University of Patras

Department of Electrical and Computer Engineering

Electronics and Computer Division

Design and Development of end user development platform for mobile games using social media

Diploma Thesis of

RUBÉN CÉSAR LLORENTE MUÑOZ

STUDENT OF THE UNIVERSITY OF VALLADOLID

SUPERVISOR: PROF. N. AVOURIS

DIPLOMA THESIS NUMBER:

PATRAS, JULY 2012

To my parents and my family for your support and understanding all these years,

to Elisabeth, for your encouragement day by day, you'll be always in my heart,

to my friends, because when I need you, you are always cheering me up,

to my tutor, Christos Sintoris for trust me during the developing of the project,

to all the fantastic people I have met during my stay in Greece.

Thank you all!!

CERTIFICATION

It is certified that diploma thesis with the title:

DESIGN AND DEVELOPMENT OF A GAME EDITOR FOR CITYSCRABBLE APPLICATION INTO THE GOOGLE+ HANGOUTS

Of the sudent of the University of Valladolid

Llorente Muñoz	Rubén César
(Surnames)	(Name)

Was presented in public at the department of Electrical and Computer Engineering of the University of Patras, Greece on July 13th, 2012.

The supervisor	The head of the electronics and computers division
N. Avouris	E.Housos

Abstract

In the present project I purpose the design and development of a collaborative game editor for the Android CityScrabble application based in the Google+ Hangouts Platform.

CityScrabble is a kind of Scrabble game where several players can play in a real location. This kind of game is called location-based. The game can be located in all around the world. Each player will be able to choose a topic about which he plays in every moment, and about this topic he must answer a group of hints (called elements in the editor). If the player answers correctly a hint, it will add the score value of the hint to his score and all the players will be informed about it in real time.

Many technologies have been used in the development process. First of all and because the platform is Google+ I tried to use only technologies directly supported by Google (Google + API, Google Maps API, Hangouts API, Google AppEngine, Google Datastore) but Google AppEngine and Google Datastore gave me problems with the Python server so I decided to use MySQL as database management system. The queries are requested from PHP files allocated in the university server. Moreover, to allocate the Javascript, CSS and XML files needed I have used the server appspot provided by Google.

As I said before, the editor is collaborative. This means that more than one person can be editing the same game at the same time. So it is for this reasons that Google+ Hangouts is the platform; to achieve this broadcasted edition. The hangouts allows to the participant to share the application, documents (by Google Docs), the screen, etc. And also provides videoconference and chat system.

If I may return to the subject of the editor, it has been developed trying to be compact (due to the conditions of the Hangouts), grouping all the information in a little space and trying also to be intuitive for the user. Despite of this and due of the information compression in the interface, the final user may need a guide to introduce himself in the edition. This guide is included in the chapter 5 of the diploma thesis.

I would like to finish this report remarking the utility that an editor might have in a game as CityScrabble, allowing to create games of any subject in any place of the world. I would like to remark also, that this editor has the potential to be improved adding much more features and allowing to edit other different kind of games for CityScrabble.

CONTENT INDEX

FIGURE INDEX

1. INTRODUCTION

DESIGN AND DEVELOPING OF A GAME EDITOR FOR CITYSCRABBLE APP

In this first chapter I will describe briefly the main aspects of the project. First of all I will point out the motivation I had to do this work. After that, I will explain the fixed objectives and requirements established. In the third point I will explain what is CityScrabble and the profits of a game editor for it. After this point, I will explain why we have chosen the Google+ Hangouts as the platform for the game editor. Finally, I will talk about the chapters of this thesis including a brief description of each one.

1.1. Motivation

Location-based games are games which the game play somehow evolves and progresses via a player's location. Thus, location-based games almost always support some kind of localization technology, for example using GPS. "Urban gaming" or "street Games" are typically multi-player location-based games played out on city streets and build up urban environments.[i]

In our case, CityScrabble is a real time network multi-player game, for mobile devices based on the Android platform. In this game, several players can play at the same time in a specific location, which can be diverse.

The fact that the location is variable gives CityScrabble the power to be played in all around the world. But to play in all around the world it is necessary to create specific games for each location.

At this point we face a problem: there are uncountable locations and uncountable kind games to be created separately. The best option is to create a game editor that allows to create any kind of game in any place of the world. This option releases the developers of the problem of develop any kind of games and give the user the possibility to design its own game.

1.2. Objectives

The objective of this project is to develop an application that works into the Google+ Hangouts able to create and edit games for the CityScrabble Android application.

In order to achieve this objective the application will be firstly developed as a normal website and after, it will be adapted to the Google+ Hangouts. The main reason to work in this way is to work quickly since upload the complete application all the times a change is made to the Hangouts would burden the development.

The first important requirement of the editor is to achieve the storage of data in the hangouts in order to save the data of any game to edit or modify it later. Another important requirement for the game editor is to allow live edition and multiple users editing in the same session (one of the reasons to choose Google+ Hangouts).

Finally, the game created must be a JSON or XML file in order to be read easily by CityScrabble.

1.3. CityScrabble

CityScrabble is a real time network multi-player game, for mobile devices based on the Android platform, using the programming language Java Android

The application is a kind of Scrabble game in which several players can play in a real location. This game is a kind of game called location-based. The location of the game can be diverse, it can be from museums to entire cities. Each player will be able to choose a topic about which he plays in every moment, and about this topic the game will show a group of hints (elements). If the player goes to the correct location associated to the hint, it will add the score value of the hint to his score and all the players will be informed about it in real time.

1.4. Google+ and the Hangouts

1.4.1 Google+

Google+ is the Google social network. It is a set of different services like video chat or group messenger providing lots of tools to manage our personal relationships and our web interaction. Like in other social networks we have a personal profile where we can share links, ideas, photos or videos but the way how we interact with this things does Google+ different.

Circles. This is the ground of Google+. The main idea of this social network is that everybody has friends but not with all of them we want to share the same information. Therefor, when we add new info we select the circles (where we include our friends, family, etc.) which can access to this information.

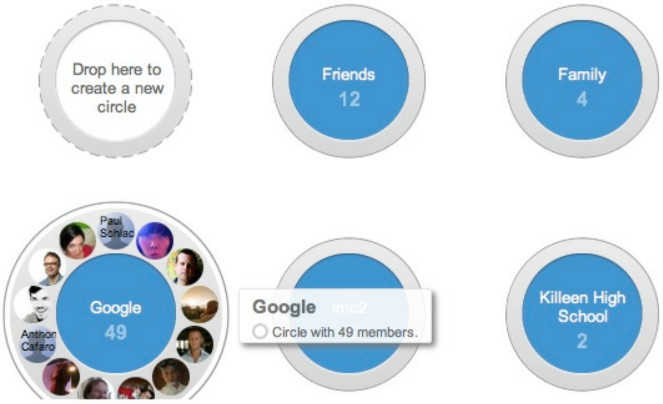
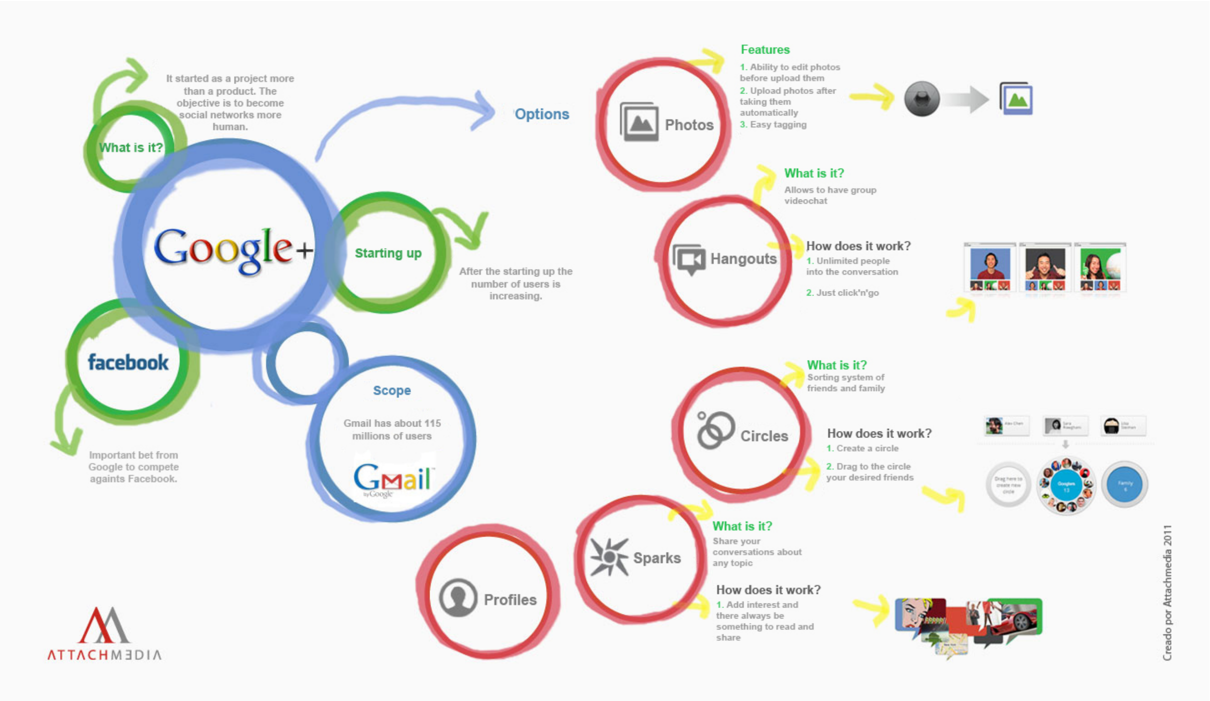


ILLUSTRATION 1. GOOGLE+ CIRCLES

+1 button. With this button a website can be recommended from anywhere in the Internet. After clicking +1 button an item of the site will appear advising about the site to our friends in Twitter, Facebook or another social networks.

Sparks. Sparks works like +1 button. It allows to recommend and to see link recommendations of our friends. But, furthermore we can use it to see interest. For example, if somebody wants to receive information about sports, he can add it as an interest and see a list with the added interest which is updated constantly.

Mobile. The mobile area allows us to share our location to our circles. Moreover, with instant upload we can upload pictures in the same moment that we take them (after giving the permission to Google+). Another application of the mobile area is huddle.



It started as a project more than a product. The objective is to become social networks more human.

After the starting up the number of users is increasing.

Important bet from Google to compete againts Facebook.

Gmail has about 115 millions of users

1. Ability to edit photos before upload them
2. Upload photos after taking them automatically
3. Easy tagging

Allows to have group videochat

1. Unlimited people into the conversation
2. Just click'n'go

Sorting system of friends and family

1. Create a circle
2. Drag to the circle your desired friends

Share your conversations about any topic

1. Add interest and there always be something to read and share

ILLUSTRATION 2. GOOGLE+ DIAGRAM

This is a group chat (similar to Whatsapp or Blackberry Messenger) with the advantage that the groups are done automatically (our circles).

1.4.2 Hangouts

Hangout is the tool for public videochat in Google+ with unlimited number of viewers. These conversations can be recorded and added to Youtube when they are finished. Moreover, the viewers can watch the video but they can't read the chat messages if this messages are not for them. Furthermore, if there are many viewers, the sound can be silenced in order to remove the background audio. In addition, it's possible to share documents using Google Docs or presentation with SlideShare, share the screen of our computer or share applications made for the hangouts.

