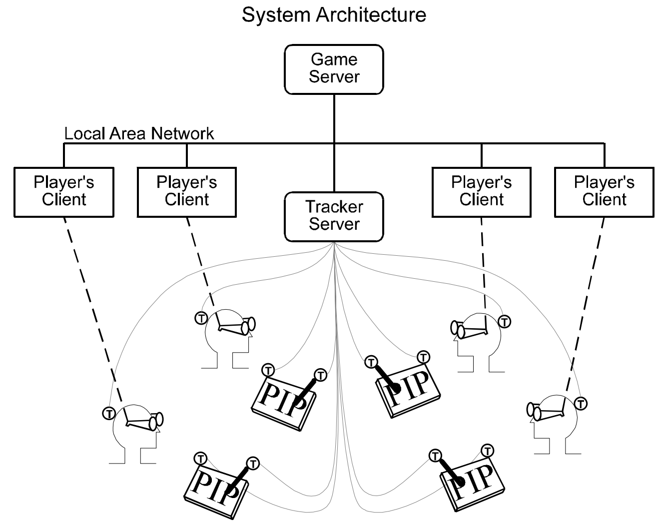

Figure 1. The client-server architecture of our system.

rather than a dedicated multi-user support for a number of participants with different private views.

HMD based Augmented Reality (AR), as pioneered by Sutherland [19] and brought to a first real application by Feiner et. al. [10], offers a good compromise between unencumbered social communication and private display of information. Collaborative AR systems, such as Shared-Space [3] and Studierstube [11] have extended this approach in applications to scientific visualization and collaborative work to a group of participants.

3.2 Our Approach

The setup we have chosen is similar to that of *Studierstube* [21], consisting of private see-through head mounted displays (Virtual I/O i-glasses!) and a Personal Interaction Panel (PIP) [20] for each user. HMDs and interaction devices are tracked in position and orientation with a magnetic tracking system. The see-through HMD does not block the view onto the environment, so additional to verbal communication gestures of other players can be recognized. Using dedicated display devices for each user, display of private information is supported. The Personal Interaction Panel consisting of panel and pen allows powerful two-handed manipulation of virtual objects with high precision. Interaction with the game is independent from other users as the interface serves as personal tool.

The system architecture consists of a game server maintaining the graphical database of the game and running the simulation based on the input data from magnetic trackers and input devices. Clients for each user render the customized view on the shared environment. Tracker data and information of the buttons on the pens are distributed over the local network using a tracker server. Using a multi-cast group every participant gets synchronized updates of tracker data for rendering.

Figure 2. A view onto the hardware environment visible to the players. Each player wears a see-through head-mounted display and interacts with a Personal Interaction Panel.

This approach allows scalability in the number of users to a certain extent. An overview of the system architecture can be seen in Figure 1. Figure 2 shows our VR-hardware components as seen from an observer.

4. INTERACTION TECHNIQUES

We augment the real environment with additional information. To avoid context switching from one space to the other, the virtual overlay has to behave like the real world. The fewer new metaphors we introduce the more natural the interaction will be and the easier to play the game.

According to [13] interaction can roughly be categorized into *object manipulation*, *navigation*, and *system control*. Interaction in our gaming applications covers besides some obvious system controls mostly direct object manipulation. Users play the game in our environment essentially by manipulating virtual objects (tiles, dices, cards) in front of them. Most of the actions that will occur during direct manipulation are actions such as “Put-That-There” inspired by the motto of [4].

Our concept relies on a proximity based drag-and-drop model for 3D user interaction. Such actions have in addition to the geometric transformation also semantic meaning.

4.1 Face-snapping for Precise Object Placement

To implement this simple concept *we need to overcome the limited precision* of the applied VR hardware. Low resolution HMDs and the lack of haptic feedback decreases overall precision of systems based on this hardware. Additionally magnetic trackers provide noisy position and orientation measurements decreasing performance in object manipulation tasks. Due to that objects can not be placed very exactly in space. Especially, moving one object face-aligned onto another which is a very often performed task, is really hard to achieve. Instead of applying quality enhancing techniques on raw tracker measurements, like filtering or prediction, we enhance manipulation precision on the scene level. We identified snapping as a very powerful tool for aligning objects precisely, speeding up manipulation tasks.

4.1.1 Related Work

Several solutions have been proposed for the problem of precise object placement. Collision detection, an often used approach in common VR-systems, does not add very much value to that problem. The avoidance of interpenetration does only help a little for the alignment task and is computationally expensive.

Bier’s important work on snap-dragging in 3D [2] was one of the first solutions for direct 3D manipulation with high precision. Although his method is very intuitive, it can not be straightly used for direct manipulation, too many additional commands have to be set up during the task.

