WeQuest: A Mobile Alternate Reality Gaming Platform and Intelligent End-User Authoring Tool

Chinmay Barve and **Sanjeet Hajarnis** and **Devika Karnik** and **Mark O. Riedl**
School of Interactive Computing
Georgia Institute of Technology
Atlanta, Georgia 30308
{cbarve, sanjeet, karnik, riedl}@gatech.edu

Introduction

An *Alternate Reality Game (ARG)* is an interactive narrative that uses the real world as a platform. An ARG layers a fictional world over the real world such that, as a player moves through the real world, a narrative structure plays out. Typically a puppetmaster monitors players from remote, making adjustments to the narrative arc as necessary (analogous to a *branching* storyline). Many ARGs utilize actors planted throughout the area in which the game is being played to interact with players in real time. Geo-location enabled mobile devices, such as smart phones, are increasingly being used to track ARG players and to deliver content from the fictional world.

Although ARGs are growing in popularity, they are significantly limited in several ways that prevent ARGs from being utilized by mainstream game players. First, there is a substantial cost in running an ARG if puppetmasters and confederates must be hired, limiting the number of players that can participate in an ARG at any given time. Second, ARG storylines reference real world geographical locations and landmarks in the real world to advance the narrative structure. Consequently, a particular instance of an ARG is fixed to a specific region of the real world. For example, a conspiracy story to be played in New York City cannot be played in London without substantial re-authoring. As a result, participating in an ARG is not a common entertainment activity for most.

In this paper, we introduce a suite of technologies designed to overcome scalability issues of ARGs by automating the puppetmaster and by encouraging end-user authoring of new location-specific storylines. Our suite consists of:

1. A game engine that runs on a geo-location enabled mobile device and plays the role of the puppetmaster.
2. An authoring tool that supports end-user authoring of novel ARG storylines.
3. An AI tool for *translating* ARG storylines to new locations.

We have developed a flexible game engine, called *WeQuest* that can play out an ARG storyline. *WeQuest* is a game engine in the sense that it does not have a fixed storyline, but can be used to download and play any storyline. *WeQuest* uses the geo-location features of the mobile device to step through a storyline. An authoring tool makes it easy for amateurs to construct novel ARG storylines for the areas in which they live.

Finally, to promote sharing of end-user authored storylines, we provide an intelligent tool that can translate an ARG storyline set in one location to one that can be played in a disparate location. With this tool, a storyline that references locations in midtown New York City can be translated to uptown New York City, Atlanta, or anywhere else. The AI translation tool attempts to find analogues between locations in the original storyline and the new area in which the game is to be played.

The Game Engine

In *WeQuest*, storylines are represented by a dependency graph. Nodes correspond to events and reference specific GPS coordinates. Arcs between nodes represent dependencies such that an event cannot fire unless (a) the player is within a certain radius of the event's GPS coordinates and (b) all preceding events have completed. Nodes can also require players to take certain actions. For example, an event may require the player to active a virtual inventory item or to engage in a dialogue with a virtual through a dialogue tree on the device.

A dependency graph is a basic technique for managing lock-and-key style game play. Unlike finite state machines, dependency graphs can support branching, partial ordering of events, and parallelism. Note that we do not attempt to alter the storyline based on player actions as a true puppetmaster might; any branching must be pre-authored into the dependency graph.

The *WeQuest* game engine supports multiplayer cooperative and competitive play. Dependencies are synced with an online database so that team members can unlock different parts of the storyline independently of each other. GPS locations of players are periodically synced with each player's instance of the game engine so that a map interface can show each team member locations.

The Authoring Tool

The *WeQuest* authoring tool can be accessed through a Javascript enabled web browser. It supports visual authoring

