Master Project Class, Spring 2011
Final Report

z00m!
A Multiplayer Game With A Purpose

Team: Gryffindor

Tilman Dingler (tvdingler@usfca.edu)
Nabeeh Ghanem (nghanem@usfca.edu)
Juliet Rubin (jcrubin@usfca.edu)

Project Website: http://www.yahoo.tilmanification.com

Project Sponsor: Yahoo! Local with Jon Rahoi ((jon@rahoi.com)
Supervisor: Prof. Dr. Jeff Buckwalter (buckwalter@usfca.edu)

Table of Contents

1. Abstract

The objective of this project is to close the gap between news and the geographical location they are most relevant to. Machine learning algorithms and natural language processing can do a bulk of work by analyzing texts for semantics and geographical entities. However, there are tasks that are easier and more accurately done by humans. So, when it comes to allocating news items to relevant neighborhoods, human input is highly valuable. Thus, we want to create a mechanism in form of a game to motivate people to provide this data.

Luis von Ahn has coined the term Games with a Purpose (GWAP) where human players are encouraged to provide valuable data while playing and being entertained [4]. We took this as a motivation to create the game *zoom!* where players gradually narrow down geographical areas with relevance to specific news items.

2. Introducing *zoom!*

Our team was given the task of creating a computer game that would accurately collect data about articles' geo-locations in order to determine where they were most relevant to. However, we were given the freedom to design this game. After brainstorming and coming up with a couple of game ideas, we decided upon the game *z00m!.*

z00m! encourages a 2-player collaborative effort to identify geographical locations where certain news articles take place. By narrowing down a location the players receive points according to the level of detail, but only if they both agree on a location. The game is played simultaneously and in each round a news item is presented to both players. They then have to decide what location on the map this article could be relevant to (country, state, city, region, neighbourhood). If they agree, they move into the next detailed map. If they disagree, another news item is presented and another round is started. During the game, we collect the geo-tagged information about news items, so that we are able to collect multiple votes over time and thus verify the correct location an article is most relevant to.

Rules of the Game
When a player starts a new game, he or she is paired with another player. A timer is set to 4 minutes, and the first article is presented along with a pre-processed tag-cloud (Fig. 1). The tag cloud is used to facilitate the faster information intake and includes the most important keywords in the article.

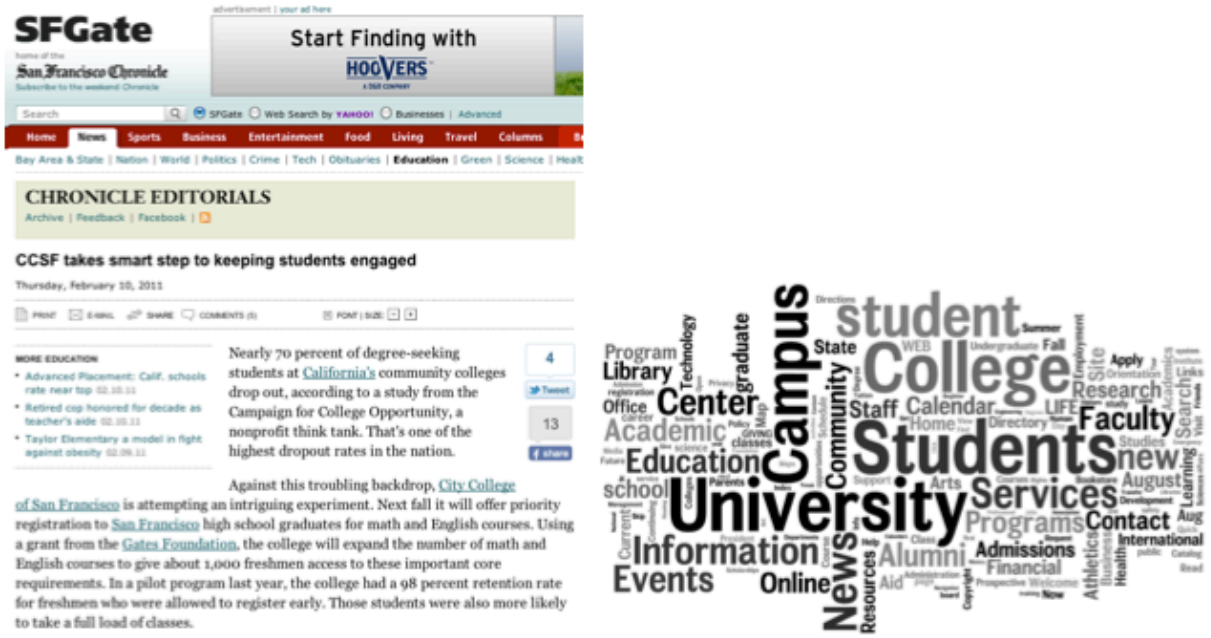

Fig. 1: presentation of news item: full article and tag-cloud.