The whole field of constraint based modeling (summarized in [14]) deals also with high precision. There, constraints are used to automatically keep some kinematic relationships between parts of the scene. Some systems maintain a constraint graph to store this constraints for further use [9].

4.1.2 Our Approach

We use face-snapping, where objects, which are close to each other are aligned automatically by calculating an alignment constraint between the faces – the objects snap onto each other. If an object has to be placed onto another object, the user simply moves it close to the other, face-snapping aligns them automatically.

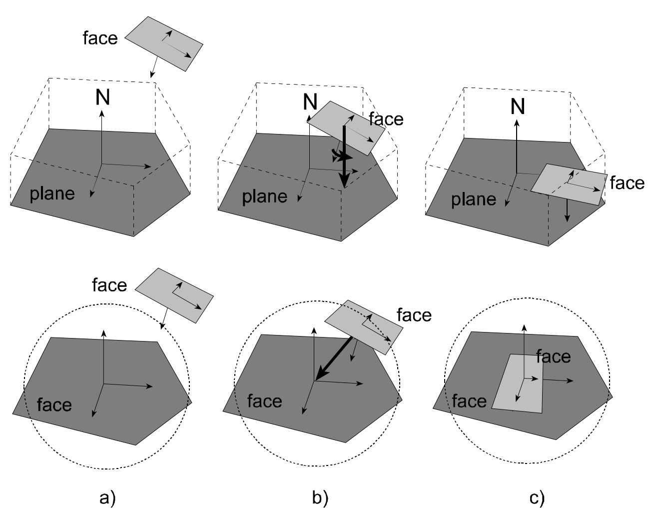

Figure 3. Face-to-plane snapping (upper row) and face-to-face snapping (lower row). A face is approaching a constraint region (a), moves into the region (b), and finally snaps (c).

In comparison to Fa's [9] automatic constraint recognition process, our approach does not introduce explicit constraints, which can be inserted and deleted from a data structure.

During dragging of an object by the users' pen for each frame all possible snapping conditions are checked and the one with the highest priority is performed immediately. The dragged object is moved according to the geometric constraints defined by the two geometries. Because the detection and calculation process is invoked after every pen movement, the two faces keep snapped until the snapping condition is no longer valid. If the user releases the object during a snap, the object stays aligned with the other, but no constraint is kept for the future.

Looking at the previously presented object manipulation requirements of our application, we find that we need only a very small subset of constraints. We use face-to-plane (binding three DOF) and face-to-face constraints (binding all 6 DOF) which are enough to place for example game-tiles in a board game. The face-to-plane constraint allows 2D translation and 1D rotation on the plane (see Figure 3 upper row), the face-to-face constraint fixes the two faces onto each other (see Figure 3 lower row). In both cases the snapping condition for the automatic recognition process measures the distance and the relative angles between the faces. The distance is measured between the plane and the center of the face for face-to-plane constraints and between the center of the two faces for the face-to-face constraint, respectively. If both, distance and angle between planes are lower than a defined limit, the snapping condition is satisfied. To avoid jitter in the snapping, which results from noisy tracker coordinates, we applied a hysteresis on the snapping condition.

In environments with many potential snapping-faces still a high computational effort is needed to find the nearest snapping face pair. However the constraint-recognition algorithm must be very efficient to be used in as real time application, so we use an axis-aligned bounding-box hierarchy for each, the static part of the scene, the geometry on each PIP and the dynamic part of the scene. This algorithm is similar to the broad phase algorithms like [12] for collision detection. For a small number of dragged objects (each user has just one pen) we can achieve real-time performance.

The result of the constraint recognition is a list of valid snapping face-pairs. For simplicity we process only one snapping action at a time. We choose the nearest face pair (i.e. having the smallest difference in angle and distance) to perform the according transformation to the dragged object. This kind of priority selection behaves also very naturally because objects snap only to the nearest objects as expected by the user.

Besides the described advantage of fast and precise object placement the snapping movement itself gives feedback for the user that he has placed the object correctly. So there is no additional effort needed to show that a positioning action has been completed.

4.2 Spatial Controlled Semantics

Object placement using snapping is not purely a geometric alignment task. This type of manipulations also effect semantic meaning. The problem is to read the semantic meaning out of the geometric action.

4.2.1 Related Work

Mine et. al. [16] introduced the idea of gestural actions to identify this type of actions and trigger semantic meaning. One nice example was the interpretation of the movement of throwing an object over the shoulder as deleting this object.