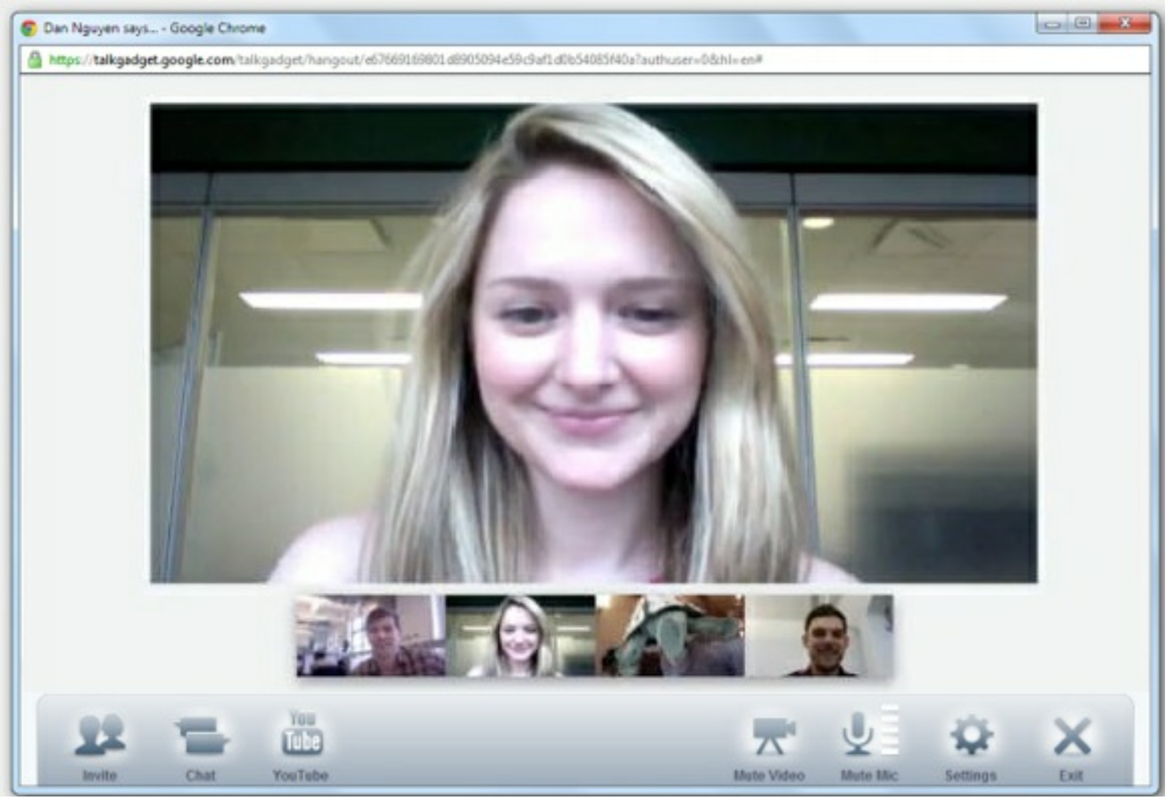

Dan Nguyen says... - Google Chrome
https://talkgadget.google.com/talkgadget/hangout/e67669169801d8905094e59c9af1d0b54085f40a?authuser=0&hl=en#
You Tube
Invite
Chat
YouTube
Mute Video
Mute Mic
Settings
Exit

Illustration 3. Google+ Hangouts

1.5. Thesis organization

The rest of the report follows the next organization:

- **Chapter 2 - Tools used.** In this chapter I will describe the main features of the tools used to do the project and the possible choices.
- **Chapter3 - Design.** Here I will describe how the application has been designed and all the features of the design.
- **Chapter 4 – Development and implementation.** In this chapter I will show the process followed to develop the application.
- **Chapter 5 – Result of an evaluation session.** This chapter will include the conclusions obtained of an evaluation session of the application in the laboratory.
- **Chapter 6 – Manual for the user.** This chapter is a manual for the final user who will create and modify games for CityScrabble.
- **Final conclusions and future possibilities.**

2. TOOLS USED

2.1. Operating System

An operating system is a software which works like interface between hardware devices and the programs of the user in the computer. It is the responsible of management, coordinate activities, do the resources exchanges and being the station of the applications executed in the machine.

2.1.1 MAC OS X

Mac OS X is an operating system developed and sold by Apple included in their computers since 2002. It's based in UNIX and was built over the technologies developed in NeXT between 1985 and 1996.

Mac OS X is based on the Mach core and some parts of FreeBSD and NetBSD implementations were added to the system. While Jobs was out of Apple the company tried to develop a next generation system unsuccessfully. Finally, Steve Jobs went back to Apple and started the project Rhapsody that would be known in the future as Mac OS X.

2.1.2 Linux

Linux is an operating system Unix-based. It appeared in the early 90's due to Linus Torvalds who wanted to improve Minix. Since then, the number of developers interested in create and develop this system has grown at an incredible rhythm.

What Linux strictly is, it is the core of the operating system, which has implemented standing for the actual hardware (USB, printers, networks, etc.). This code includes libraries and utilities distributed under free GPL license.

The main features of Linux are that it is multitask, multiuser, multiplatform, multiprocessing, it has memory protection between process, it stands a lot of kinds of system files, the core stands many network protocols and it allows to share by a network files and printers even with another operating system. The power, stability, price and portability of Linux made it a very good operating system and it has a leader position in the Internet and possibly it will have it in the future, in another fields of computing.

2.2. Web servers

A web server is a computer that stores websites on the Internet and delivers web pages to viewers upon request. This service is referred to as web hosting. Every web server has a unique address, called an Internet Protocol address, that tells other computers connected to the Internet where to find the server on the vast network. Web hosts rent out space on their web servers for people or businesses to set up their own websites, and the web server allocates a unique website address to each website it hosts.

When someone connects to the Internet, his personal computer also receives a unique IP address assigned by his Internet service provider (ISP). This address identifies the computer's location on the network. When he clicks on a link to visit a website, his browser sends out a request to the IP address. This request includes return information and functions like a postal letter sent across town, but in this case the information is transferred across a network. The communication passes through several computers on the way, each routing it closer to its ultimate destination. When the request reaches its destination, the web server that hosts the website sends the page in HTML code to the requesting computer's IP address. This return communication travels back through the network. The computer receives the code and the user's browser interprets the code and displays the page for the user in graphic form.[ii]

2.2.1. Apache

Apache is the most spread out web server in the world, due to its solid implementation. Firstly, Apache were a patches group for the NCSA web server, these patches were made by webmasters for this server and when the development of the NCSA server stopped, some of the webmasters created the Apache group (A PatCH sErver) .

The main features of Apache are:

- **Reliability.** Near 90% of the best servers work with Apache.
- **Multiplatform.** Apache works in almost all the operating systems in the world, so it is almost universal.
- **Standard.** It follows the standards of the market.

- **Extensibilty.** It is possible to add modules to extend the abilities of Apache. Apache has a lot of modules: to generate dynamic content (PHP, Python, Java…), answer to SSL requests, create virtual servers, etc.
- **Programming languages supporting.** Apache works with Perl, PHP and script languages. It works with Java as well and JSP sites and provides support for dynamic content.
- **Support multiple hosts.**
- **Easy to create and manage logs.** Apache allows the creation of log files like the administrator wants having more control of the server.
- **User authentication.** Apache can manage the authentication using DBM databases, making easy the access restriction to some sites.
- **Custom errors.** Apache allows to customize the answer to the possible errors in the server. It is possible to run any script when something happen.
- **Free software.** Apache is a technology with free source code

2.2.2. Microsoft Internet Information Server

This is the Microsoft web server. It's the second server in the market due to it is integrated in Windows Systems and it is very easy to set up. Moreover it is a stable server but with the disadvantage of being only for Windows and being proprietary software.

Its main features are:

- **State detection.** It can detect the state of the work process and recycle them.
- **Support multiple hosts.**
- **Integrates ASP.NET .** It offer to the developers a lot of functionality levels and the quick development of applications.

2.2.3. Sun ONE Web Server

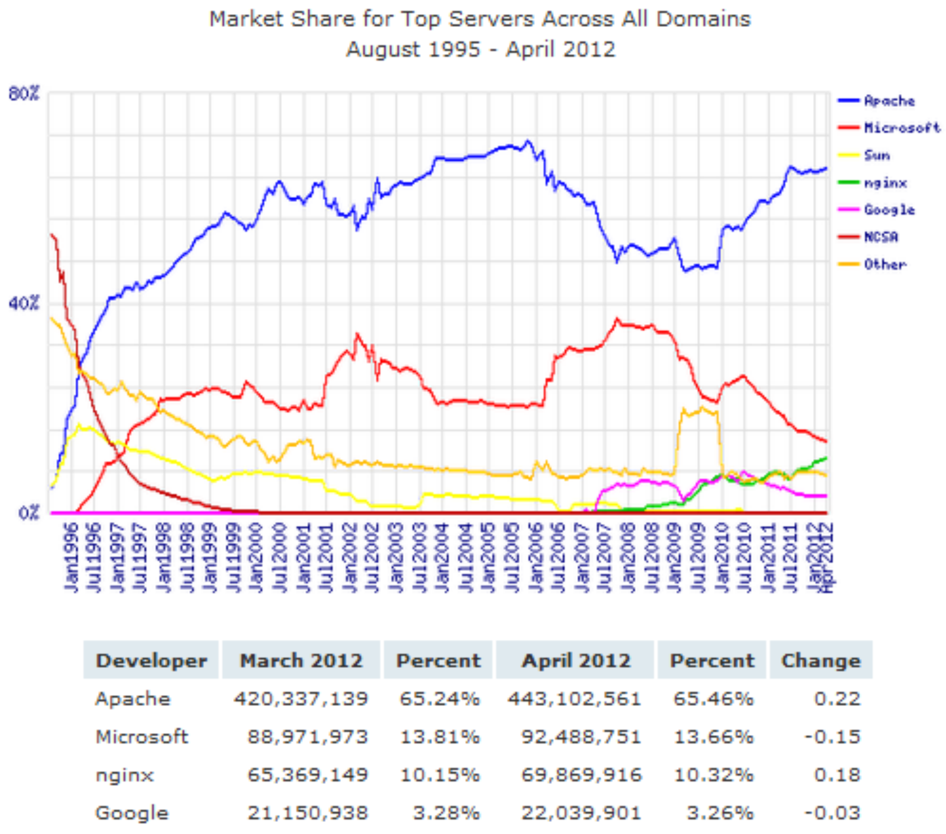
Sun One web server is the server of Sun Microsystems. From version 7.0 started to be free so, it could be a great competitor in the future. It is multiplatform and allows complete support with Java 2 Enterprise Edition (J2EE).

Its main features are:

- **Directory server.** Allows to manage and protect the identity of the network across multiple platforms.
- **SunScreen Firewall.** It is a corporate firewall that allows control of system access and perimeter security in stealth mode.
- **Solaris Containers.** Allow network servers to be partitioned flexibly in separate runtime environments, providing a greater use of resources.
- **Sun ONE Portal Server.** Provides to communities of users, secure access to any application or web service with only a browser.

2.2.4. Comparative of the web servers use

As we can see in the picture, the dominating web server is Apache, Microsoft server is losing users and Google and nginx started 4-5 years ago and they are growing. However, to use a web server or another one depends on many factors, like economical, technical, reliability, etc. In our case we chose the reliability and transparency of Apache during the implementation proccess.



Developer	March 2012	Percent	April 2012	Percent	Change
Apache	420,337,139	65.24%	443,102,561	65.46%	0.22
Microsoft	88,971,973	13.81%	92,488,751	13.66%	-0.15
nginx	65,369,149	10.15%	69,869,916	10.32%	0.18
Google	21,150,938	3.28%	22,039,901	3.26%	-0.03

Illustration 4. Web server evolution by netcraft

2.3. Database management systems

A database management system (DBMS), sometimes just called a *database manager*, is a program that lets one or more computer users create and access data in a database. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, who else may also be accessing the data. In handling user requests, the DBMS ensures the *integrity* of the data (that is, making sure it continues to be accessible and is consistently organized as intended) and security (making sure only those with access privileges can access the data). A DBMS can be thought of as a file manager that manages data in databases rather thanfiles in file systems. In IBM's mainframe operating systems, the nonrelational data managers were (and are, because these legacy application systems are still used) known asaccess methods.[iii]

2.3.1 MySQL

MySQL is a relational database server very fast and strong. Currently is the most widespread DBMS in the Internet.

It is free software, released under GPL. This manager was created to be speed so it does not have many of the characteristics of the largest commercial operators, such as Oracle, Sybase or SQL Server. But mySQL is most suitable for applications that require many reads and few writes and do not require advanced features, such as web applications. Gradually the missing elements are being incorporated into MySQL by internal developments and by free software developers.

The main features of mySQL are:

- **Multiplatform.** It works on almost all operating systems, which makes it almost universal.
- **Large subset of the SQL language.** It is not fully compliant to the standard SQL/92, but includes a large subset of the instructions that it offers.
- **Different storage options.** Depending on whether you want speed in operations or more operations available.

- **Transactions and foreign keys.** MySQL has achieved stability compared to other platforms on the market.
- **Indexing text fields.** Improving performance in searches.
- **Other improvements.** Such as the row-level locking, query cache, the UNION instruction and the delete and multitable update.

2.3.2 Microsoft SQL Server

One of the DBMS created by Microsoft, along with Access. In the first moments SQL Server, was based on the Sybase database system, on which Microsoft had a cooperation agreement. So after a long development, and starting from the beginning, Microsoft's developed their own system.

Main features of Microsoft SQL Server:

- **Ease to install, distribute and use.** SQL Server includes a set of administrative and development tools that improve the ability to install, manage and use SQL Server across multiple sites.
- **Scalability.**
- **Data storage.** SQL Server includes tools for extracting and analyzing summarized data for online analytical processing. SQL Server also includes tools for visually designing databases and analyze data using questions in ordinary language.
- **System integration with other server software**. SQL Server integrates with e-mail, Internet and Windows.

2.3.3 Oracle

Oracle started in the late 70's and early 80's and is currently the most powerful DBMS market.

Its main features are:

- **Multiplatform.** It runs on all platforms, from PC to a supercomputer and many of popular operating systems such as Windows or UNIX systems.
- **Backup and recovery.** Oracle provides hard methods for making backups and data recovery.

- **Space Management.** Oracle provides flexible management of space. Some disk space can be assigned for storing data, and control the assignments having control about how much space should be reserved for future requirements.
- **Openness Connectivity.** Oracle provides connectivity to and from third-party software packages.
- **Integrity** mechanism. Oracle server is also responsible for the integrity of the data. If any fault happens while a user is changing data in a database, it has the ability to cancel any suspicious transactions.
- **Parallel query component.** It is used with multiple CPU computers, and allows query processing to be distributed among multiple server processes.
- **PL / SQL.** With this programming language, procedures can be kept stored in the database.