Both players are presented with the same news item as well as with a map, as shown in Fig. 2.

There are 5 levels of details for the map view:

1: countries
2: states
3: regions
4: cities
5: neighbourhoods

On the first level both players have to select a country to which the news item is most relevant to. By clicking on the map, a country is selected. A double click submits the selection and the map zooms in to the selected country. Feedback is provided to let the user know if her partner has made her selection, to motivate to quickly make a selection.

If both players agree on a location (Fig.4.), the players receive points and move to the next zoom level. The map's level of detail gets more refined the further the players narrow down the location and incrementally more points are awarded for each level.

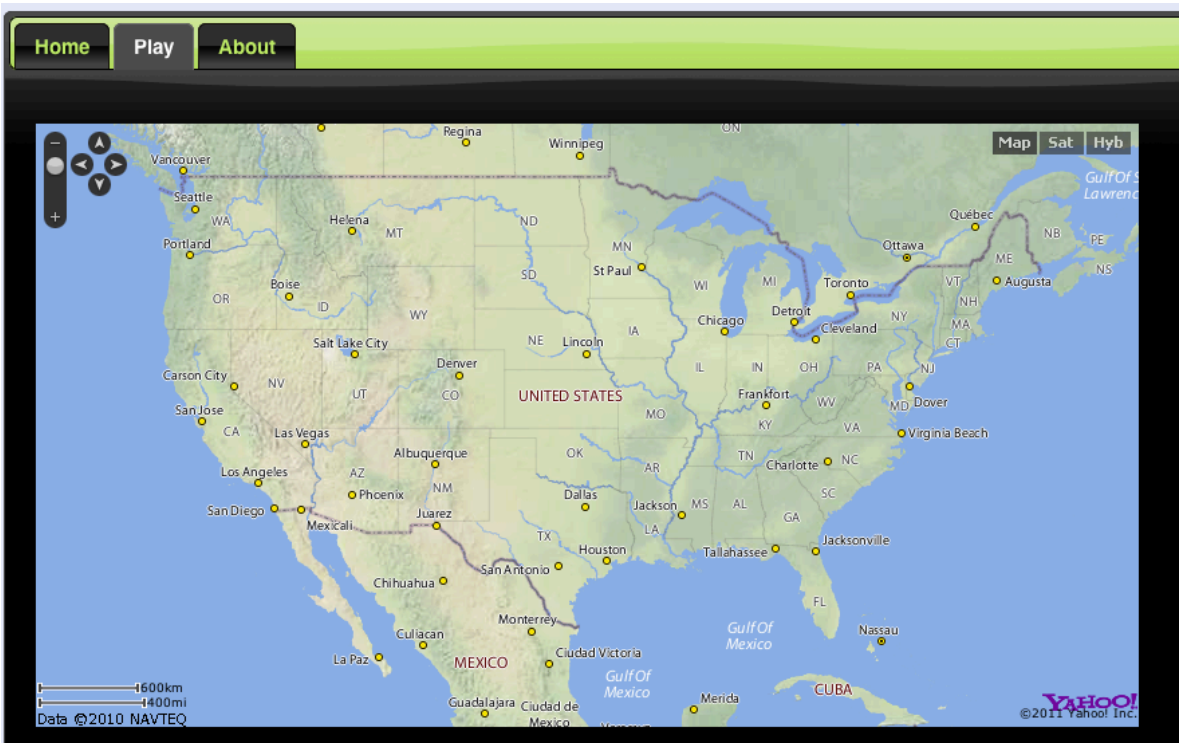

Fig 2.: Interactive map for capturing user input (source: http://maps.yahoo.com).

In case of disagreement, the game moves on to the next article and the map will focus back on the country level. For example, if one player chooses California and the other player chooses New York state, both players disagree and thus do not get any points (Fig. 3).

Fig.3.: Two players selecting different locations results in location disagreement.

After 4 minutes the game is over, a point summary is given, and the players are asked if they would like to play a new game. Limiting the game time naturally terminates the game and also motivates players to tag as many articles as possible.

The beauty of this game is the combination of a fun player experience and the harvesting of useful data in order to geo-tag news items. Another useful aspect is the potential usage

of raw data as well as pre-tagged data. New informational items can be fed into the system just as well as already pre-tagged data for further refinement and verification.

Fig.4.: Two players selecting same locations results in location agreement.