Bukowski et. al describe in [6] a software framework to populate 3D environments with objects on the "WALKEDIT" desktop based 3D design system. They map 2D cursor positions into 3D space and enhance object placement with pseudo-physical behavior. In a second step they associate objects implicitly based on geometric characteristics, like distance to nearby entities. Associated groups are dynamic and can have hierarchical structure. This work has shown that "magic" - i.e. pseudo-physical rules - can enhance interaction with 2D interfaces and increase productivity.

4.2.2 Our Approach

We introduce the general concept of *regions* as an extension to previous approaches. Regions are dynamic, logical groups of objects in a scene. They act as a container to hold groups together, identifying some kind of association of members. Regions can also be placed into regions, allowing the hierarchical association of objects contained in these groups. Regions are unambiguous to objects located in them.

To employ this organizing method in our board gaming scenario, regions are assigned to geometric objects with some spatial extent. Thus logical groups can be formed by recognizing geometric conditions between this extent and arbitrary objects in the scene. An object is moved from one region into another if it is moved in 3-space out of the area of one region into the area of another region. There are three problems with this approach:

- It is very time consuming to calculate geometric intersections between regions and individual objects, especially if there are many regions in the environment.
- If regions do not have a visual representation (as they are only logical groupings), the user needs a separate feedback for entering and leaving a region.

- If regions overlap, there has to be a simple decision mechanism to decide for each object to which region it belongs.

In our direct manipulation scenario of the game, we identified the geometric condition to be identical with the snapping condition of two faces or a face and a plane.

Using this approach, snapping is a mechanism to read out semantic meanings from the geometric actions. The same interaction event can be used for precise direct manipulation and semantic control. Snapping constraints give the necessary visual feedback of docking, e.g. game-pieces stick dynamically to other game-pieces. In parallel to that a semantic action is triggered to associate the manipulated object to the target, where it was moved. We call this process *region transition*, assuming that all objects are associated to a region in the beginning. This is guaranteed by the game engine, which associates each participating object (i.e. game-pieces) to regions. Overlapping geometric conditions are resolved by snapping priorities, assigned knowledge-based. This is done in advance, in the design process of the application.

Spatial controlled semantics read out from geometric actions has been verified as a powerful technique for game-piece manipulation. Using this paradigm, grouping of tiles and thus playing games is very natural and provides a high quality input to the game-engine, running the game logic. Manipulated items in a game should however not be always visible to each participant, therefore a privacy and publicity management of visual information is needed. We present an security management approach, which is highly flexible, and suits different AR or VR application scenarios.

5. THE CONCEPT OF SECURITY AND PRIVACY

Most multi-user VR applications present the synthetic environment to each user in the same way, albeit from different viewpoint or resolution due to LOD selection algorithms. Like early 2D collaborative systems they simply replicate the common database and show the same visual content.

5.1 Related Work

Smith and Mariani describe in [18] a mechanism to present subjective views in an existing distributed multi-user environment. Query results in the shared database are presented subjectively to users by assigning object modifiers to found entities. Thus relationships and representation of the requested data can be tailored to user specific needs and additional information is not cluttering up the scene for other participants.

Agrawala et. al. present the Two-User Workbench in [1], introducing the potential to display customized views to two users. They also propose different partitioning techniques to present information.

While these solutions provide security for visual information, multi-user interaction in shared environments leads to even more complex problems. W. Broll offers in [5] a good overview on what distributed application might implement to handle concurrent object access. The paper identifies locking and the use of master entities as primary solutions for multi-user interaction in VR systems.

5.2 Our Approach

We introduce a concept similar to that found in CAD or GIS packages for *layer*ing information. Our investigations lead to the result, that in most scenarios groups of objects with the same security status can be identified. So usually the number of different security statuses is much less then the number of objects, which have to be considered for security management.

Furthermore, we discovered that in gaming scenarios security “presets” can be found for certain parts of the game, which do not change during the play, e.g. the table is always visible for each participant, but one player’s game-pieces remain only accessible for him- or herself. This defines security characteristics for logical parts of the game and thus for logical groups of game-pieces.

We identified regions - described in the previous section - as the groups to hold a specific security state. The security information controlling the behavior of the common scene traversal for different players is concentrated here, coding specific variations. This concept can be best described with a set of keys and locks as shown in Figure 5. The locks are associated with objects in the scene. Different keys are handed over during traversal of the scene graph to the different players, when entering the hierarchy associated to a region. Another dimension to our key system is added by the diversity of scene traversal actions, like rendering, picking, snap-condition evaluation, etc.. Therefore we use complex keys encoding security for each action type, presented by different edge-shapes in Figure 5 b) and c).