2.3.4 PostgreSQL

PostgreSQL is a System Manager Database Object-Relational, released under the BSD license(Berkeley Software Distribution in Spanish, Berkeley Software Distribution). It is an alternative to other database systems open source (like MySQL) and to proprietary systems such as Oracle. It has the advantages of open source systems and has achieved the power of proprietary systems. Despite this PostgreSQL has a problem compared with other free DBMS, this is its slowness and the increased system load.

The main features of PostgreSQL are:

- **Views.** It is able to generate views over the tables.
- **Triggers.** Triggers can be associated to the tables increasing their ability to self-management.
- **Multiversion concurrent access.** Tables are not locked or even rows, when a process is writing.
- **Several languages.** It has the ability to host programs on the server in several languages.
- **Table Inheritance.** You can re-use the operating system tables.

At the beginning I was thinking in use PostgreSQL because it is fitted as an standard in Apple systems (I used a Macbook to develop the application) but I realized that there are better solutions.

2.3.5 AppEngine Datastore

The App Engine Datastore is a schemaless object datastore with a query engine and atomic transactions. It offers two possibilities : The Datastore Python API and the Datastore Java API.

The python interface includes a rich data modeling API and a query language similar to SQL called GQL. In the other hand, the Java SDK includes implementations of the Java data objects interfaces and the Java persistence API. Both provide scalable storage for web applications paying special attention on query performance and reads. An application creates entities where the data values are stored as properties of an entity. The application can perform querys over entities. All queries are pre---indexed for fast results over very large data sets.

App Engine provides two options for data storage :

- ○ **Master/Slave Datastore**. This option offers strong consistency for all reads and querys, the disadvantage are periods of unavailability during data center issues or planned downtime.
- ○ **High Replication Datastore (HRD).** It offers a big availability for reads and writes but it has the highest level of latency writing.

The datastore saves data objects called entities. Each entity has one or more properties, that means specific values of one of the allowed data types. For instance, a property could be a string, an integer, etc. The Datastore can execute multiple operations in a single *transaction*. A transaction cannot succeed unless every one of its operations succeeds; if any of the operations fails, the transaction is automatically rolled back. **This is especially useful for distributed web applications,** where multiple users may be accessing or manipulating the same data at the same time.

2.3.6 Google Cloud SQL

Google Cloud SQL is a web service that allows to create, configure and use relational databases with App Engine applications. It is a fully---managed services that maintains, manages and administers MySQL databases. The system has a high data portability and the ability to move easily data into and out of the cloud wich allows to have the database working fast in the App Engine application. Moreover, Google Cloud SQL is very robust on account of it provides high data availability replicating data to multiple geographic regions.

Advantages:

- **Easy to use.** It provides a graphical user interface to manage the database instances.
- **Fully managed.** All the database management chores is done by Google.
- **Highly available.** Service is available even if a data center becomes unavailable.
- **Integrated with Google App Engine.**

Disadvantages :

- The service is in **limited preview.**
- It is necessary to **pay a fee.**

2.3.7 Google Cloud Storage

Google Cloud Storage API is an experimental, innovative and rapidly changing feature for App Engine. At this moment this possibility is not valid for us because the changes may make incompatible this API for App Engine.

2.3.8 Chosed Data Storage Method

At the first moment of the project development I chose the AppEngine Datastore with the Python API because is very strong with distributed web applications but inside the Google+ Hangouts there are several problems with AJAX request to Python servers in all the browsers (except Firefox) so the app was only working in Firefox, moreover the development was being slow having to upload to the server all the files when I made changes. So because of these problems, I decided to use mySQL (doing the querys from PHP) and develop the app faster using my local server. After the development, I have achieved the app in a normal website and I only have to adapt it to the Google Hangouts so, it is possible to create and edit games in a website or in the Google+ Hangouts with the distributed way.

2.4. Programming languages

A programming language is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal. Each language has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions.

High-level programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands, called machine languages. Each different type of CPU has its own unique machine language.

Lying between machine languages and high-level languages are languages called assembly languages. Assembly languages are similar to machine languages, but they are much easier to program in because they allow a programmer to substitute names for numbers. Machine languages consist of numbers only.

The choice of which language to use depends on the type of computer the program is to run on, what sort of program it is, and the expertise of the programmer.[iv]

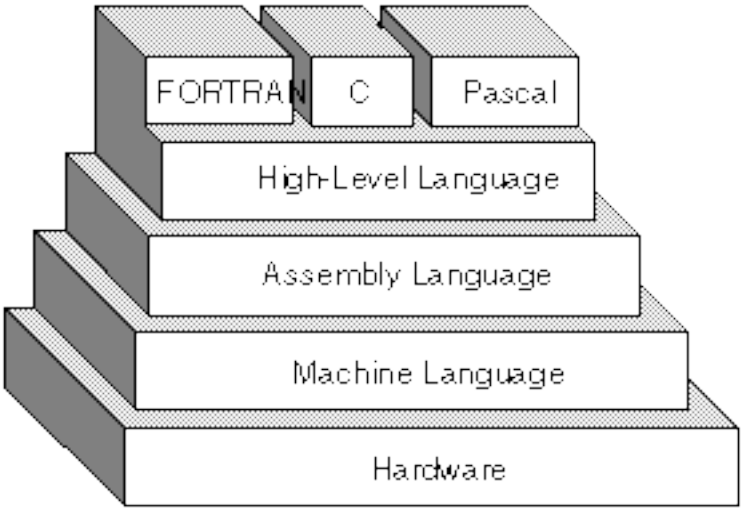


ILLUSTRATION 5. PROGRAMMING LANGUAGES SCHEME

2.4.1 PHP

PHP (PHP Hypertext Preprocessor) is a scripting language open source high-level and interpreted. It is specially designed for the development of dynamic websites because of its robustness, power and speed and can be embedded into HTML code.

Main features:

- Large number of databases supported.
- Its syntax based on Perl and C makes learning fast enough to programmers of these languages.
- The embedded HTML code allows developing interfaces quickly.
- Version 5 of the language is object oriented and has modern features such as reflection and a lot of development Frameworks.

2.4.2 HTML

HTML, English acronym of Hyper Text Markup Language is a markup language designed to structure and present texts in hypertext form, which is the standard format of websites.

Thanks to the Internet and browsers type Explorer or Firefox, HTML has become one of the most popular formats for document construction.

HTML is a child of SGML, although there are versions of XHTML that are descendants of XML. HTML uses tags or marks to define the way that the text should appear in the browser, as well as images and other elements on the computer screen.

Every tag is identified for this enclosed (<>), and some have attributes that can take any value.

2.4.3 CSS

The cascading style sheets (CSS) is a language used to define the style of a structured document written in HTML or XML.

The idea behind the development of CSS is to separate the structure of a document from its presentation.

Advantages[v]:

- **Separation of content from presentation.**
- **Site-wide consistency.**

- **Bandwidth.** A stylesheet, will specify the style once for a range of HTML elements selected by class, type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element.
- **Page reformatting.** With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices.
- **Accessibility.** Without CSS, web designers must typically lay out their pages with techniques that hinder accessibility for vision-impaired users, like HTML tables.

2.4.4 Javascript

JavaScript is an interpreted language website oriented, with a Java similar syntax. Brendan Eich at Netscape Communications Company invented the language, which is the company that made the first commercial Internet browser.

It is used in HTML websites to perform tasks and operations within the client application, such as form validation.

jQuery[vi]

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. Used by over 55% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, dual-licensed under the MIT License or the GNU General Public License, Version 2. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

infoBubble

infoBubble is a javascript library plugin used to change easily the style of InfoWindows in a Google Map v3. The style of InfoWindows is fixed and very hard to modify only with CSS, because of that this library was developed making easy giving style to InfoWindows.

Simplemodal

SimpleModal is a lightweight jQuery Plugin which provides a powerful interface for modal dialog development. Think of it as a modal dialog framework. SimpleModal gives you the flexibility to build whatever you can envision, while shielding you from related cross-browser issues inherent with UI development.[vii]

MapIt

MapIt is a jQuery plug-in that interacts with the Google Maps API. MapIt groups and organize all the locations into an easy to use interface. All the heavy lifting is done so it isn't neccessary learning the Google API calls.

2.4.5 SQL

Structured Query Language is a declarative language to access relational databases that allows you to specify various types of operations on them. It combines algebra features and relational calculus allowing launching queries in order to retrieve relevant information from a database.

Currently, the SQL is the standard for the majority of commercial DBMS. SQL provides a rich functionality beyond the simple query of data. It takes the role of data definition language, a views definition language and data manipulation language.

It also allows the granting and denial of permissions, the implementation of integrity constraints and transaction controls, and alteration of schedules. This language has also an extension called FSQL (Fuzzy SQL) for the treatment of vague information using fuzzy logic in fuzzy databases.

2.5. API's

2.5.1 Google+ Hangouts API

Google+ Hangout API allows to the developers to create and register apps that run inside Google+, allowing to share contents on public manner, indentify themselves or even keep conversations. According to Google+ the API should be used based in openness, the respect on user data and putting the user before everything.

The current API has been launched as a preview and it's planned to change and improve it soon.

The apps are represented by an XML file and the application code is written in HTML, CSS and Javascript so external Javascript apps or style sheets documents can be loaded.

To run the app inside of a hangout is necessary to publish the gadget XML in a public URL and register the app in the Google APIs Console. Once this is done, the app can be run starting a hangout from the APIs Console.

There is a Javascript interface to hangouts wich allows the app to do things like list the participants, control the view of the interface or control the media settings. The Hangout API is divided into 5 namespaces wich group in turn the hangout classes functions and events. The namespaces are the following:

- **namespace gapi.hangout.** It provides information such as the list of participants, the locale and if the app is initialized and visible.
- **namespace gapi.hangout.av.** It provides ability to control media settings.
- **namespace gapi.hangout.av.effects.** Allows to add sound effects and attach image overlays to faces.
- **namespace gapi.hangout.data.** Allows the participants to sharing data.
- **namespace gapi.hangout.layout.** It provides ability to set UI layout elements such as the video feed, chat pane and notices.

Moreover, Google offers libraries for several programming languages such as .NET , Java, Objetive C, PHP , Python or Ruby increasing the possibilities of the developers.

In addition, the last version of the API was launched last 8th February and it's the version 0.2 so as can be seen, this is a very new platform with great growing expectations which provides the possibility to be learnt easily by new developers.

2.5.2 Google Maps Javascript API v3

Google Maps JavaScript API allows inserting Google Maps in web pages. Version 3 of this API is designed to provide more speed and an easily use of the maps in mobile applications and in the traditional desktop browser.

The API provides several utilities for manipulating maps and to add content to the map using many services, allowing creating robust maps applications on web sites.

2.6. Google App Engine SDK (Python version)

App Engine is a web hosting service provided by Google free of charge until certain quotas. This service allows to run applications on Google's infrastructure. At this moment, free accounts are limited to one gigabyte of permanent storage and enough bandwidth and CPU for five million visitors a month, and if the application exceeds these quotas, additional quotas can be bought.

App Engine includes the next features:

- Dynamic web server, fully compatible with common web technologies.
- Permanent storage, browsing, sorting and transactions.
- APIs for authenticating users and sending email using Google Accounts.
- A complete local development environment that simulates Google App Engine on the computer.
- Task queues that perform works outside the scope of a web application.
- Scheduled tasks to trigger events at specific times and at regular intervals.

2.7. Data Exchange

2.7.1 JSON[viii]

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.

An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on the next forms:

- ○ **Object.** An object is an unordered set of name/value pairs. An object begins with "{" and ends with "}". Each name is followed by ":" and the name/value pairs are separated by ",".

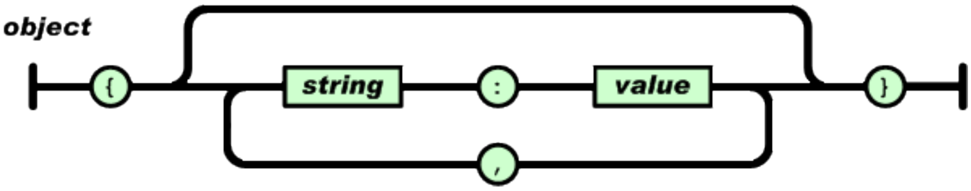


ILLUSTRATION 6. JSON OBJECT

- ○ **Array.** An array is an ordered collection of values. An array begins with "[" and ends with "]". Values are separated by ",".

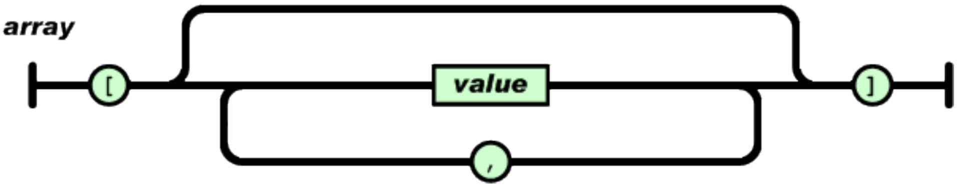


Illustration 7. JSON Array

- **Value.** A value can be a string in doubles quotes, or a number, or *true* or *false* or *null*, or an *object* or an *array*. These structures can be nested.