3. Goals

The overall goal is to connect news items with geographical location they are most relevant to (Fig.5).

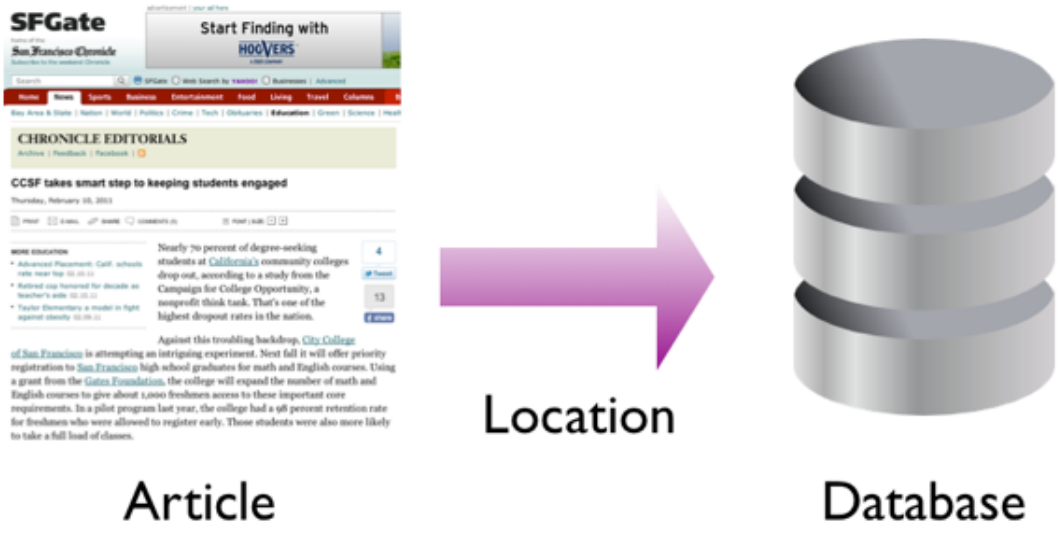

Fig.5: Goal behind the scenes: Feeding article locations into Yahoo!'s database

In order to collect this data we created a Game With a Purpose (GWAP) [4] that motivates users to provide this kind of data.

Therefore we needed a user front end that facilitated user interaction and data input. Double clicks zoom into specific areas. That way the game flow is not interrupted. It is important for a seemingless game experience that there are only few delays due to data reload.

But before we could feed news items into the system they needed to be pre-processed in order to extract the keywords for the tagcloud. The reason behind displaying a tagcloud was that players would not always want to read the entire article, but furthermore skim through it in order to find clues about where this article could be relevant to. Thus, we needed some data mining algorithms that extract most important keywords.

4. Design

Fig.6 shows the general architecture of the interplay between players, server and our databases.

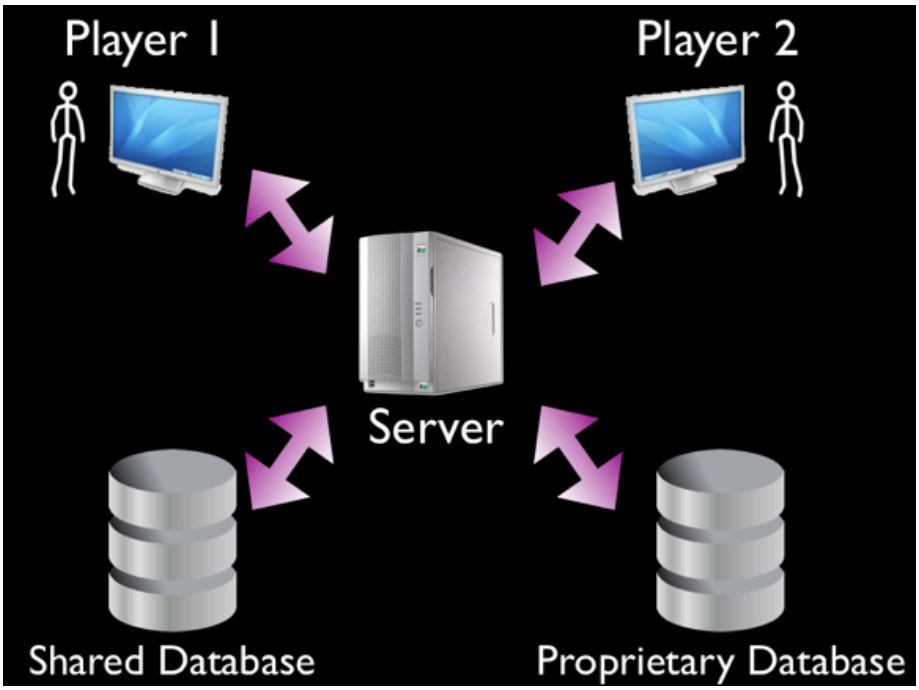

Fig.7: Architecture overview and interplay between players, server and databases.