To implement this approach we encode the security information in *layer-nodes*, stored in the common scene graph, which also holds the whole graphical database. This concept allows easy replication to clients. To render this common description in a player customized version, only the player-id has to be set at the very top of the hierarchy. We encode security information in a 2D matrix for different players of the game and different actions.

So unlike many other approaches privacy information is not stored as object property of each entity, but at a higher level as region property. Layer-nodes can be everywhere in the scene-graph. In this way it is possible to form a hierarchical security structure. Sub-layers inherit rights from the super-layer. This allows to define group rights in an easy way.

Finally, we extended the underlying system by components to interpret security information at any point of the hierarchy. Using information passed down in the hierarchy, sub-hierarchies, e.g. objects, or in our case game-pieces, can react to the security information, behaving differently depending on the security setting at higher levels.

Thus members of groups inherit security state, transition to another group automatically causes an implicit state change as shown in Figure 4 and Figure 5. However, our security layer concept is so general, that it can implement other security strategies. Storing security information on item basis by defining a layer to each object resembles previous work.

The system is open for dynamic modification and reconfiguration of security presets at runtime. Although we think that a careful application design with the identification of good security presets can avoid this, and free the application from additional security management.

The presented approach leads in our gaming scenario to a very effective and powerful mechanism for security management. In the design step we identified the logical regions of the common table and player owned regions to carry security information.

In Mah-Jongg presets for one player are set, so that he or she can see and manipulate the own game-tiles, but no other player can see any valuable information. While a player moves one tile of his own set of tiles to the common table-top, a region transition is triggered by the geometric constraints of snapping. This region-transition from one region to another causes the game engine to move the sub-graph of the game-tile from one place in the hierarchy into the place of the destination region, i.e. the tabletop-region.

As the game-piece now inherits a new security information, it behaves differently, and is rendered visible for every player, due to the security settings of the common tabletop-region (A more detailed description can be read in the implementation section.).

Note, that the game engine does know nothing explicitly about security management, as the only step it makes, is to move one sub-hierarchy in the scene-graph into a different place. This approach is simple yet powerful enough to meet most requirements. It is transparent to the application and does not cause additional management load as presented in Figure 5.

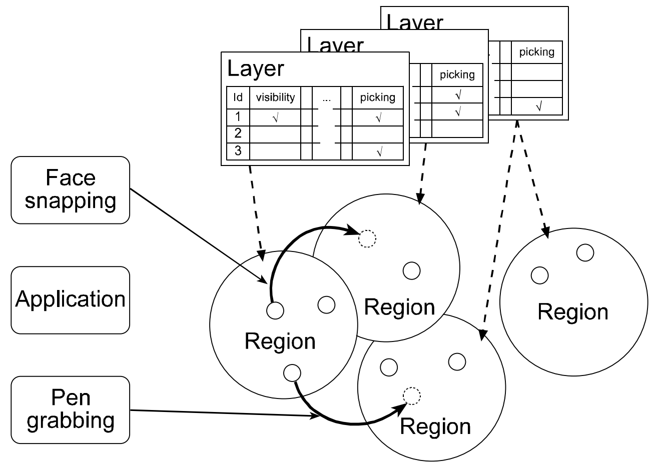

Figure 4. Object transitions between regions can be triggered by events coming from snapping, the pen or the application itself. Layers encode privacy information for players and are assigned to regions. Transition of an object (dots in the regions) into regions with different privacy, will change the privacy of the objects automatically.

5.3 Private Help

Additional to simple modification of object appearance, the presented security layering mechanism is easily applied to achieve *private help*. Private help is any kind of help information which can only be seen by the user, who requested the help. Other users should not see the help information and would be probably disturbed by it. Sometimes it is desirable that the application gives a hint to a specific user, which other users should not see, e.g. in a teacher – student collaboration, where only the teacher can see the right answer.

To implement private help all the geometry, which form the help information, has to lie on a specific help-layer. Only the user, who should see the help information has rights on the help-layer. Other users do not have rights on this layer and therefore they can not see, nor manipulate the help information.

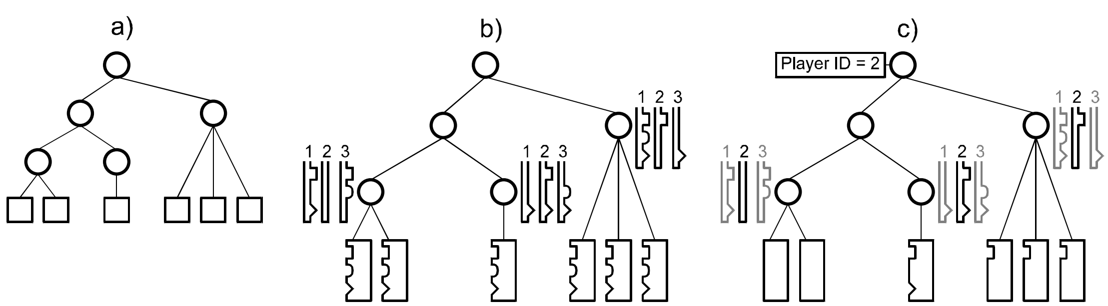

Figure 5. Representation of the scene graph. Figure a) shows the graph as the game engine sees it. Figure b) shows security information - presented as “keys” - in nodes belonging to different user IDs. The leafs hold sub-graphs, with general “locks” that represent different behavior during traversal. Figure c) shows how player 2 sees the common scene graph.