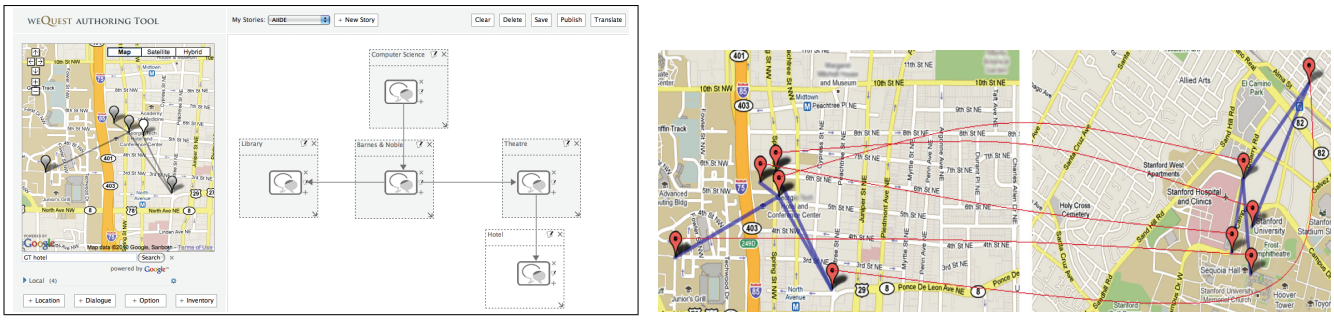

Figure 1: A portion of the *WeQuest* authoring tool (left) and an illustration of location translation (right).

by allowing event nodes to be created and dragged around a workspace area. To facilitate authoring of geo-location material, the authoring tool is integrated with Google Maps™. Authors can view the storyline in its dependency graph form or superimposed on a map. New event nodes can be created by querying locations through the map interface. Figure 1(a) shows a portion of the authoring tool being used to create a branching storyline. Once the new storyline is complete, it can be "published" to a server for immediate download to mobile devices running the game engine.

AI Location Translation

Location translation is a process whereby the locations referenced in one storyline are mapped to analogical equivalents for a new area. If a storyline cannot be played by a particular user because the storyline references locations that are not in the user's vicinity (neighborhood, city, etc.), the user can invoke a translation tool through which an analogous storyline is created to be played in the user's geographic area.

Analogical reasoning is a difficult problem, typically requiring large amounts of well-formed knowledge. However, analogies can also be found using statistical information retrieval through search engines such as Google™.In the current version of *WeQuest*, authors are required to provide tags for each event node indicating the salient features of that location. These tags are sent to the Googe Maps™API along with the the GPS coordinates of the new area to which the storyline should be translated. For each event location, some number of results are returned that can be considered analogical equivalents.

We use a modified dynamic programming algorithm to select which of the possible alternatives retrieved should be used for each original location. The dynamic programming routine uses a heuristic that compares the overall lengths of the storylines, individual segment lengths, and the shape of the paths as indicated by angles between each segment. Shape is especially important to avoid unnecessary cutbacks or to preserve paths that look like loops. The dynamic programming algorithm has been modified to work on branching structures by implementing the following assumption: an optimal decision for a node is dependent only on optimal decisions for any prior nodes, according to the dependency graph. See Figure 1(b) for graphical depiction of a storyline that has been translated.

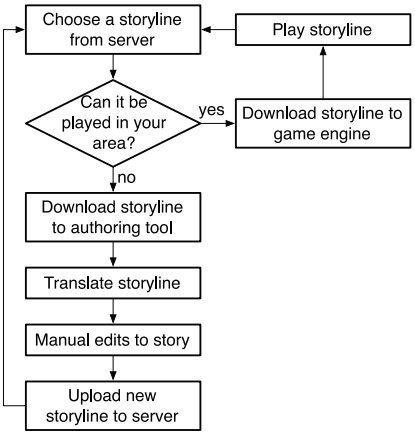

Figure 2: Flow of end-user authoring with location translation.

Statistical translation is not guaranteed to work 100% of the time. If mistakes are made, the user has an opportunity to review and manually edit the new storyline. The philosophy behind this human-in-the-loop approach is that it is still easier to make a few simple edits than to author from scratch. Once the storyline has been corrected, it can be re-published for others in the same area to play. Figure 2 shows the end-user authoring process with AI location translation.

Conclusions

Alternate Reality Games enable games to be played away from the desktop or console and require tight coupling between real and fictional worlds. ARGs have not achieved mainstream status because of their dependency on human puppetmasters and on references to specific geological locations and landmarks. One way to overcome the scalability issues of ARGs is through massive amounts of content creation – lots of storylines to be played in lots of places. In this work we attempt to make it easy to author and share storylines through intelligent authoring tools that can translate a storyline from one place to another.

Acknowledgements

The work presented here was supported by Humana, Inc.