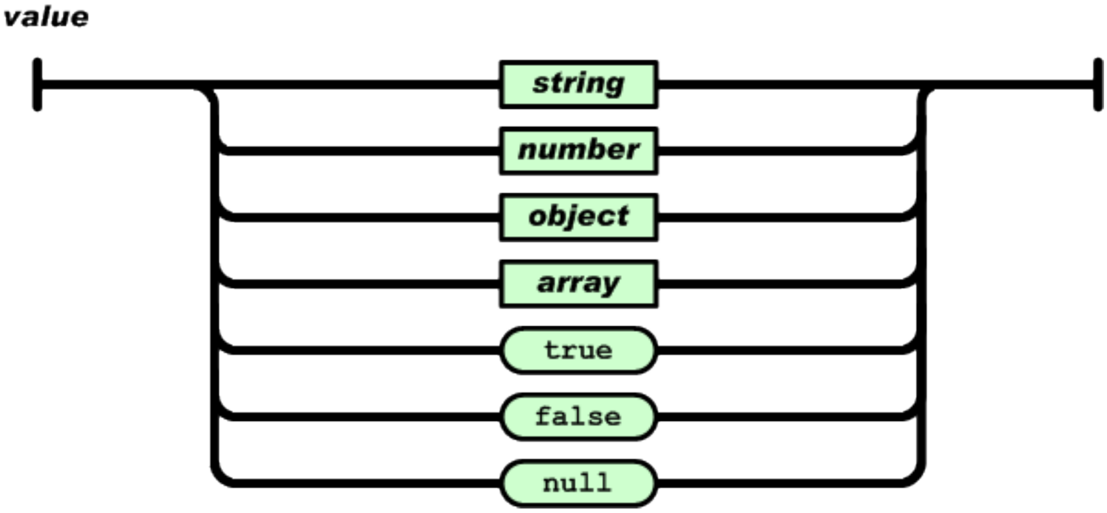


Illustration 8. JSON Value

- **String.** A string is a sequence of zero or more Unicode characters, wrapped in double quotes, using backlash escapes. A character is represented as a single character string. A string is very much like C or Java string.
- **Number.** A number is very much like a C or Java number, except that the octal and hexadecimal formats are not used.

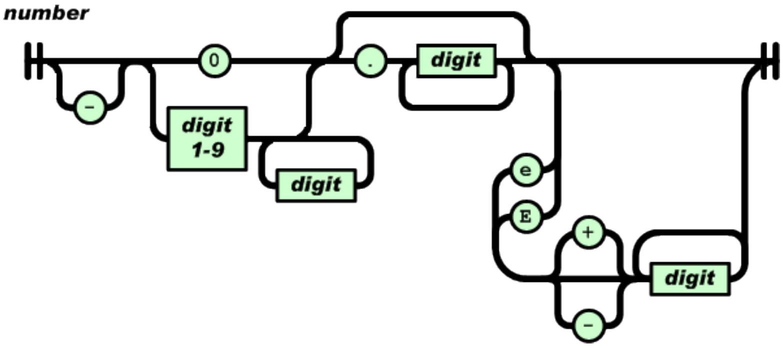


Illustration 9. JSON Number

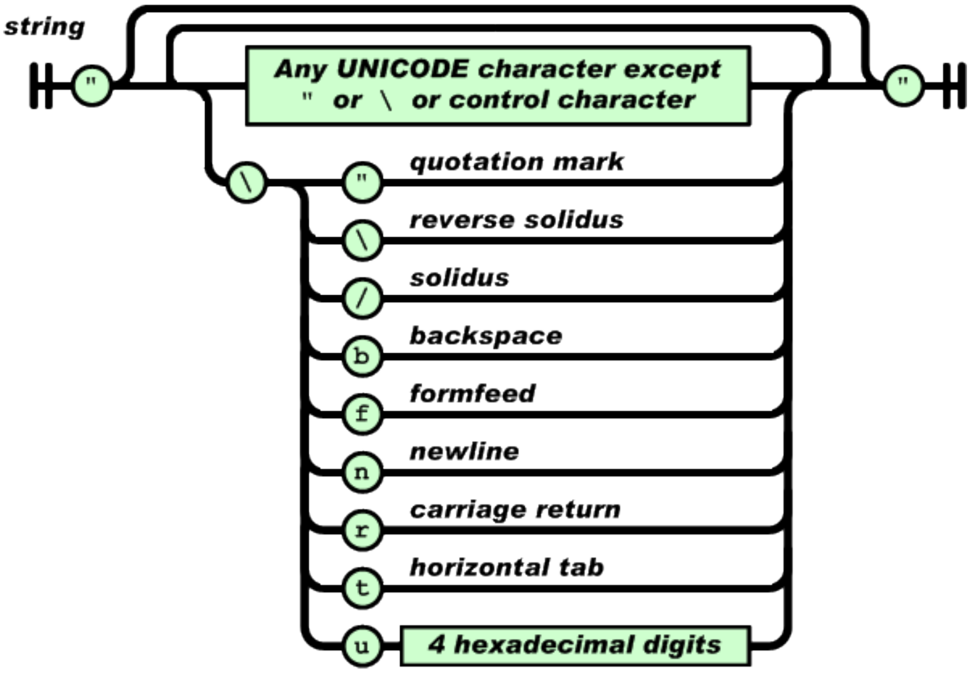


ILLUSTRATION 10. JSON STRING

2.7.2 XML Pull Parsing

XML is a markup language developed by the World Wide Web Consortium (W3C). Derived from SGML and allows to define the grammar of specific languages (in the same way that HTML is itself a language defined by SGML) to structure large documents. XML support databases, and is useful when multiple applications are required to communicate with each other or integrate information.

Pull parsing[ix] treats the document as a series of items which are read in sequence using the Iterator design pattern. This allows for writing of recursive-descent parsers in which the structure of the code performing the parsing mirrors the structure of the XML being parsed, and intermediate parsed results can be used and accessed as local variables within the methods performing the parsing, or passed down (as method parameters) into lower-level methods, or returned (as method return values) to higher-level methods. Examples of pull parsers include StAX in the Java programming language, XMLReader in PHP and System.Xml.XmlReader in the .NET Framework.

A pull parser creates an iterator that sequentially visits the various elements, attributes, and data in an XML document. Code which uses this iterator can test the current item (to tell, for example, whether it is a start or end element, or text), and inspect its attributes (local name, namespace, values of XML attributes, value of text, etc.), and can also move the iterator to the next item. The code can thus extract information from the document as it traverses it. The recursive-descent approach tends to lend itself to keeping data as typed local variables in the code doing the parsing, while SAX, for instance, typically requires a parser to manually maintain intermediate data within a stack of elements which are parent elements of the element being parsed. Pull-parsing code can be more straightforward to understand and maintain than SAX parsing code.

2.8. Architecture and hosting

As it was said before (inside the Abstract), firstly the application was going to be fully hosted into appspot.com via AppEngine but due to problems with the communication problems I decided to change the storage to the university server. Finally, the hosting system is divided in static files and dynamic files.

Static files (XML, JavaScript, CSS files and images) are allocated into the appspot in the next URL: http://rllomun.appspot.com/ . And the dynamic files (PHP files) and the database are hosted into the university server (http://150.140.188.242/gce).

To receive correctly the information from the university server it was necessary to include the next header into the PHP documents:

```
header('Access-Control-Allow-Origin: *');
```

This is a cross-site HTTP request that requests for resources from a different domain than the domain of the resource making the request. Finally, the response between the crossed domains is sent as a JSON object.

2.9. Conclusions about the tools

With the above, we arrive at the conclusion of the tools available on the market and the features that each one offers. For this project I have decided to use free software (except for the operating system, I used Mac OS X because it is the main system in my computer and to develop would be quicker), so I opted to have Apache as web server, MySQL database, Javascript and PHP as scripting languages (at the first moment I had decided to use App Engine but due to problems with the storage explained before I changed my mind). This decision is due to the boom of these tools led to a market that is integrated into a package called MAMP (acronym for Mac, Apache, MySQL and PHP).

MAMP is one of the best tools available for the users to use a versatile and powerful web server. Though they are created separately, each one of the technologies that form MAMP has many common characteristics. Especially interesting is the fact that these products can work in a wide range of hardware, with relatively small requirements without losing stability. This has become the alternative to MAMP (or LAMP in Linux case) most suitable for small and medium enterprises.

Advantages of the use of MAMP:

- **Simple.** The code is relatively simple, with little changes from one platform to another.
- **Security.** Patches are generated soon after found a security hole.
- **Comfortable Updates.** Software updates via Internet.
- **Own improvements.** There is the possibility to increase the services and functions from the source code.

Disadvantages of the use of MAMP:

- **Shortly intuitive.**
- **Knowledge.** Sometimes updates require knowledge of the system.
- **Time.** Configure some services require long time.

3. DESIGN

DESIGN AND DEVELOPING OF A GAME EDITOR FOR CITYSCRABBLE APP

3.1. Introduction

The way to develop applications for the Google+ Hangouts is very similar than developing websites. The difference is that inside the hangouts it's better to develop the application in only one screen. It is possible to have more than one screen like in a website and move among them using links but the people inside the conversation may loose the attention and the hangouts experience could become extremely slow.

For this reason, it is better to have all the necessary elements for the application in the same screen, this implies that these elements have to be very compact and if we have many elements, organize them in order to keep the attention of the user and always thinking that the user has to know what is happening and where is everything.

As I explained before, CityScrabble can be played in any location. Following the philosophy of compact elements, I have used a dynamic map (the city showed change every time that we select a game) to choose the location.

Now, we have the location and we need the topics and the hints (called elements in my case). All the elements belong to a topic (only one) and are associated with all the markers (only one is the correct, but the others could be relatives with less score or in other case with score 0). In order to be the more compact possible and clear for the user I have chosen a table to relate these four kinds of elements (topics, elements, markers with the exact position and score).

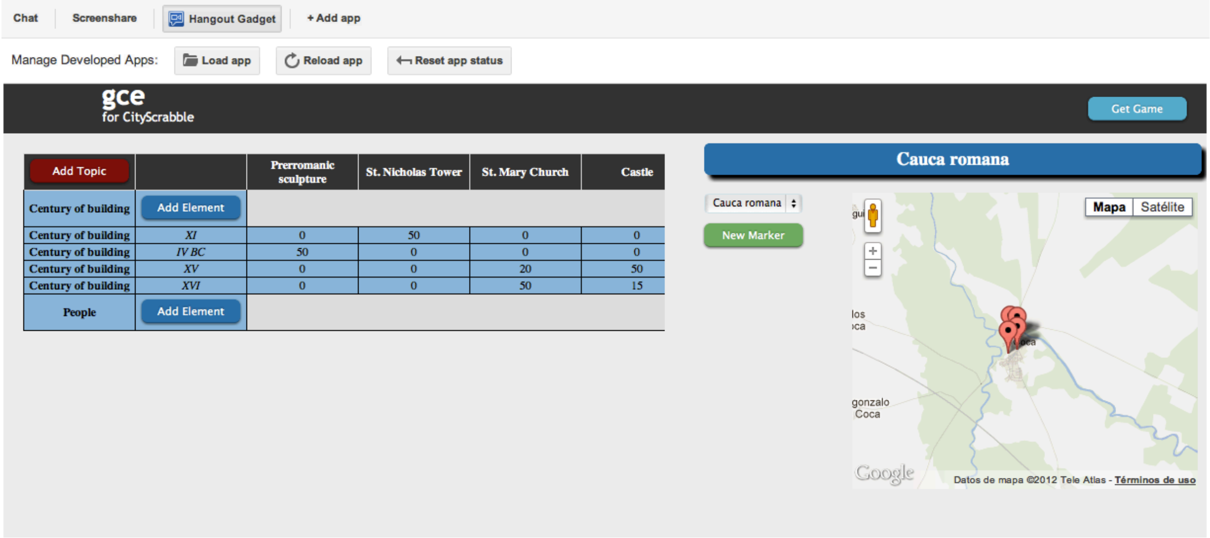
I also wanted that all the elements were the most editable possible allowing to change their names and their features or to remove them. For this purpose, I have used hidden DIV's with pop-up style.

Finally, to obtain the JSON file of the game I have used a button that recovers all the information of the game and generates the file.

3.2. The screen

As I have mentioned, it is better if the editor only has one screen with all the elements and options. The screen of the editor has the next elements:

- **Game form.** Where we select the game that we are going to edit or to create a new game.
- **Dynamic map.** Here we can add all the markers we want for the different places in the game.
- **Dynamic table or Associative Matrix.** Showing all the relationships between topics, elements, markers and their score in each case.
- **Marker generation button.** Clicking it and then clicking in the map, we will create a new marker in the place clicked.
- **Game generation button.** It creates the JSON file with all the data of the game.