The application domain of multi-player games is a good test-bed for our security management approach. Depending on the definition of layers, independent subjective versions of the same scene can be presented to participants providing a private view. The private space of one user is protected from other users, while public spaces allow access to everybody. Independent from visual appearance, manipulation or snapping characteristics may be hidden from others. Assignment of rights remains an open question and application programmers have to adjust security presets carefully to provide meaningful combinations.

6. IMPLEMENTATION AND RESULTS

A large number of different games seem to fit excellent into our concept. All kind of non-computer based board games like pictionary, any card game are good candidates but also many of the existing multi-user console games can gain additional benefit of the augmented reality setup. For evaluation of our system we selected a game which is widely known around the world and relies also on the use of social communication channels: Mah-Jongg.

The current hardware set-up as shown in Figure 1 supports up to four users playing simultaneously in an augmented environment. Rendering is done on a SGI Maximum Impact R10000 with Impact Channel Option (ICO). The ICO is utilized to split the frame buffer into four regions and to convert output to VGA signals. Using converters images are presented in the HMDs. Our implementation using Open Inventor can deliver independent line interleaved stereoscopic rendering for four players at about 10 frames per second.

6.1 Mah-Jongg

6.1.1 The Game

Mah-Jongg is a very old traditional Chinese game, with roots going back into the dust of ancient centuries (Figure 6). In the beginning it's been only played by emperors and those in the know, because it's secret. Later public got access to it, and in our century it become popular around the world. The high number of enthusiastic players developed many variants of the game, yet the basic rules are still kept.

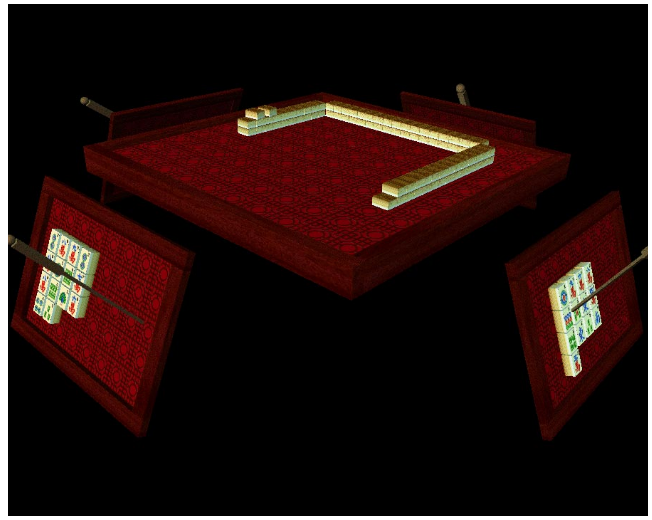

Figure 6. Mah-Jongg. The augmentation overlay which is rendered and presented in the see-through HMD, showing four PIPs.

We have selected Mah-Jongg as the major implementation example of our concept, because it is known world-wide, the main rules are easy to understand, and many of our system features can be presented with it. Mah-Jongg is played in our "Salon-de-Jeux" setup by sitting around the table wearing see- through HMDs. Each user also has a Personal Interaction Panel to manipulate the augmented game. The panel carries one players tiles, which are manipulated with the stylus.

6.1.2 Spatial Controlled Semantics

To play Mah-Jongg, mostly tile shaped game-pieces have to be transferred from one location to the other. Tiles are being transported from the wall to the users hand, within the users hand from one place to another to form groups, and from the hand to the table. We have defined several regions for the game: the wall region, the table region, the region on the PIP representing the users hand, the region of the pen for the transition of tiles. In contrast to all other regions the pen region has no assigned snap condition geometry. The region transition of tiles into and from the region is triggered by pressing and releasing the button on the pen (see also Figure 4).

Figure 7. A grouping sequence. A tile is grabbed with the pen and snapped to another tile to form a group. This action triggers semantics to insert the help-shovel showing additional help information. In this case the name of the combination is displayed.

