Interactive Cards
A game system in Augmented Reality

João Alexandre Coelho Ferreira, Instituto Superior Técnico

Abstract: Augmented Reality can be used on innumerous topics, but the point of this work is to explore the relation between Augmented Reality and table-top games. To explore this relation, we created a table-top game where several key concepts of collectible card games and miniature games come together – Interactive Cards. To support this table-top game, we developed a generic gaming system, configurable by XML files, to create a virtual representation of the table-top game environment. The system was developed using Virtools to create the virtual world and ARToolKit to detect the user interactions with the game components. The prototype was evaluated by key users and it got positive results, although the users agreed that some characteristics could be improved.

Keywords: Augmented Reality, table-top games, cards, Virtools, ARToolKit

1. Introduction

Nowadays, games play a big part in our lives, partly because of computer games. But a lot of people prefer other kinds of games: table-top games, social games, board games, etc. Common to all these people is their goal: to have fun. We play games to amuse ourselves, be it in front of a computer trying to finish that last level or surrounded by friends playing the latest novelty in board games. What we propose is a way to bring together computer games and table-top games.

The biggest advantage of computer games it’s their ability to immerse the player in its virtual world. The player has total freedom to move around the virtual world and interact with it and getting feedback about that interaction. On the other side, it is only possible to interact with the virtual world with devices that create a barrier between the computer game and the player.

Table-top games, on the other hand, allow the player to directly interact with the game environment, normally composed by various types of bits and pieces. The table-top game’s environment is not a dynamic environment because all the components are static. Nothing happens without an action from the player. Besides that, these actions are just physical interactions with the game components and, usually, are not representative of what is being executed (for example, an attack between to characters is just a mere dice throwing).

So, we have these two types of games in opposing corners: computer games are more immersive than table-top games but have limited interaction; table-top games allow the player to directly interact with the environment but are unable to make the player feel has a part of that environment.

This is where Augmented Reality comes into play. Given its properties, it is possible to create new ways of interaction with computer games and also “give live” to table-top games. This paper presents a proof of concept: a table-top game will be created and a computer system will also be created to breathe life to the table-top game.

2. Interactive Cards system

2.1 Overview

This is a generic game system using cards and tokens to represent characters or other playable items and is to be used in conjunction with ARToolKit to represent the battle in a computer screen, it can be used for a variety of games ranging from medieval fantasy to science fiction space warfare.

The characters represented in the cards are shown in the computer screen, substituted by theirs 3D models, in a real time manner and can be seen attacking and moving in the playing field area. Buildings, terrain and objects can also be represented by their 3D models whenever placed on the playing field. Players can choose or activate characters by touching the card and in the computer screen we can see the maximum range we can move them or the possible targets for their attacks.

Most of the is configured using a XML editor were you can edit all game variables, such as battle points objective, the playfield and the all the cards (characters, support and terrain).

2.2 The game

This game can be played with the ARToolKit and in its standalone version. This means that the game rules work alone but the ARToolKit enhances the gameplay and speeds it up as the computer makes all math calculation and shows the result as well as enhancing the game with computer graphics and animations. The AR version supports network play via TCP/IP direct connect were two arenas can play against each other in remote locations.

Three types of basic cards are defined:
• **Character**: these cards represent a monster, hero, a battleship or even a tank, each of these cards will have a set of attributes representing its skills.

• **Support**: these cards represent special conditions or events that happen throughout the game such as thunderstorms or hurricanes, it may represent also equipment or spells the characters can have or cast. There are two types of support cards, Equipment and Special. Support Equipment cards can be played by a character to enhance its abilities and support Special cards represent spells or other special abilities or feats performed by the characters.

• **Terrain/Object**: objects or even buildings that stand on the playfield, they can influence the play by blocking, giving bonus, generating mana/resource points.

Each player plays with a team of characters that they choose at the beginning of the game comprised of character cards whose point cost total the battle points objective and a deck of 40 support cards.

Each character has 5 skills and each skill has a value, the higher the value the better the skill.

• **Movement (MOV)** – distance in cm that the character can move.

• **Armor (ARM)** – resistance to combat damage or magic damage the character has.

• **Health (HEA)** – the amount of damage that it can withstand before dying

• **Close Combat (CC)** – the skill of the character in close combat.

• **Ranged Combat (RC)** - the skill of the character in ranged combat.

Besides the characters skills, each character has three generic attributes.

• **Resource points (RP)** – representing the wealth or resourcefulness of the character, i.e. the influence of the character to get/buy equipment.

• **Special points (SP)** – representing the magical power of the character, the ability to call for arcane and magical resources or even cast spells or perform feats.

• **Points cost (PC)** – representing the cost of the character, the higher this cost the more experienced and skilful should be the character.

During each turn, each player has several phases to go through:

• **Disengage**: restore the initial state of all the characters controlled by the active player.

• **Engage**: choose which action each character will perform on the current turn. There are two kinds of actions: movement and action.

• **Movement**: all characters that received a movement order in the engagement phase can move in any direction as long as its MOV allows.

• **Action**: all characters that received an action order in the engagement phase can execute an attack (melee or ranged) or use a special ability, represented by a support card.

### 2.3 System setup

A webcam will be needed to play the augmented version, to allow the pattern tracking. The following figure shows how the system should be setup:

![Interactive Cards system setup](image)

For the augmented version, the distance between the webcam and the playfield must be measured carefully. The system is optimized to a maximum distance of 50 centimetres between the webcam and the playfield. Beyond that, the system becomes unable to track the patterns correctly, resulting in strange behaviors.

### 2.4 System architecture

The system was developed using Virttools, so it shares a lot of its architecture. The following figure illustrates the system architecture.