The news items presented are retrieved from a shared database that we host filled with articles and can be shared with other applications. This database can be filled automatically via crawlers and furthermore contains data items from related tagging projects. A second database (our proprietary database) saves user information, game sessions and additional article information, such as extracted keywords.

One thing to note about the game is that we do not necessarily need any previous information about the article. The tagcloud (Fig.1) is created by automatically analyzing the text, thus combining computer and human computation of texts. However, if previous information about a news item, i.e. from the geo-tagger, is present, the game can be used

to verify or refine this information. At some point it gives the opportunity to start the starting zoom level at a more detailed point. Let's say we already know from previous games or tagging data that an article is relevant to Northern California. Players can directly be presented with a detailed view of Northern California and asked to select the specific region, city and neighbourhood.

This way, the game does not rely on pre-tagged data, but provides the functionality to incorporate such data for further refinement and verification.

Verification takes place through iteratively feeding the same articles to different players. Each time the game is played, a set of articles is being processed. After some time we will have a pretty good guess from the user input where this article is situated, e.g. after 10 congruent location selections. The threshold for verification is to be determined.

The quality of the data is assumed to be quite high thanks to the collaborative game design. Players are randomly (maybe depending on ISP location) paired and thus don't know each other. However, in order to be successful at the game and reach a high score, they need to make rational guesses about the items' locations. Thus, the likelihood for spamming is relatively low and can be easily discarded as outliers once a specific news item has been passed through several game rounds.

The client-side architecture is split into 3 layers, as shown in Fig.8.

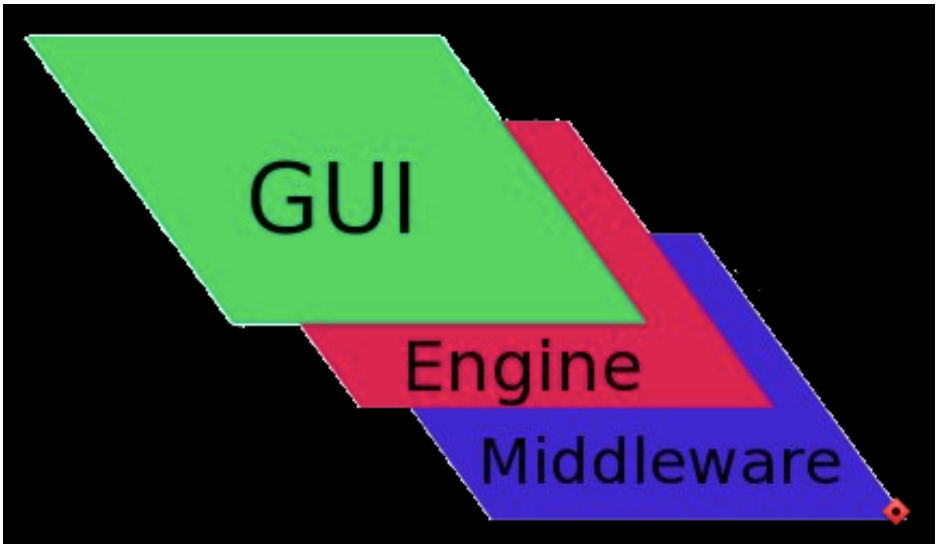

Fig.8: Front-end divided in 3 layers: Graphical User Interface, Game Engine and the Middleware

The first layer, the Graphical User Interface (GUI) allows the player to interact with the game and displays the game state. The underlying game engine receives the user input from the GUI, processes it and forwards it to the middleware which takes care of the client-server communication.

Users interact with the front-end in order to input location data relevant to a news item. Therefore the game runs in a web browser retrieving data such as news items from our server. Important for a compelling game is a smooth game flow. Thus, special

considerations needed to be given to a fluent selection of regions and the zooming into different levels of details.

6. Implementation

We use a combination of Javascript, HTML, CSS and multiple APIs to interface with (such as Yahoo Maps API, YUI3). The front-end is supported by the Yahoo! User Interface 3 (YUI3) API [5], which is a JavaScript and CSS library that facilitates UI design. Yahoo! Maps API [6] provides us with map controls to focus the map on a specific location and zoom functionality.