Players can independently manipulate their private tiles. Tiles snap to each other providing visual feedback and to trigger semantics. In this way groups can be formed and braked dynamically by pushing tiles together or moving them apart. Grouping generates additional regions, allowing to add other tiles to that sub-region. Additionally the grouping mechanism gives feedback to the game engine to provide help information and determine game state (see Figure 7).

6.1.3 Privacy in Mah-Jongg

Naturally the real game relies very much on the honesty of players not to look into other players tiles. This gives the game a secret and mystic touch. In general the rights to see tiles and to manipulate them is governed by game rules and tradition. The gaming situation decides what tiles can be picked up by which user. In the real game these conflict situations are solved over the social channel.

Our layering concept allows to support this type of privacy by assigning different security levels to regions. Tiles on a players panel can only be seen and manipulated by himself (see Figure 8). The texture containing the tile's sign is switched on and off by inheriting privacy information from the PIP region. Picking up the tile by pointing with the pen inside it and pressing the button transfers the sub-graph of the tile to the pen region. The pen region has the same privacy settings like the PIP, so that other players still can't see the texture while manipulating the tile.

Although the pen has the same security as the panel - it uses a reference to the PIP security settings -, it was necessary to define this region. The hierarchy of transformations allows thus to define a local pen coordinate system, which is transformed by tracker updates in world coordinates.

Moving the tile close enough to the table region, the tile snaps onto the surface, indicating a region transition. Releasing the pen button confirms this transition. As the security definition of the table region enables viewing for each user, the tile texture becomes visible and every player can see it (Figure 10 shows the sequence for a typical region transition).

Although privacy is supported by information layering, the PIP incorporates simply by its physical properties an additional kind of privacy. Players may hold the panel in a position, so that tiles are visually hidden to other players, however they do not contain any useful information in all other players customized views.

6.1.4 Private Help

Additional information is provided to players on private layers (Figure 9). Our Mah-Jongg implementation has a semantic controlled help system supporting visual feedback for understanding, and additional geometry for group manipulations. Forming groups of tiles results in the creation of a *shovel* placed under the grouped tiles. Text on the handle of the shovel indicates valid combinations. This kind of help-information can be used for more than status reporting. The owner of the group can also move the whole group by picking the handle of the shovel and moving it to another region. On the table shovels get displayed to every user for easier calculation of scores at the end of the game.

7. CONCLUSIONS

We described a multi-user augmented reality gaming system for board games which maintains social communication and provides private space within the common game to allow individualism. We demonstrated our concepts by the implementation of the Mah-Jongg game, to test our setup and interaction techniques.

Basically our tests show, that it makes fun to play in such an environment! Even very inexperienced users find out very fast how the game is to be played without any additional information on the general use of the system. The shortcomings in precision of the hardware can be adequately compensated by tools like the described snapping mechanism.

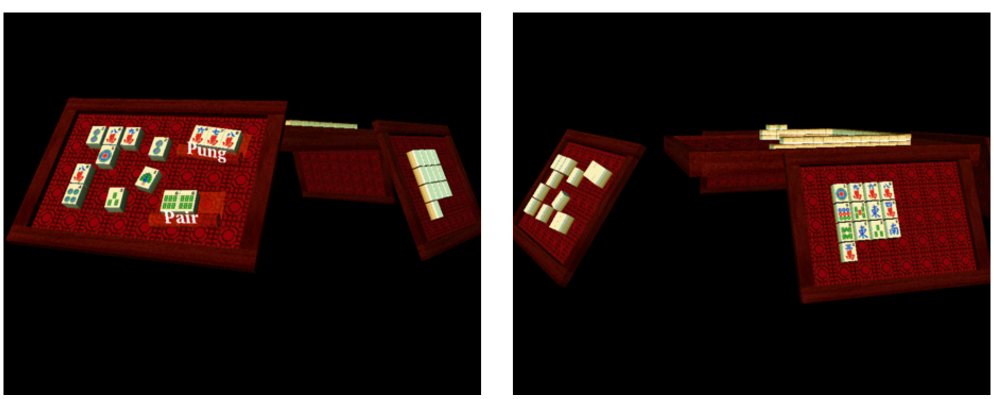

Figure 8. Privacy: The same situation during the game seen by two different players. Characters on the opponent's tiles, as well as help information is not shown. To keep a consistent augmentation, tiles are not hidden entirely from other players. This would cause unexpected popping artifacts during region transition actions.

