Interactions Methods for Immersive Magic Systems in LARP through Augmented Reality

Zach Laster

University of Helsinki
Department of Computer Science

November 22, 2013

Interested in trying the prototype? Go here: `www.cs.helsinki.fi/u/laster/2013/fall/AR/larpcaster.apk`

The goal of the project is to craft a an interaction mechanic to enable magic in a fantasy roleplay game through the use of Augmented Reality (AR).

In order to accomplish this, a very basic game with a strong focus on magic was designed in order to set the baseline standards to acheive and test against.

The primary target of this system will be live-action roleplay games. Such a game is typically played in real-time and is very much bound to the real world. This implies several limits upon the allowable game mechanics, particularly with movement of characters and the general lack of availability of real magic.

Live-Action Roleplay (LARP) is effectively the live variation of pen-and-paper roleplay games games. They can range in setting, from historical recreations to high fantasy, and the rules between any two games can vary largely, especially on details of the execution of combat. For this case study, the focus will be on high fantasy settings, where magic plays a major and obvious role.

A large portion of LARP-style games, like most roleplay games, focuses on social interaction between players.

This differs from most computer-based games, which tend to inadvertently eliminate social situations from games.

One possibility to help bring virtual gaming back to a true social setting is to utilize *ubiquitous computing*[Björk et al., 2001] in games. A combination of AR and live-action gaming seems an ideal blend on that path.

Since we are focusing on the illusion of magic, the game rule set should be almost entirely magic dependent.

In order to simplify interaction with the virtual world, the magic system utilizes "wands" to cast spells. With this in mind, the two strongest influences on the game design will be Dungeons and Dragons and Harry Potter style magic. [1]

Examples of spells a player could cast include `Fire Bolt`, `Flame Wall`, `Teleport`, and `Invisibility`. Each of these spells has fairly unique requirements from the game system.

[1] The magic system of the Harry Potter lore is terrible and illogical, at least from a game standpoint, but the interaction itself is still good as a reference point.

`Flame Wall` is the simplest spell, requiring only handling collisions with elements that move into it. `Fire Bolt` requires real-world collisions and physics, which is a step up from the Flame Wall. `Invisibility` requires the ability to track an individual as they move and to apply an "invisibility effect" to them, regardless of pose. Lastly, `Teleport` is an advanced mechanic with multiple possible solutions, such as making the user invisible and intangible (to in-game objects) for the duration of transit or making the game world freeze until the player relocates.

`Teleportation` and `Invisibility` are staples of most magical game systems. While AR cannot directly produce any form of `Teleportation`, it may be able to create effects that facilitate illusions of `Teleportation`, such as making the user invisible or pausing the game. `Invisibility`, on the other hand, is a very localized illusion, and might be possible using Diminished Reality. In-game objects are also possible to represent using AR. This means that spells which would produce visual effects directly, such as `Fire Bolts` or summoned creatures, are readily possible.

In order to make the interface system as extensible as possible, we use hand-held "wands" to draw points in space.

These points can then be recognized as some form of *Glyph* or symbol. This recognition draws from normal character recognition using drawn points.

We can additionally handle direct gestures such as shaking or moving the wand in particular patterns.

For simple spells, a single recognized Glyph can produce a spell result, using on location and orientation. The simplest example is casting a `Fire Bolt`, which will originate from the center of the glyph and travel in the direction the Glyph is "facing"[2].

More advanced spells could use larger, more complex Glyphs, or even multiple Glyphs chained together. The latter would require that the user have control over when they are creating particles. Ultimately, this does not affect the core premise of the interaction.

[2] Assuming Glyphs are essentially 2D characters drawn in the air, this would give them a normal which is the direction the user was facing when they drew it. Alternatively, the normal may face the user and the inverse normal would be the direction away. In any case, the Glyphs are highly orientable.

The prototype allows users can create virtual particles in the air. It is a networked system, which means that other users can connect to see the same particles or to create their own.

Ideally, wierless wands and visors would be available for such a system, but a smartphone can serve as well. The prototype is built for Android devices, though only some are supported. I used a Samsung Galaxy S2 for testing.

The primary challenge that must be overcome is to accurately and precisely position the devices in space.