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
Century of building	*XI*	0	50	0	0
Century of building	*IV BC*	50	0	0	0
Century of building	*XV*	0	0	20	50
Century of building	*XVI*	0	0	50	15
People	Add Element				

ILLUSTRATION 11. SCREEN

In the figure, the screen can be seen editing a game in my hometown (Coca), which I'm going to use to explain the design of the editor. Mainly, we can divide the screen in two important parts: the map part (including the game name, the game selection form, the marker button placer) and the table.

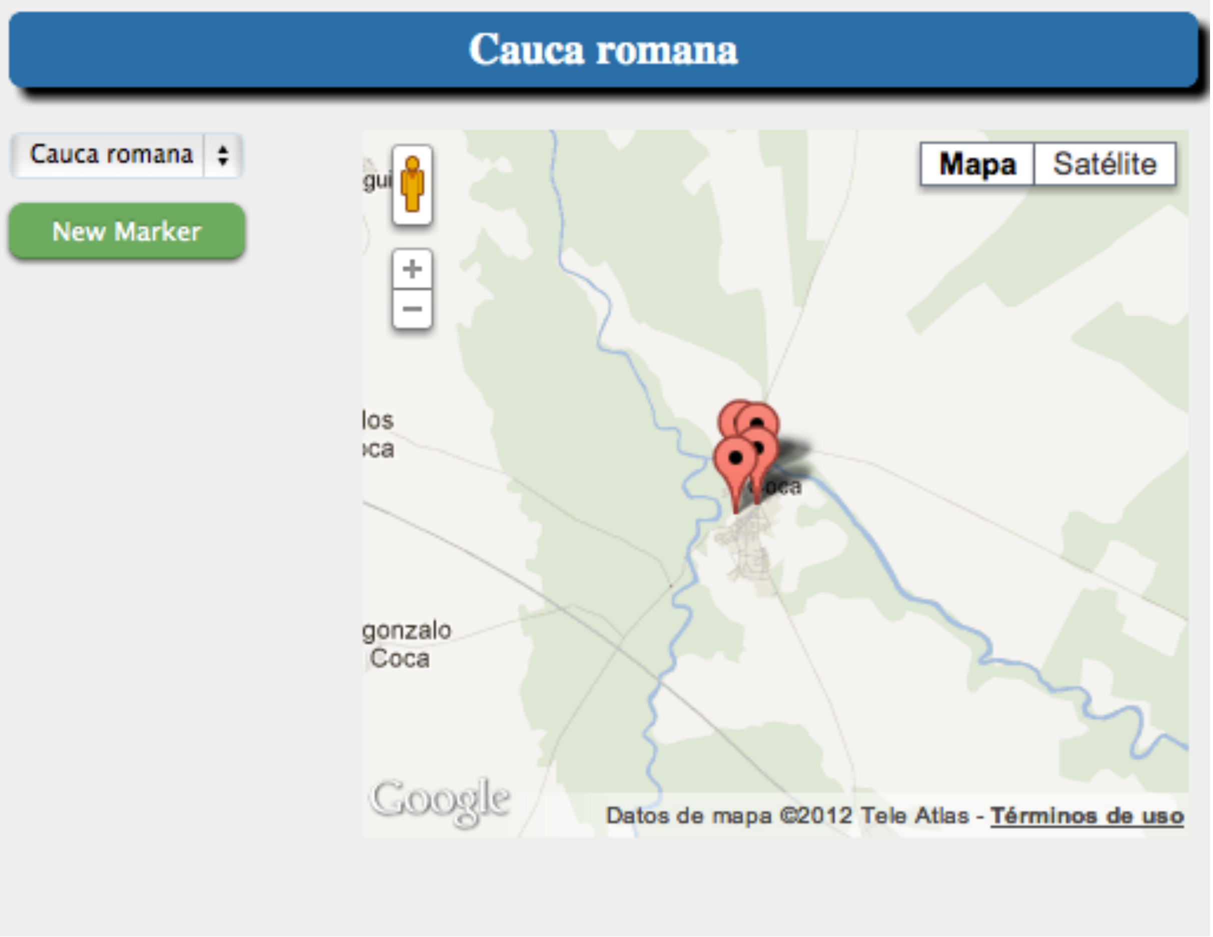


ILLUSTRATION 12. MAP PART

3.3. The map part

3.3.1. Game name label

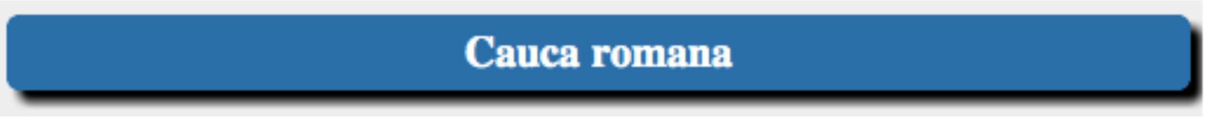
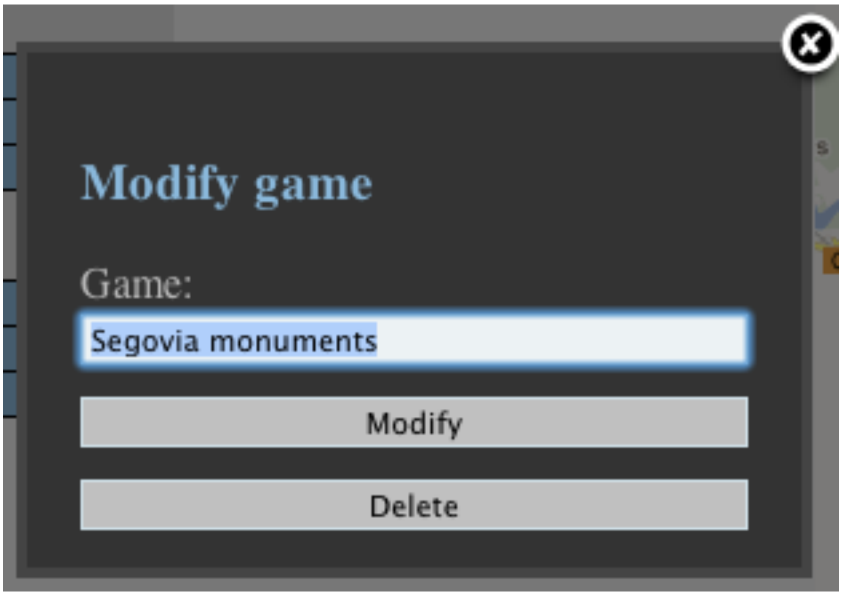


ILLUSTRATION 13. GAME NAME LABEL

It seems to be only one label showing the name of the game but clicking on the name it appears a hidden window giving us the ability to change the name of the game or to delete it.

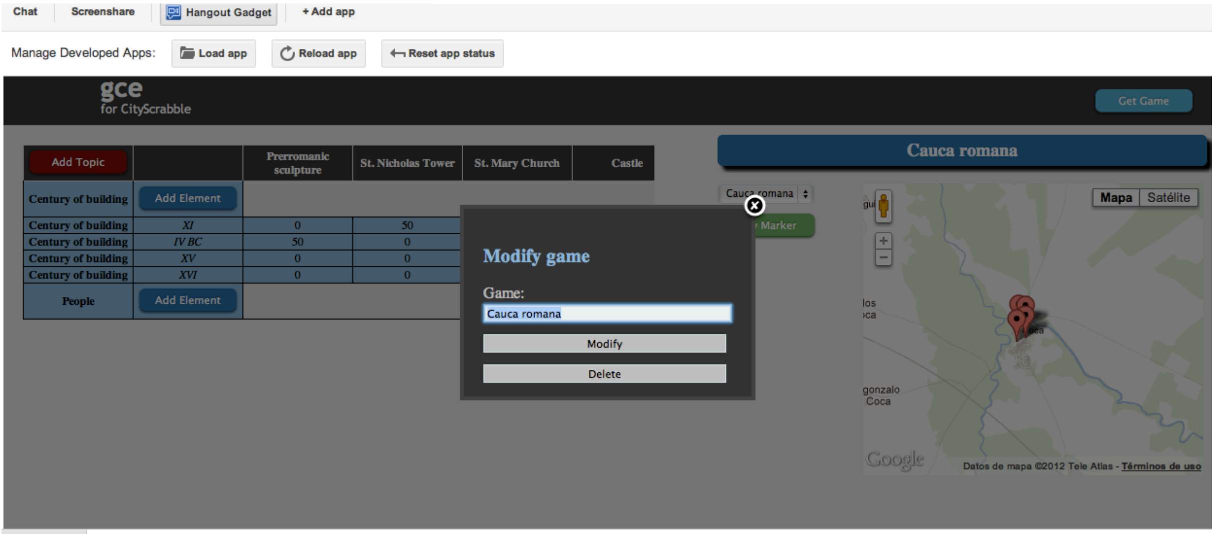
This kind of window appears in centre of the screen and catches the attention of the user darkening the back as we can see in the figure.



Modify game

Game: Segovia monuments

Illustration 14. Modifying a game name



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
Century of building	XI	0	50		
Century of building	IV BC	50	0		
Century of building	XV	0	0		
Century of building	XVI	0	0		
People	Add Element				

Modify game

Game: Cauca romana

Illustration 15. View of the dark background

3.3.2. Game selection form

The game selection form is a simple Select pane that includes the list of games stored in the database and the option to create a new game.

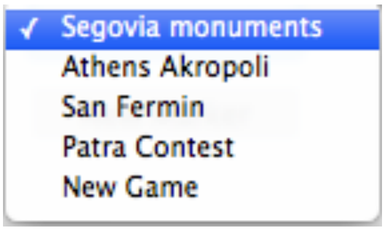
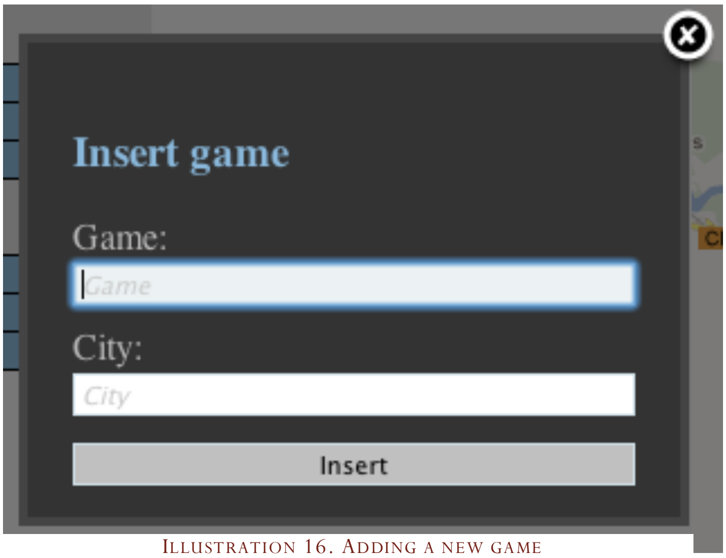


Illustration 15. Game select

When another game is chosen the map shows automatically the city where that game will be played and if we choose to create a new game, it will appear a hidden window like the next:


Insert game

Game:

Game

City:

City

ILLUSTRATION 16. ADDING A NEW GAME

3.3.3. The map

The map is centred in the city chose when a new game was created and it shows the markers placed in the different locations where the contestants will have to go when they play the game.

Adding markers to the map is very simple: “Add marker” button must be clicked and after that just clicking in the exact point where we want the marker in the map the new marker will appear. To fill in the information about the marker a new hidden window will appear like Illustration 19.