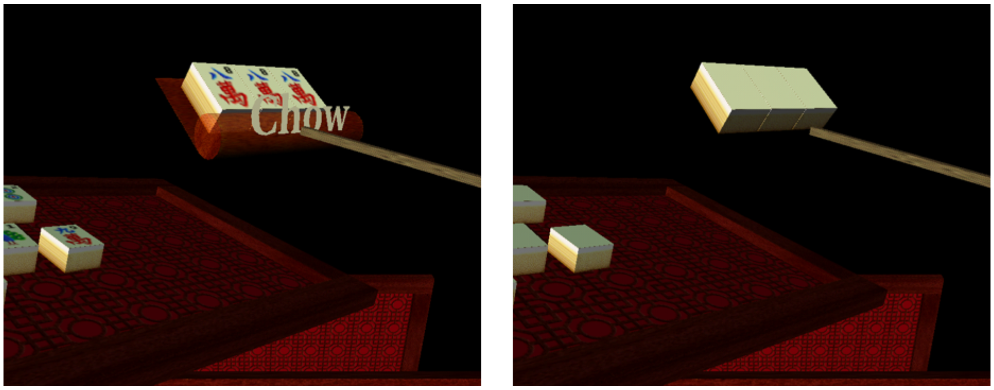

Figure 9. Private help expressed as a *shovel*. This additional handle helps to manipulate groups together. The same situation with another players rights is shown on the right side.

Especially in regions where there are only a few snapping faces, the snapping conditions can be very generous, so that objects snap onto each other even if they are not very close. This allows rapid actions to be performed giving the game a dynamic character.

If object manipulation is restricted to drag-and-drop like actions, as presented in our examples, snapping is a powerful substitute for collision detection in virtual environments. As objects snap to each when they are near, interpenetration happens very seldom and only in cases which do not disturb the user. Moreover, in some cases collision detection would rather hinder easy manipulation of objects.

Private help is seamlessly integrated in our approach, and is a natural extension of the game itself. In our tests, users have turned on private help all the time.

Compared to the original game, the augmented version is capable of changing the whole mood of the game with additional 3D graphics. A simple, yet effective demonstration of this concept are the “augmented hats” in the figure on the first page. Furthermore the flexible setup supports different types of games and the private help system is an additional feature of the proposed system. We also think that inspired by our setup, game developers can find entirely new gaming ideas in 3D.

Currently our system consists of a commercial available standard hardware, but we see a good chance that the system could be produced as a console game for multiple users in future. A ”game-box” could contain game-server and rendering clients as well as the tracker source. If the game-box is placed on a table, players can sit around that table holding their pen and panel, the game-board can also be displayed on that table which also gives a haptic feedback.

As our hardware-setup is lent from a scientific-visualization system it is only natural to project our interaction techniques and privacy concepts back to that application area. We think that an adapted version of our system-concepts could enhance scientific visualization applications. Multiple scientists discussing one common visualization are able to switch on and off individualized information they like to see personally. Simplifications induced by the gaming domain could be removed to support other type of applications. We see a great potential for our setup to be used also for 3D education- and presentation-systems in the near future.

8. ACKNOWLEDGMENTS

This work was funded by the Austrian Science Foundation (FWF) under project no. P-12074-MAT and the City of Vienna. Special thanks to Toni Fuhrmann for providing the tracker-implementation and Dieter Schmalstieg for valuable discussions. Further information about this project can be found at: http://www.cg.tuwien.ac.at/research/vr/gaming/.

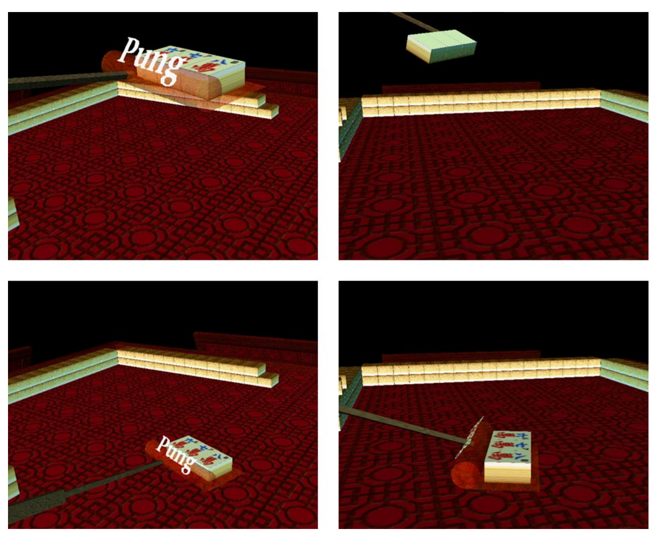

Figure 10. A player is placing a “Pung” combination onto the table (left column). A different player is observing this action (right column). This sequence gives an example for a region transition with changing privacy.