Interactive elements as well as the game logic is implemented in Javascript, an object-oriented scripting language. Between the browser client and the server back-end data is exchanged via HTTP requests. This way, the website never needs to be refreshed, but communication to the server happens in the background. Server responses are encoded in JSON (JavaScript Object Notation), a lightweight data-interchange format.

The server that controls feeds the articles, matches up players and synchronizes game states, is implemented using the NodeJS framework [7], an event-driven framework that allows coding in Javascript.

For the databases we used MongoDB, an open-source, document-oriented database written in C++ [8]. The database holds the data items, such as articles, keywords, locations and user account information.

When a new article is fed into the database, we analyse its content via a word frequency algorithm. We calculate TF-IDF (term frequency–inverse document frequency) scores, a statistical measure used to evaluate how important a word is in an article in order to extract the 10 most important keywords. These keywords build the basis for the tagcloud which is displayed to the players alongside the article to facilitate getting an overview of the article's content.

Splitting the development of z00m! into to the 3 big areas of front-end, game engine and back-end, we were able to assign team member responsibilities and divide tasks adequately.

7. Testing

During the development process, we constantly tested new functions and modules via Unit Testing. During Unit Testing parts of a program are tested separately to make sure the individual parts of code are working correctly. Unit testing generally takes place during the development phase of a project and works best when each test case can be tested independently of each other.

Throughout our project we have used Unit Testing when we wanted to integrate a new aspect of the project into our code. One example of this is when we integrated a feature that finds the geo-location of coordinates. We first wrote a javascript file that tested if this

feature worked and then added it to the project. If we had instead tried to add it to the game as we were coding this feature, any problem that arised would have been a lot more difficult to pinpoint because we wouldn't have known if it was a problem with the new code or a problem with the already written code. Using Unit Testing cuts down on the time it takes to write a new feature and it also helps avoid bugs by checking for errors before integrating it with the main code.

Another technique we used for testing and evaluation is Paper Prototyping. Paper Prototyping is a technique that helps developers during the design process to meet user requirements and expectations. It includes paper sketching and drawing of the interface to build a prototype or model of a design that can be tested in very early stages of the design process. Paper Prototyping is simple and can provide valuable early feedback for interaction and UI design. We created a couple Paper Prototypes to test our UI and to view different ideas before integrating one into our game (Fig.9).

Fig.9: Paper Prototype we used to test the User Interface early in the design process

When we had our first prototype finished, we Usability Testing in order to identify usability issues and evaluate the game flow. Usability Testing is a technique used to evaluate a product by testing it on users. We conducted usability tests with our prototypes and used

our findings to iteratively improve the design. We tested our project on teammates, other developers, and end users(non-developers) to get well rounded feedback.

Before our final release, we applied Stress Testing in order to test the server's capability of supporting multiple games at the same time. Stress Testing is a technique to determine if a project is reliable even when its capabilities are pushed. We used a minimal form of Stress Testing to make sure that our project would run correctly even when many players were playing simultaneously. We did this before our SLS talk to make sure the game would work correctly when the audience was asked to play.

During Integration Testing individual software modules are combined and tested for compatibility as a group. Naturally these modules have been successfully unit tested before they are evaluated according to their interplay with other system parts. The goal is to test the interface between several modules (especially when developed by different parties) to ensure that they are correctly interacting with each other.

z00m! potentially will be integrated into current Yahoo! Local services. The final step of *z00m!* going live will be a reality check conducted by the Yahoo! Local team. They will do System Testing to verify if the application as a whole works and does what it was designed to do. *zoom!* will be adjusted to fit into their product portfolio. However, this part of Integration Testing will be beyond our project's scope and will be handed over to the Yahoo! Local team.

8. Results and Deliverables

From putting together the team and coming up with the game idea until the final prototype that allows multiplayer to log on and play the game, 14 weeks passed. We split these week into work packages as table 1 outlines.

In the following we will walk the reader through the flow of the game:

When a user wants to play the game, she logs in by providing username and password. Alternatively, there is a registration for new users (Fig.10).