Most positioning systems will not work for this context. GPS is the highest precision positioning built into most phones, and it can be off by ten meters when performing at its best. Relative positioning using the in-built sensors such as the accelerometer have strong tendencies to drift, meaning that motion would be exaggerated and the particles would drift.

SensorFusion, which is applying all of the native sensors to provide more advanced information, is better, but still produces drift over time.

Wi-Fi based positioning can be utilized, but this is still error prone and requires very established Wi-Fi hotspots and the system must be trained to the network topology.

Robotics has used Simultaneous localization and mapping (SLAM) for decades in order to build maps of a robot's environment, allowing it to traverse and navigate that environment.

We can apply a similar approach using a camera built into a handheld device to construct similar mappings. This will allow us to precisely and accurately determine the location of the camera in space.

This approach has some drawbacks, particularly that the map must be built in advance and is slow to make. However, it also means that the prototype can be used anywhere and does not require any devices beyond the used smartphone.

The PointCloud API was used to provide SLAM mapping mechanics for the API. The API was originally developed for iOS and is being ported to Android. Because of this, it does not support very many devices yet and is not fully functional.

A device running the prototype can determine it's location and orientation in space. This allows us to create particles releative to a devices position within a virtual space.

The particles can be rendered in 3D with location relative to the camera, so they actually appear to be at the locations at which they were created.

Once a user has a map built and saved, they can load it to host a network session. Others can join this session.

When a user first connects they recieve the map from the server host as a file, which is then opened locally. This allows the devices to orient themselves within the same space without knowing the location of other devices.

Devices can then send and recieve broadcast point creation signals. Each device notifies all of the connected devices in the session of every point they create, and are similarly notified.

It is up to each device to track the points of each player. Every player is given a randomly generated UUID so that other devices can identify them.

For the prototype, the AllJoyn API was used. This API allows devices to communicate to each other via a bus system and proxied interfaces, meaning the devices are not aware of the network communication.

The results for the prototype are promising. It succeeds in it's goals, allowing points to be created and rendered with high precision in space.

There are some issues to be addressed, particularly with the positioning system. PointCloud works well for it's goals, but does not scale well to covering an entire room. It also does not extend to allow for full geometry modelling at present, meaning it does not aid in physics applications.

There are also many things which can yet be applied to the prototype, such as recognition for the Glyphs.

We built a basic game design, devised an interaction method to enable that game design in an augmented reality setting, and then implemented a prototype to test the premise of that method.

Overall, the prototype was a success and indicates strong possibilities for future work. There are numerous and exciting directions in which to take this system.

As technologies improve, various aspects of the system will become easier to accomplish, particularly device location. Various end-goals will also benefit and become more feasible in an AR setting.

1. More advanced or usable positioning and/or approaches
2. Glyph recognition
3. Game Mechanics (Such as spells or basic game content)
4. Virtual element occlusion or collisions against real objects
5. Porting to other devices, such as headsets

Questions?

Links

1. Report: www.cs.helsinki.fi/u/laster/2013/fall/AR/laster_larpcaster.pdf
2. Slides: www.cs.helsinki.fi/u/laster/2013/fall/AR/laster_larpcaster_presentation.pdf
3. Prototype: www.cs.helsinki.fi/u/laster/2013/fall/AR/larpcaster.apk

13th Lab.
Pointcloud.
[Online; accessed 03-October-2013; `http://pointcloud.io`].

Björk, S., Falk, J., Hansson, R., and Ljungstrand, P. (2001).
Pirates! using the physical world as a game board.
In *PROCEEDINGS OF INTERACT 2001*, pages 9–13.

Herling, J. and Broll, W. (2010).
Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments.
In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 207–212.

Klein, G. and Murray, D. (2007).
Parallel tracking and mapping for small AR workspaces.
In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan.

Sachs, D. (2010).
Sensor fusion on android devices: A revolution in motion processing.
[Online; accessed 03-October-2013; `http://www.youtube.com/watch?v=C7JQ7Rpwn2k`].

Shala, U. and Rodriguez, A. (2011).
Indoor positioning using sensor-fusion in Android devices.
PhD thesis, Kristianstad University, Sweden.

Wetzel, R., McCall, R., Braun, A.-K., and Broll, W. (2008).
Guidelines for designing augmented reality games.
In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, Future Play '08, pages 173–180, New York, NY, USA. ACM.