![Interactive Cards system architecture](image)

The system runs on a single composition (.cmo file). This file contains all the necessary scripts for the system execution, except the 3d models that represent game objects. The 3D models are stored in.nmo files, one for each model, so that can only be loaded when
needed. All the composition requests are managed by the Behavioral Engine.

2.4.1 ARToolKit Integration

To better understand the ARToolKit integration, let’s first look at the script that controls its behavior:

The building block **Camera Video to cvImage** implements some OpenCV functionalities, namely the image capture from the webcam. Through the cvImage to texture building block, each image captured by the webcam is transformed into a texture so it can be applied to a 2D frame, to allow the stream to be visualized by the player. Each image will also be used in the **ARToolkit tracker** building block to track the patterns. To ease the communication between ARToolKit and Virtools, an array was created. This array stores all the info necessary for the pattern tracking. Virtools will only read this array and it’s the ARToolKit that changes the information in it:

- Path to the file that represents the pattern.
- Size of the pattern.
- Object referential.
- Pattern visibility.
- Pattern position.
- Pattern orientation.
- 3D object associated with the pattern.
- Virtual camera (Virtools element that represents the webcam).

2.5 System implementation

Although it is only one composition, the system can be divided in three modules: virtual version, augmented version and multiplayer augmented version. Common to all these three versions are the configuration files.

2.5.1 Configuration files

One of the goals of this system is that its content can be easily configured. It is not the goal of these files to allow rules configuration. Three files were created to configure all the common settings on all versions and a forth one was created for only the augmented versions.

2.5.2 Game Manager

Given the rules being action oriented, it is easy to implement all those actions in Virtools. Below we describe what kinds of scripts were created:

- **Configurations files** (4 scripts) – reading all the information from the configuration files and storing it so that it can be used by the game manager
- **Turn phases** (5 scripts) – one script for each phase and each script controls all the actions that can be executed on each phase.
- **Virtual version UI** – controls all the virtual version UI elements.
- **Support cards** (7 scripts) – seven types of support cards were defined and for each one of those types, a script that executes the action was created
- **Multiplayer** (4 scripts) – control all the multiplayer process, from the server creation until the distributed objects.

To control all these scripts (and some more…), a global script was created, called Game Manager. This manager is responsible for the correct execution of the game since it decides what scripts to activate and when.

2.5.3 Multiplayer

Virtools has a module called Virtools Multiuser Server that was used for the development of the multiplayer component. There were two possible ways to implement the multiplayer component: a dedicated server or an embedded server. Since the game is designed for two players, there is no need to create a dedicated server so we opted for the embedded one.

Several scripts were created to control the multiplayer component. Has the server is embedded in the composition and, therefore, in one of the players computer, these scripts be able to distinguish between server and client. Again, it’s the Game Manager that controls this process.

The communication between the two computers is made through distributed objects and network messages. The distributed objects broadcast their state automatically to all clients connected to the server, making simple updating the position, orientation and visibility of all virtual objects to all clients. Network messages are used for all the other information that distributed objects can’t handle (action, animation synchronization, etc.)

3. Results

The end result was a prototype of the system composed by the following components

- Virtual version
- Augmented version
- Augmented version with multiplayer
These three versions are all independent and playable. Our efforts were mostly directed to the augmented version, so it is the most polished one. That doesn’t mean that the other versions are not playable. Just that there are some details that were only implemented in the augmented versions, like the range check.

The following figures show the final prototype:

![Figure 4 Virtual version. In this example each player controls three characters. We can also see the cards the active player has in his hand (bottom) and the information panels (right).](image)

The virtual version is player in a single computer, in hot seat format, where a player executes all his actions and steps aside so his opponent can perform his. This procedure continues until one of the players is defeated.

![Figure 5 Augmented version. In this example, the active player (right) controls two characters, a unicorn and a knight. His opponent controls a dragon. The active player is casting a fireball though his unicorn.](image)

This version doesn’t have any kind of UI elements, except some information about the active player phase or error messages.

### 3.1 Prototype evaluation

The final prototype was tested with key users, in order to evaluate its quality and acceptance.

Each user was allowed to fully explore the system and play any number of games he wanted. At the end, it was asked to the users to answer a small survey to evaluate some aspects of the prototype and to score the system as a whole. The following graphic displays the results the scoring.

![Figure 6 Global scoring of the Interactive Cards system prototype, 10 being the highest score](image)

**Strong points:**

- Cards (layout, information, size)
- 3D models
- Virtual version’s user interface

**Weak points:**

- Battlefield size
- System setup
- Game feedback to the player

As the results show, the prototype was well accepted by the users, but there is still room for improvements, as we can see by its weak points.

### 4. Conclusions

We managed to do what we wanted: to bring together table-top games and computer games. In the end, the Interactive Cards system can be played in three different versions: real version, using the cards; virtual version, using the computer; and the augmented version, using both the cards and the computer.

The system is generic and it can be easily adapted to any gaming environment, though it requires some time to do this. It is necessary to change the configuration files, bringing in the new elements for the new environment.

The users’ evaluation of the prototype was very positive. Of course there are still some aspects needing some improvement and some functionality to develop, but globally, they were pleased with the final result.

### 4.1 Future work

Aside from developing some of the system functionalities that were left out, like for example, the virtual version’s multiplayer component, the next step will be to improve the prototype to a final version. For that, an extensive testing phase will be needed to discover and correct possible bugs. Also it would be interesting to test alternatives to ARToolKit and see if
this framework is really the best solution for this system.

5. References


3. ARToolKit Home Page (http://www.hitl.washington.edu/artoolkit/)