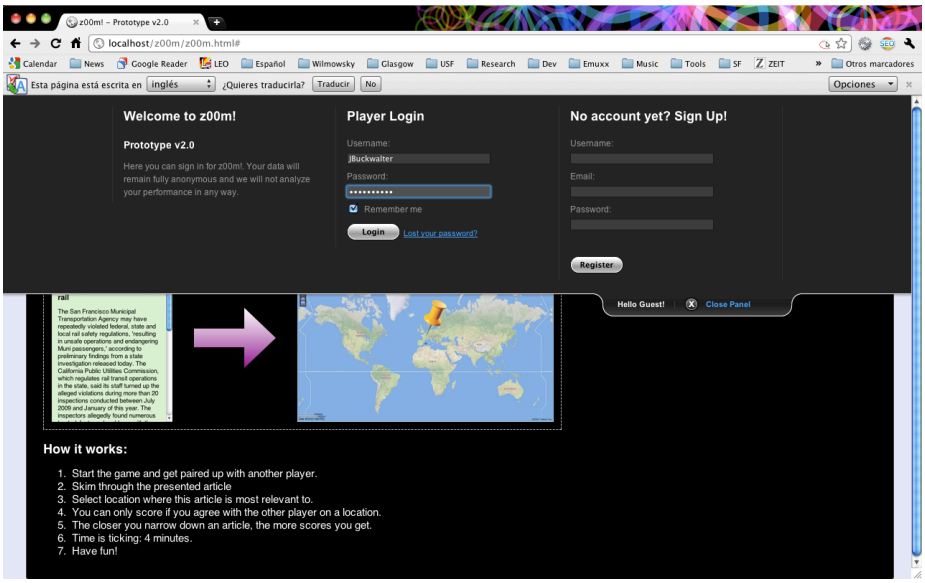

Fig.10: User login and registration

Timeline	Tasks	Deliverable
Week 4	Finish specification document, getting started on JavaScript, Yahoo! GeoPlanet API, Yahoo! Map API and YUI 3.	Specification document
Week 5	Quick 'n dirty front-end design, collect feedback from sponsor, start implementing front-end functions	1st Prototype: Front-end
Week 6	Database and server setup, data exchange tests	
Week 7/8	Feeding articles into database, data processing, keyword extraction	Database with real data
Week 9	Midterm presentation preparation	Revised prototype, presentation slides
Week 10 / 11	Designing the game engine including score functions and session pairing	
Week 12	Refining front-end with HTML5 and CSS3, coming up with test cases	Test-plans
Week 13	Interfacing Geo-Tagger with Geo-Game. Functional and data integration	Geo-Tagger and Game API
Week 14	Testing the fully functional Geo-game. Inviting multiple players for test sessions, data collection and observations	Final presentation draft
Week 15	User testings, debugging	
Week 16	User testings, debugging, final presentation preparation	Final presentation due
Week 17	project wrap-up, final documentations	Final documentation & deliverable due

Table 1: Timeplan, work packages and deliverables

Instructions on how to play the game are given on the start screen of the website (Fig.11).