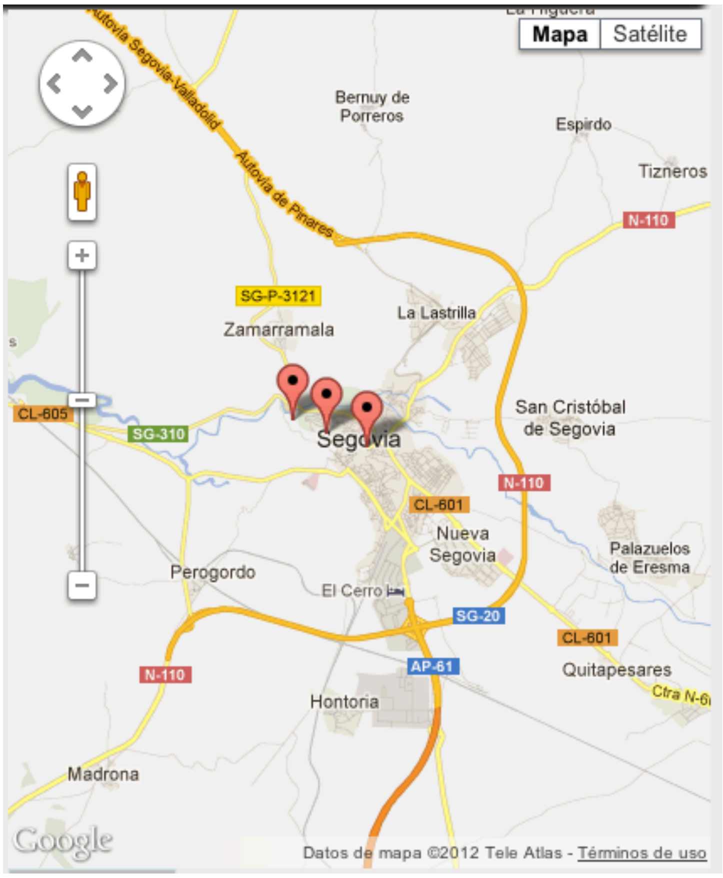
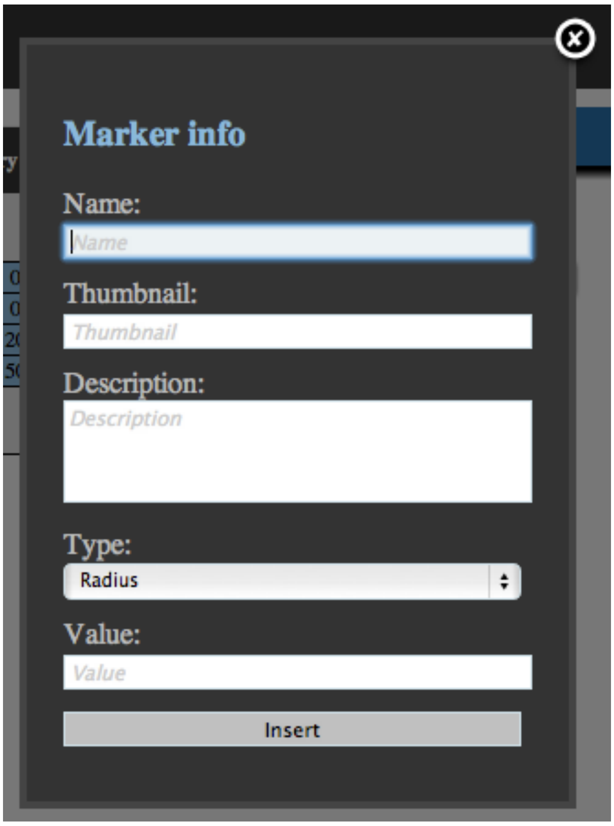


Illustration 17. Map



Marker info

Name:

Name

Thumbnail:

Thumbnail

Description:

Description

Type:

Radius

Value:

Value

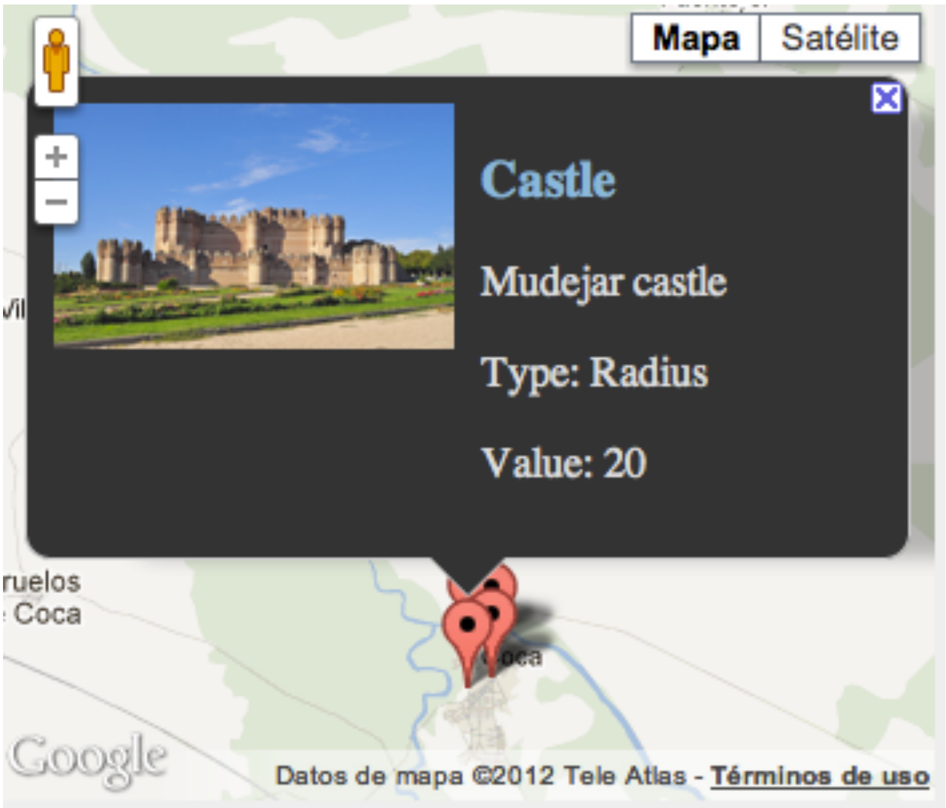
Illustration 18. FILLING IN THE MARKER INFO

This window includes a form asking for the info of the marker. First, it asks about the name of the marker, after this, the thumbnail must be a valid URL.

The third element of the form is the description. After it we have the coordinates, this element will be blocked and cannot be modified, it saves the exact coordinates of the marker.

For type input we have a selection pane with the next options: Radius, QR Code or Bearing, depending on the type of identification that CityScrabble will do when the contestants answer the element relative to this marker. Finally, we have the value of the type and the "Insert" button to store the info into the database.

Now, we have the map centred in the city of the game and the markers needed; if a marker is clicked an "infoBubble" will appear showing the information of the marker like in the next figure:



Castle

Mudejar castle

Type: Radius

Value: 20

ILLUSTRATION 18. INFOBUBBLE

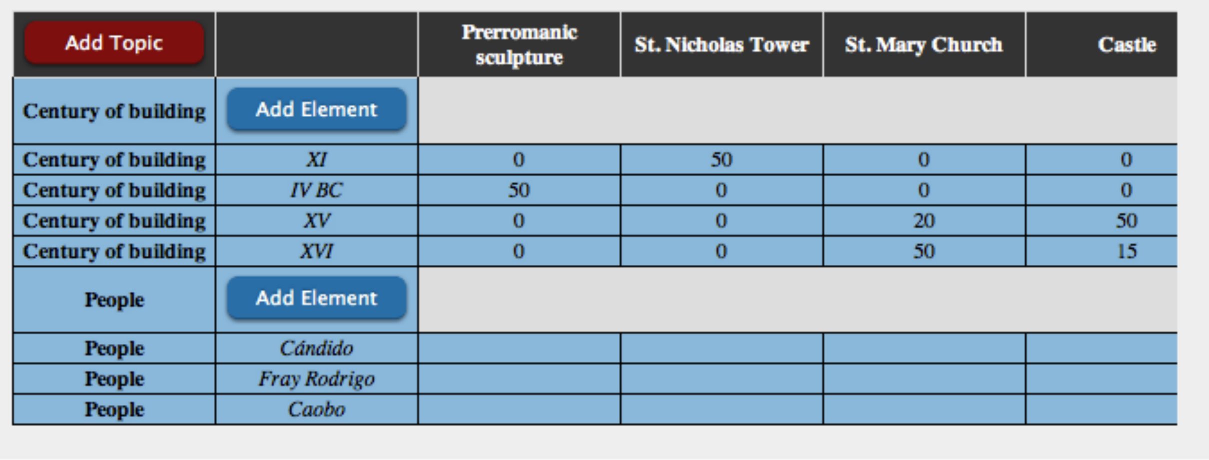
3.3.4. Game generator button

This it isn't exactly part of the mart part but it's exactly on the top of it. Clicking this button, the JSON code with all the information of the game will be created and it will appear a download window to save the file in our computer ready to insert it into CityScrabble game and play. In the next chapter I will explain how this file is generated and I will show an example.

3.4. Associative matrix or table

In order to establish relationships between the markers and the elements of the topics I have design this dynamic-built table. The shaping is explained below.

In the figure, there are four markers placed in the main monuments (the prerromanic sculptures, the Tower, the church and the castle) and symbolized in the head of the table, two topics (Century of building and People) shown on the left column of the table (**bold**) and elements of each topic placed in the next column (*italic*). For each topic we have a relationship between them and any marker in their appropriate cell where we place the score of this relationship.

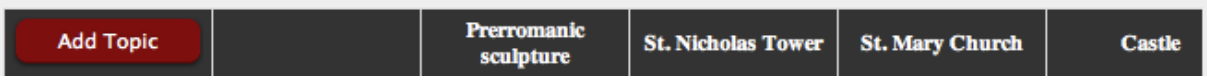


Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
Century of building	*XI*	0	50	0	0
Century of building	*IV BC*	50	0	0	0
Century of building	*XV*	0	0	20	50
Century of building	*XVI*	0	0	50	15
People	Add Element				
People	*Cándido*				
People	*Fray Rodrigo*				
People	*Caobo*				

Illustration 19. Table

3.4.1 How the table is built

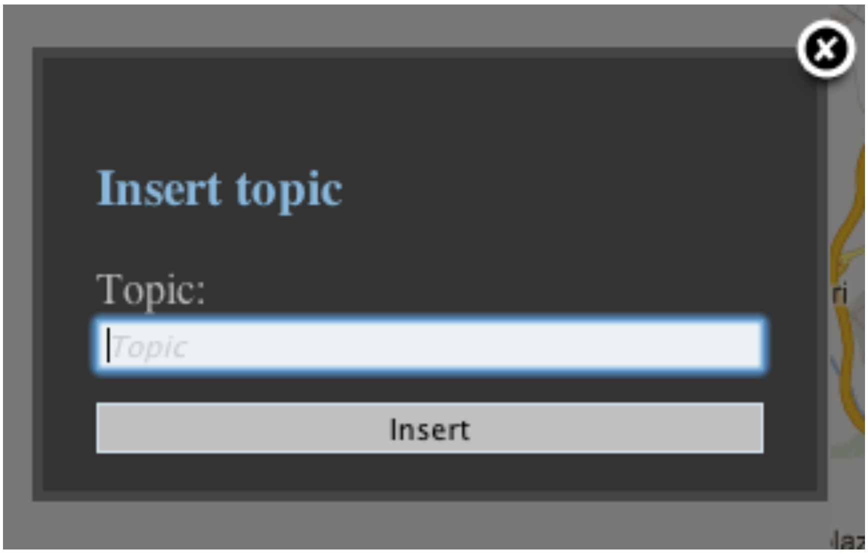
The table is a live element that has to change when we add or remove elements. Firstly, we only have markers (if we don't will appear a message saying "Add markers in order to create the table") and the table has the next shape:



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle

Illustration 20. Only markers table

Now, the table only has one row with the markers and with a button allowing us to create new topics. If we click on it, it will appear the next form asking for info about the topic.

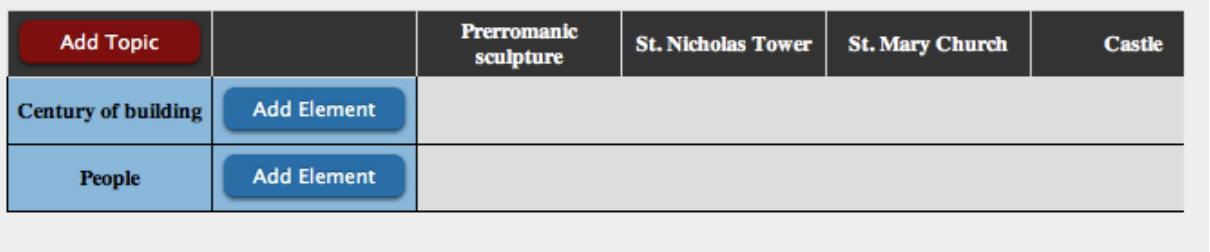


Insert topic

Topic: Topic

Illustration 22. Inserting topics

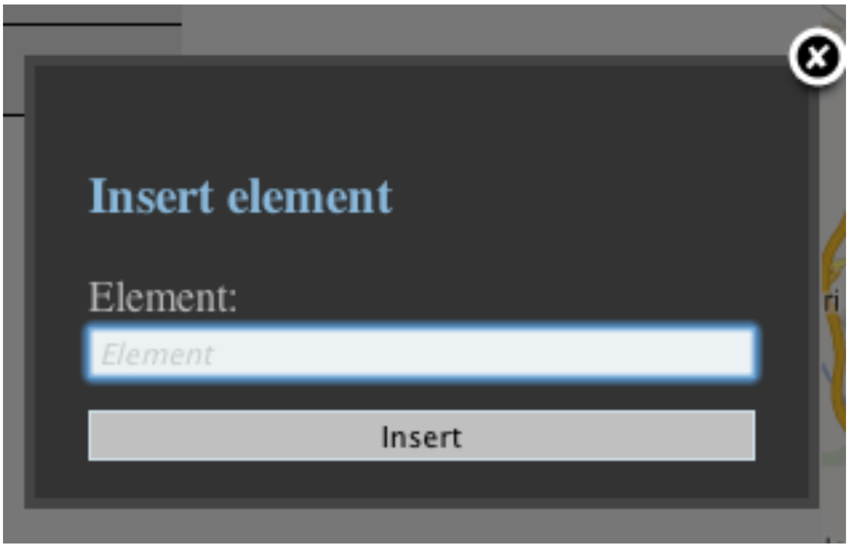
Once we have added the topics, the table will have the next shape:



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
People	Add Element				

ILLUSTRATION 21. MARKERS AND TOPICS TABLE

The app adds automatically two new rows to the table and it give us the chance to insert elements to any topic. So if we want to insert elements to a topic we only have to click in the corresponding button. Doing this will appear the next hidden window:



Insert element

Element:

Element

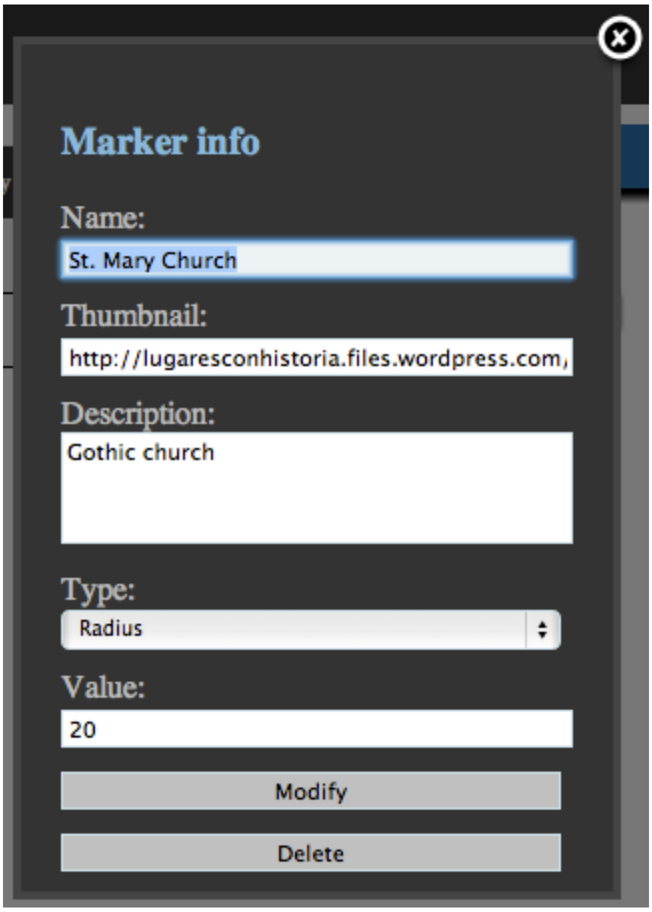
ILLUSTRATION 22. INSERTING ELEMENTS

After the insertion of the elements we will achieve a table like the one shown at the beginning of this point. Namely, the app adds the corresponding cells to insert the topics (clicking in these cells will appear a hidden window asking for the score).

3.4.2 Modifying the elements

Once we have the elements in the table, maybe the need of modify or remove them could appear (because the game has changed, because we made a mistake…). To solve this need keeping the compact form of the application the table adds on each cell a link that activates a hidden window allowing to modify the element in that cell.

For instance, we need to change the parameters of a marker because the thumbnail has changed its URL. If we click the name of the marker on the table will appear the next window:



Marker info

Name:

St. Mary Church

Thumbnail:

http://lugaresconhistoria.files.wordpress.com/

Description:

Gothic church

Type:

Radius

Value:

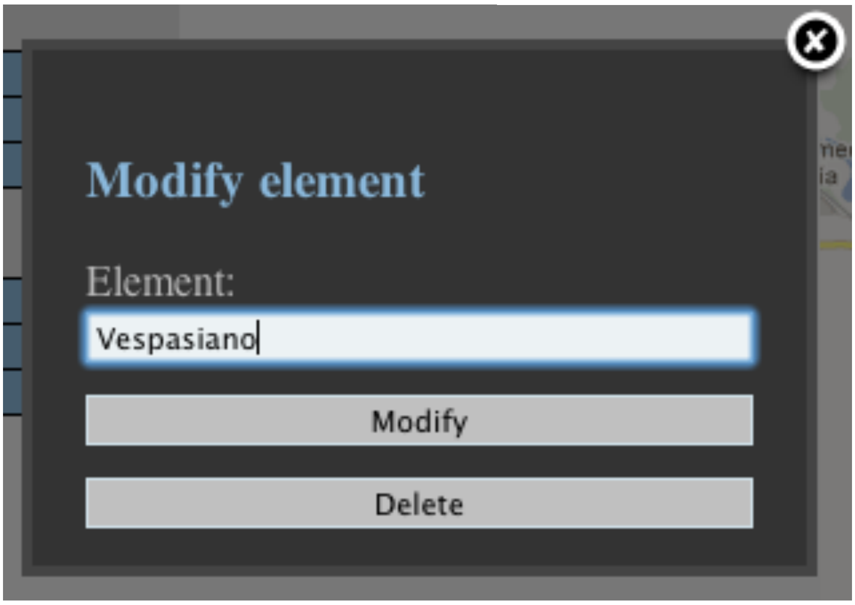
20

This window shows in a form all the information of the marker and allows to change it. Furthermore, it gives us the option to remove the marker.

If we decide to remove the marker, the column of this marker in the table will disappear as well.

In the case that the user needs to modify a topic or an element we can follow the same steps: clicking in the name of the element a new window will appear giving us the chance to modify or delete it. The next figure shows the window that would appear if the user clicks on the name of an element:

ILLUSTRATION 25. MODIFYING MARKERS



Modify element

Element:

Vespasiano

ILLUSTRATION 23. MODIFYING ELEMENTS

Finally, if the score has to be changed, we follow the same process. Clicking in the corresponding cell it will appear a window allowing the user to change score in that relationship.

4. DEVELOPMENT AND IMPLEMENTATION

4.1. Introduction

This has been a practical project in which most of time has been used to learn the technologies of use (developing test applications), developing features and implementing them. Due to this I expend one chapter of this report to explain how the editor has been developed.

To develop it I have mostly used code in HTML (embedded in PHP), PHP and JavaScript. From the start, I have tried to be the clearest possible in the organization of the code and I have tried to do a modular programming in every moment.

For this purpose, I have separated the PHP code in 16 files, each one implements a different task that I will explain hereafter. Meanwhile, all the JavaScript code is separated in of the PHP code in a file and is divided in all the functions needed for the application. Moreover, I have included the JavaScript libraries (like jQuery or simpleModal) that haven't been modified (except simpleModal where I did a little changes to adapt it to my project).

On the other hand, about the style, all the CSS code is included in one file. To write this code I have tried, in order to be clear, to apply it to the "id" of the items, but sometimes I have needed "classes".

Finally, there is the database. It is a MySQL database called "gce" and divided in 5 tables trying to be the most simple and well-working possible.

4.2. Database

In this kind of game, we can have several different **Games**, this games have **Topics**, and any topic has **Elements** about any subject. Also, there are **Markers** that symbolize the location. Any **Element** can be relative of any **Marker** through a **Score.**

I have used the next entity relationship diagram:

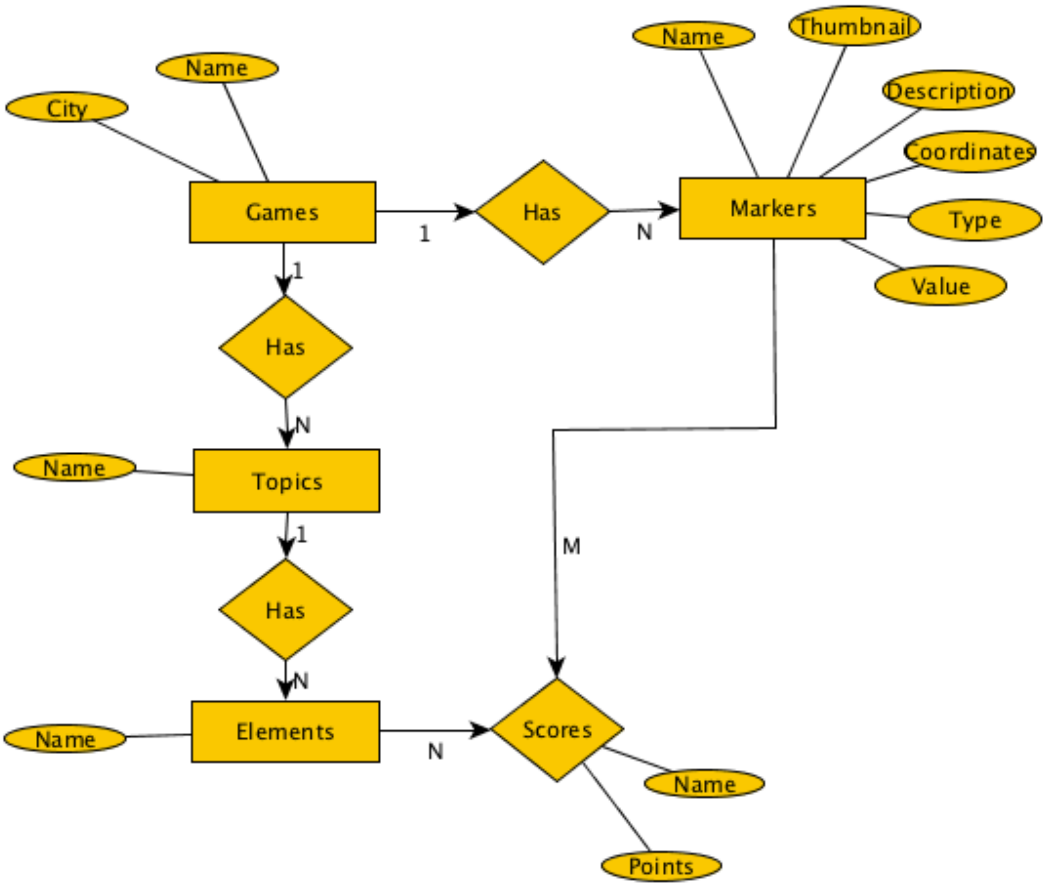
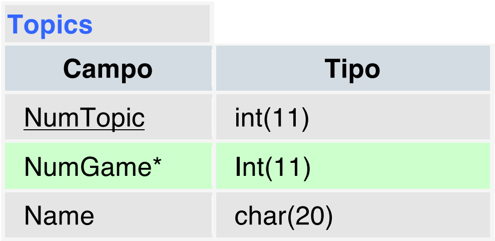


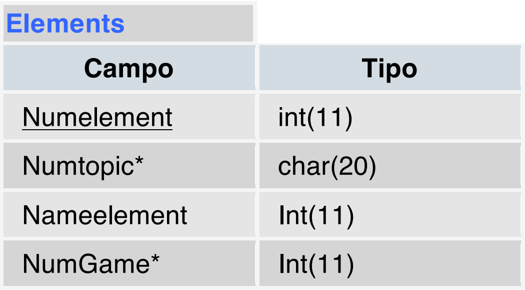
ILLUSTRATION 24. ENTITY RELATIONSHIP DIAGRAM

For this diagram, the relation model obtained is the next:



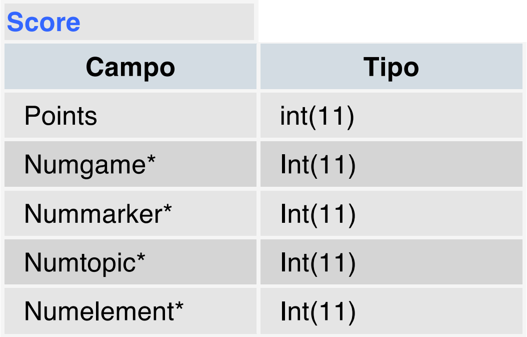
Topics

Campo	Tipo
NumTopic	int(11)
NumGame*	Int(11)
Name	char(20)



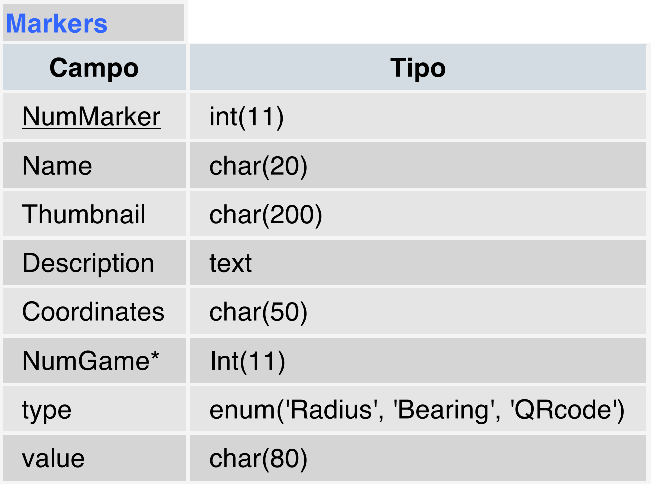
Elements

Campo	Tipo
Numelement	int(11)
Numtopic*	char(20)
Nameelement	Int(11)
NumGame*	Int(11)



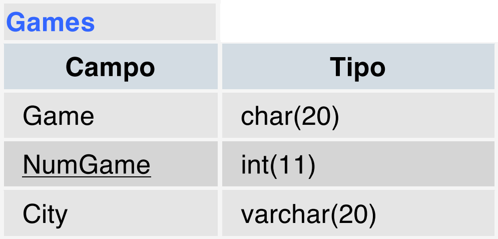
Score

Campo	Tipo
Points	int(11)
Numgame*	Int(11)
Nummarker*	Int(11)
Numtopic*	Int(11)
Numelement*	Int(11)



Markers

Campo	Tipo
NumMarker	int(11)
Name	char(20)
Thumbnail	char(200)
Description	text
Coordinates	char(50)
NumGame*	Int(11)
type	enum('Radius', 'Bearing', 'QRcode')
value	char(80)



Games

Campo	Tipo
Game	char(20)
NumGame	int(11)
City	varchar(20)

4.3. Communication between PHP and Javascript

Sometimes, due to the dynamical conditions of the app, it is necessary to get information from the database in JavaScript. To do this I have used AJAX protocol and XML.