9. REFERENCES

[1] Agrawala, M., Beers, A. C., Bernd Fröhlich, Pat Hanrahan, The Two-User Responsive Workbench: Support for Collaboration Trough Individual Views of a Shared Space, *SIGGRAPH 97*, August 1997, pp. 327-332.

[2] Bier, E.A., Snap-Dragging in three Dimensions, *Computer Graphics, Symposium on Interactive 3D Graphics '90*, 24(2), March 1990, pp. 193-204.

[3] Billinghurst, M., Weghorst, S., Furness, T. III. Shared Space: An Augmented Reality Interface for Computer Supported Collaborative Work. *Proceedings of Collaborative Virtual Environments Workshop'96*, 1996.

[4] Bolt, R., A., "Put-That-There": Voice and Gesture at the Graphics Interface, *Computer Graphics (SIGGRAPH '80 Proceedings)*, 14(3), pp. 262-270, July 1980.

[5] W. Broll. VRML: From the Web to Interactive Multi-User Virtual Reality. *Proceedings of the GI Workshop on Modelling, Virtual Worlds, and Distributed Graphics*, Bonn, 1995.

[6] Bukowski, R. W., Séquin, C. H., Object Associations - A Simple and Practical Approach to Virtual 3D Manipulation, *1995 Computer Graphics, Symposium on Interactive 3D Graphics '95*, April 1995, pp. 131.-138.

[7] Carlsson, C., Hagsand, O., DIVE - A Platform for Multi-User Virtual Environments. *Computers & Graphics*, Vol. 17, No. 6., 1993, pp. 663-669.

[8] Cruz-Neira, C., Sandin, D.J., De-Fanti, T.A., Surround Screen Projection-Based Virtual Reality: The Design and Implementation of the Cave, *Computer Graphics (SIGGRAPH'93 Proceedings)*, Vol. 27, August 1993, pp. 135-142.

[9] Fa, M., Fernando, T., Dew, P.M., Direct 3D Manipulation Techniques for Interactive Constraint-based Solid Modeling, *EUROGRAPHICS' 93 Conference Proceedings*, 12(3), 1993, pp. 237-248.

[10] Feiner, S., MacIntyre, B., Seligmann, D., Annotating the Real World with Knowledge-Based Graphics on a See-Through Head-Mounted Display. *Proceedings of Graphics Interface'92*, 1992.

[11] Fuhrmann, A., Löffelmann, H., Schmalstieg, D., Collaborative Augmented Reality: Exploring Dynamical Systems, *Proceedings of Visualization 97*, October 1997, pp.459-462.

[12] Gottschalk, S., Lin, M.C., Manocha, D., OBB-Tree: A Hierarchical Structure for Rapid Interference Detection, *SIGGRAPH 96, Conference Proceedings*, 1996, pp 171-180.

[13] Hand, C., A Survey of 3D Interaction Techniques, *Computer Graphics Forum*, 16(5), Dec. 1997, pp. 269-281.

[14] Juster, N.P., Modelling and Representation of Dimensions and Tolerances: a Survey, *CAD*, 24(1), Jannuary 1992, pp. 3-17.

[15] Krüger, W., Bohn, C.-A., Fröhlich, B., Schüth, H., Strauss, W., Wesche, G., The responsive workbench: A Virtual Work Environment, *IEEE Computer*, July 1995, pp. 42-48.

[16] Mine, M.R., Brooks Jr. ,F. P., Sequin, C. H., Moving Objects in Space: Exploiting Proprioception In Virtual-Environment Interaction, *Computer Graphics, SIGGRAPH '97 Conference Proceedings*, 1997, pp.19-34.

[17] Pausch, R., Snoddy, J., Taylor, R., Watson, S., Haseltine, E., Disney's Aladdin: First Steps Towards Storytelling in Virtual Reality, *Computer Graphics, SIGGRAPH '96 Conference Proceedings*, 1996, pp.193-203.

[18] Smith, G., Mariani, J., Using Subjective Views to Enhance 3D Applications, *Proceedings of VRST'97*, 1997, pp. 139-146.

[19] Sutherland, I., A Head-Mounted Three Dimensional Display. Fall Joint Computer Conference, *AFIPS Conference Proceedings*, 33, 1968, pp. 757-764.

[20] Szalavári, Zs., Gervautz, M., The Personal Interaction Panel - A Two-Handed Interface for Augmented Reality, *Computer Graphics Forum (Proceedings of EUROGRAPHICS'97)* 16(3), September 1997, pp. 335-346.

[21] Szalavári, Zs., Eckstein, E., Gervautz, M., "Studierstube" – An Environment for Collaboration in Augmented Reality, *Virtual Reality – Research Development and Applications*, 3(1), 1998, pp. 37-48.