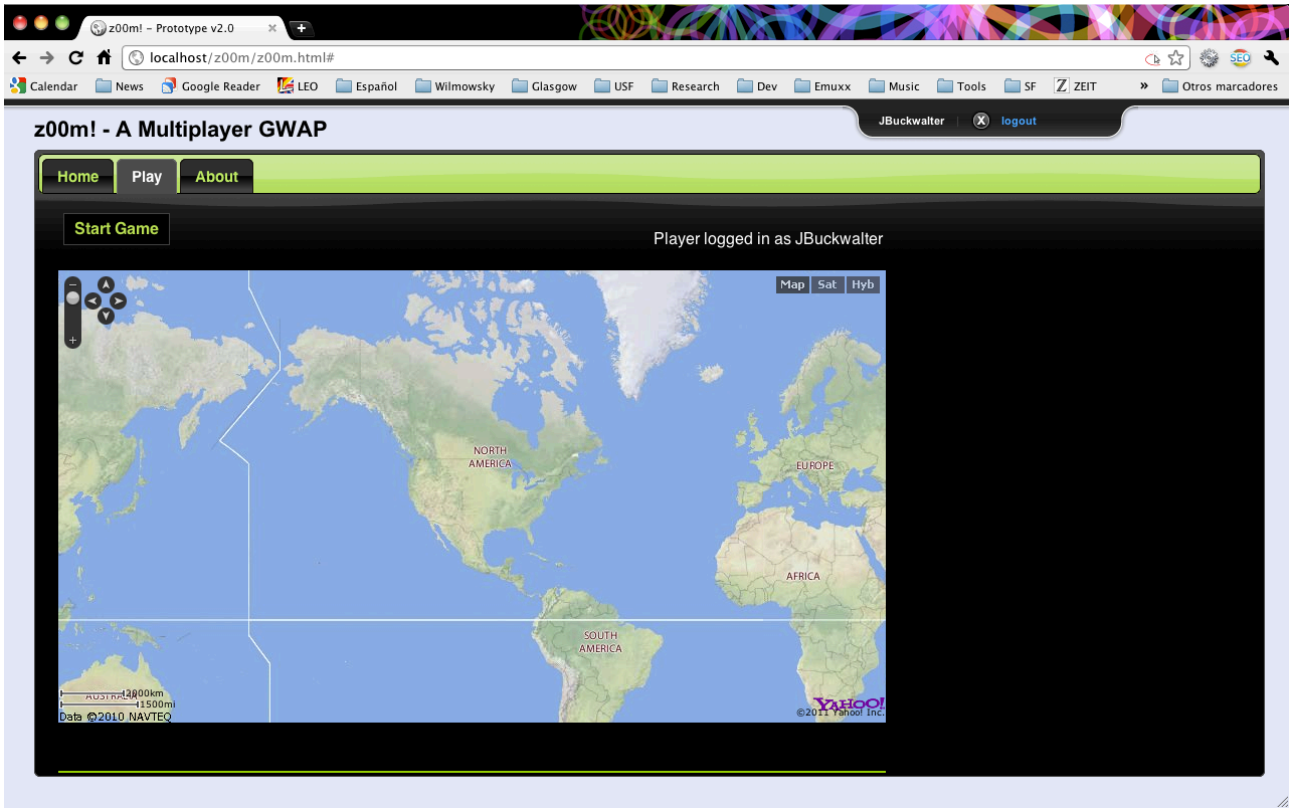

Fig.11: Start screen of the game's website

The 'Play' tab takes the user to the game screen where she can start a game. Once started the player is being paired up with an opponent. After matching with another player, the game is started and the timer is set to 4 minutes. A map is shown as well as an article and the tagcloud containing the most important words of the article (Fig.12). The players are asked to choose a location by clicking on the map. A single click selects a location, a double click confirms and submits the player's selection.

While playing the game, if players disagree on a location, no scores are given and another article is presented. In case of agreement, scores are granted and the map is set to the next zoom level.

Players can make use of the 'Pass' button they feel like they are not given enough information about the article to determine a location. In case of passing, the other player is notified and a new article is presented.

When the timer is up, the final score is presented, which is the same for both players. From here, players can continue playing the game by starting a new game session containing a new set of articles.

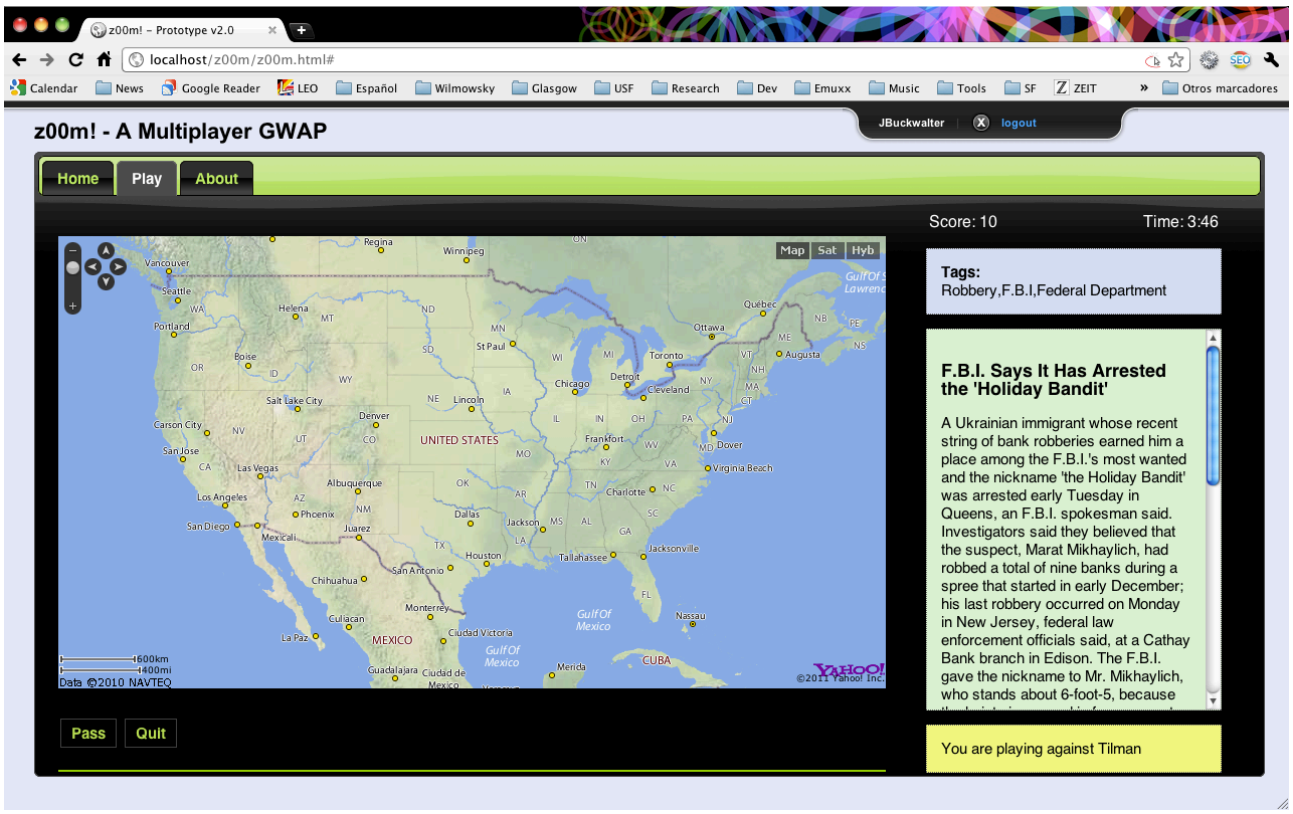

Fig.12: Game start: Map, article and tagcloud are shown.

9. Future Work

In the future we can extend this project by adding further functionalities to it, such as:

Session recording: Game sessions can be recorded by keeping track and saving the kind of decisions player make on a specific article. This way, a game can be presented to a single player whenever there is no other player around, for example. Theoretically, there should not be any noticeable difference between playing against another human player and playing against a recorded session.

Location selection: One of the tricky parts of the projects was the way how map regions are selected by the players. Currently feedback is only given after the player clicks on the map. However, hovering over the map could be accompanied by a feedback of the region the cursor is hovering over. We were also considering shapes to be drawn over certain map regions in order to facilitation selections. Another alternative would be allowing the players to draw their own shapes.

Article data: In the current version of the prototype we use raw articles that are only pre-processed by our TF-IDF score algorithm. However, it is possible to feed pre-tagged articles into the database in order to verify their locations.

Outlook: We are hoping to test this project out on Yahoo! Local's website or on the Yahoo! Game website. By making it available to a potentially huge number of people we could collect valuable feedback data to prove the game design and adjust the game.

10. Conclusion

z00m! is a multiplayer Game With A Purpose that motivates users to connect news articles to geographical locations. Therefore, we used current technologies, such as Javascript and NodeJS to implement an entertaining game for our project sponsor: Yahoo! Local. In the following we want to conclude with some fundamental learnings and take-aways:

Team work can be rewarding and is great when working on a large project. Our team had regular team meeting, about once a week, where we discussed our progress and goals for upcoming weeks. We also gave each other feedback on our code and talk about challenges we were currently facing and possible solutions to these challenges.

We had several coding days where we met in the labs and worked on our projects side by side. This was a great way to bond as a team and easily get input on what each of us were working on at the moment. We also used email as a form of communication in between our meetings whenever needed. We used google documents when writing our paper assignments for the class. Google documents gave us the ability to simultaneously work on the the same document and ask each other questions through the instant message feature. We were able to combine our code easily and frequently using the svn our sponsor set up for us.

We learned about many technologies including YUI3 and NodeJS (which our sponser suggested we should use). We attended events in the Bay area where creators of NodeJS would speak in order to improve our understanding of the framework. Through class lectures and our group meetings, we were able to improve our understanding of effective team work.

We met with our sponsor every couple of weeks and were given input of our current prototype and different techniques we could use for our upcoming work. We went to Yahoo in the middle of the semester to present our game idea and progress and were given valuable feedback from the Yahoo local team. In a final presentation at Yahoo!'s headquarters in Sunnyvale we will present our final results, hand over the program code and discuss next steps that need to be taken in order to successfully launch *z00m!* on Yahoo!'s websites.

Special thanks to Prof. Dr. Buckwalter for coaching us throughout the semester and providing us with valuable tools that facilitated sponsor communication, our presentation skills and team work.

Further thanks to Jon Rahoi for giving us valuable feedback during each of the development stages, helping us avoid 'fake work' and keeping us on track and focused.

11. References

[1] Dix, A. and Finlay, J. and Abowd, G.D.; Human-Computer Interaction, ISBN: 0130461091, published in 2004, Prentice Hall

[2] Constantine, L.L. and Lockwood, L.A.D.; Software for use: a practical guide to the models and methods of usage-centered design; ISBN: 0201924781, published in 1999, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA

[3] Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design. Technical Report IBM RC-9265.

[4] Von Ahn, L. and Dabbish, L., Designing games with a purpose, in Communications of the ACM, 2008 Vol. 51, #8, ACM

[5] http://developer.yahoo.com/yui/3/

[6] http://developer.yahoo.com/maps/

[7] http://nodejs.org/

[8] http://www.mongodb.org/