4.3.1 AJAX

Ajax is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required (JSON is often used instead), and the requests do not need to be asynchronous.

Ajax is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and to allow the user to interact with the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.[x]

AJAX sends a request to the server and waits for its answer. The easiest way I have found to use AJAX is in jQuery. The AJAX request I have mostly used in jQuery has the next syntax:

```
var URL;
var DATA;
$.ajax({
        dataType: 'xml',
        data: DATA,
        type: "GET",
        url: URL
        success: function(response){
            doSomething(response);
        }
      });
```

LISTING 1. AJAX REQUEST

We send the JavaScript variables with GET method and we receive a XML object. With the XML object received we do the necessary actions (doSomething()).

4.3.2 Data received in XML

With the AJAX request, we have called a PHP file in the server and it has responded with a XML object. Why XML? As I said before, XML is a markup language to define large structures.

In our case, we could receive a large string of data from the database and it is needed to organize this information. The way I have used to organize the info is using the markups defining inside them the name of the variable and between the markups, the value. Certainly, I could have chose JSON to do this but in the first moment I found XML clearest and easiest. As an example, I'm going to include the code that obtains the info of all the markers of a game:

```
<?php
    header("Content-type: text/xml");

    //Code after obtain the info from database:
    $row = mysql_fetch_array($resultados);
    if($row)
    {
        $output = "<?xml version=\"1.0\"?>\n";
        $output .= "<Information>\n";
        for($i=0; $i<$num_resultados; $i++)
        {
            $output .= "\t<Marker>\n";
            $output .=
"\t\t<NumMarker>".$row['NumMarker']."</NumMarker>\n";
            $output .= "\t\t<Name>".$row['Name']."</Name>\n";
            $output .=
"\t\t<Thumbnail>".$row['Thumbnail']."</Thumbnail>\n";
            $output .=
"\t\t<Description>".$row['Description']."</Description>\n";
            $output .=
"\t\t<Coordinates>".$row['Coordinates']."</Coordinates>\n";
            $output .= "\t\t<Type>".$row['type']."</Type>\n";
            $output .=
"\t\t<Value>".$row['value']."</Value>\n";
            $output .= "\t</Marker>\n";
            $row = mysql_fetch_array($resultados);
        }
        $output .= "</Information>";
        echo $output;
    }
?>
```

LISTING 2. XML PARSING

4.4. Loading the site

This application has been designed to be modified dynamically. So first of all, and to understand the structure of the site I am going to describe and to explain the DOM of the site.

The Document Object Model is a platform and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page.[xi]

The summarized DOM tree of the editor is the next:

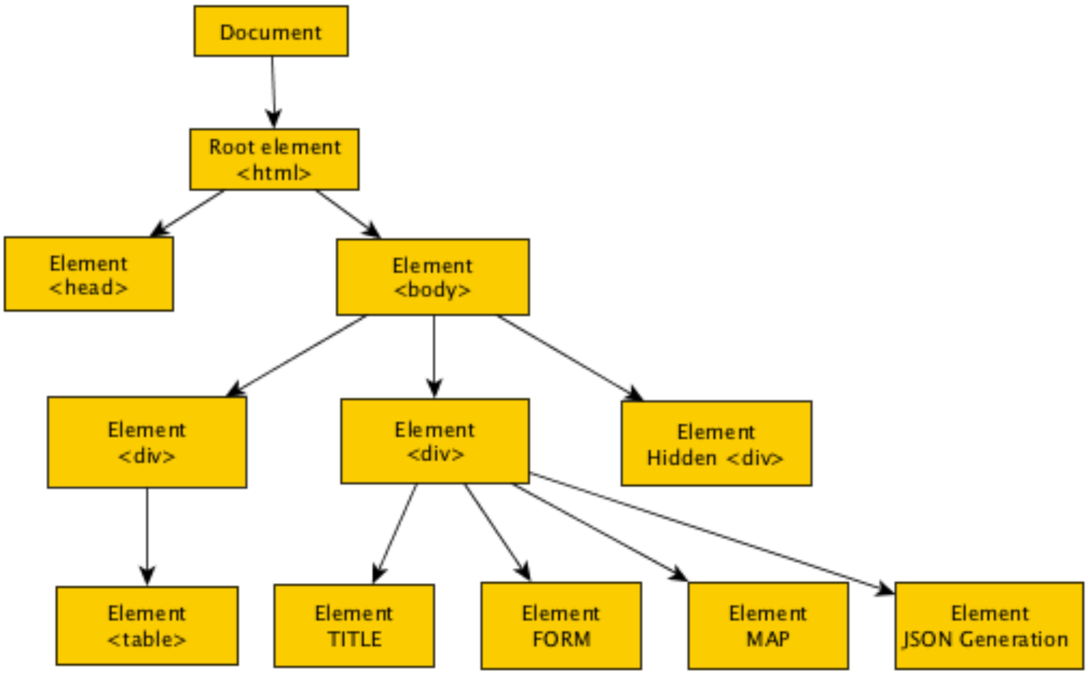


ILLUSTRATION 25. SUMARIZED DOM

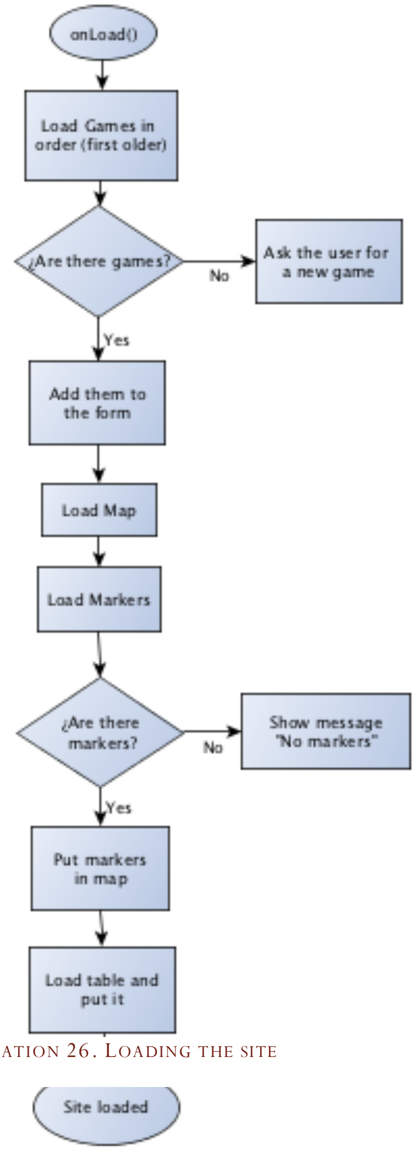


ILLUSTRATION 26. LOADING THE SITE

All the elements in the last level are dynamic elements (the JSON generator and the button to place markers are only buttons but depend directly on the other elements) and they are loaded depending on the games.

All of these elements must appear automatically when somebody access to the site so, with the event onLoad() in the body label, we call a JavaScript function to load them.Firstly, list of saved games is loaded and included in the game form. After, the oldest game is selected and the app writes its name in the title and loads its map with all the markers. Now, with this info the app will load the table. The diagram shows this process.

4.4.1 Loading games

To load the games, it is needed to access to the database and to obtain the info relative to the games.

If there are games saved in the database, the app saves them into an array and it prints them in a select list. If there aren't games the application ask the user to create a new game. When this process finishes, the application starts to load the map.

4.4.2 Loading the map

To add a map to the application I have used Google Maps API v3. The map use the city of the game to centre its position so, first of all we load the city of the game and then the map loads in that city. To load a map we use the next code:

```
var latlng; //Coordinates
  var myOptions = {
    zoom: 8,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var map = new google.maps.Map(document.getElementById("map_canvas"),
      myOptions);
//map_canvas refers to the DIV where we want the map
```

Listing 3. Loading a map

When the map is loaded, a function is called to insert the markers to finish the load of the map. To insert a marker we use the next code:

```
var marker = new google.maps.Marker({

  position: myLatlng,
  title:"Hello World!"
});

// To add the marker to the map, call setMap();
marker.setMap(map);
```

Listing 4. setting markers

After the loading of the map, the table will be loaded.

4.4.3 Loading the table

Once we have loaded the game list and the map, it's time to load the table with the relationships among topics, elements and markers. If there are markers, the table loads; if not, the application shows the next message "Insert markers in order to create the table". The design of the table was explained in the last chapter and now, I include the flow diagram in the figure 30.

Taking a look on the diagram the reader may think that if we have many markers or many elements, the table will overflow. In order to control it, I have designed the table having a maximum height and a maximum width and making it scrollable vertical and horizontally.

It would be pleasant to have the first row (with the name of the markers) and the two firsts columns (with the topics and elements) fixed. I have tried to do it with a jQuery plugin called Scrollabletable but I only achieved to have the head fixed and with bad results.

4.5. User interaction

Once the table is load, the site is ready to receive the user orders using the interface. Now the user has three options: create a new game, select another game or edit the current game.

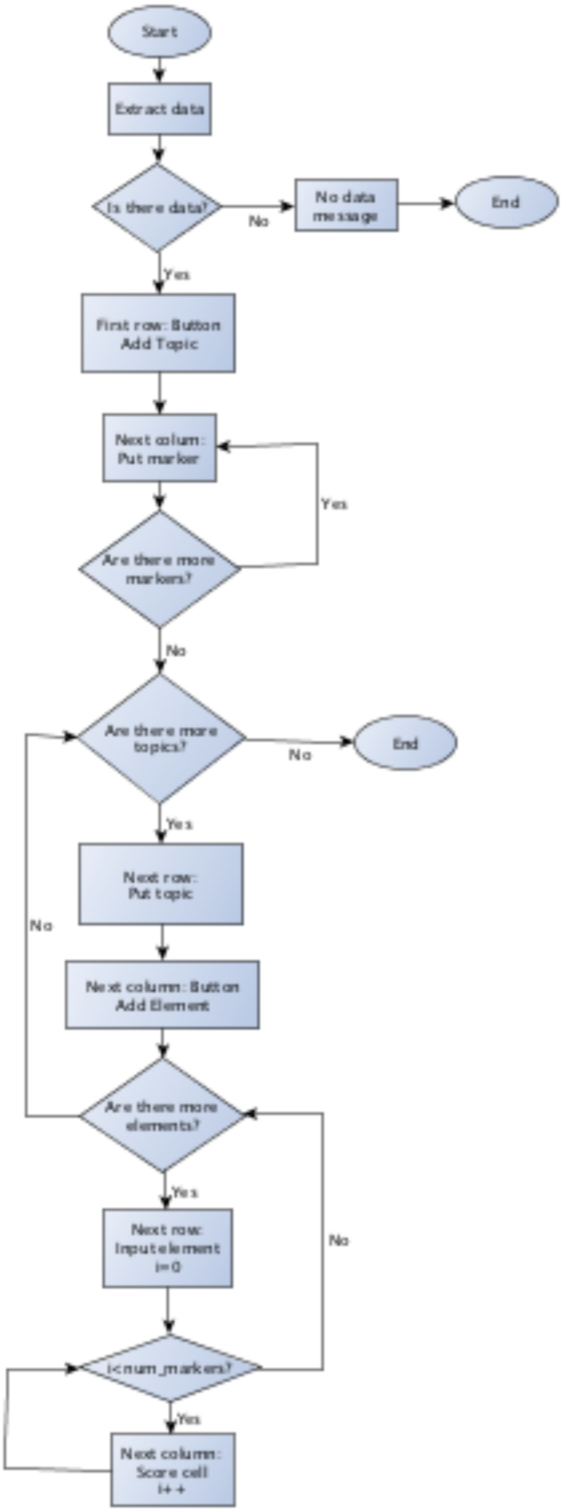


Illustration 27. Table diagram

4.5.1 Creating a new game

If the user chooses the option "New game", it will appear a pop-up window (explained in the chapter 3) asking for the name of the new game and the city.

If the user inserts a new game, firstly the application will save the new game in the database and after it the application will load the site as we explained in the last point but loading the data of the new game created.

In the case of the user decide to close this window, it will close and everything will continue like before pressing "New game".

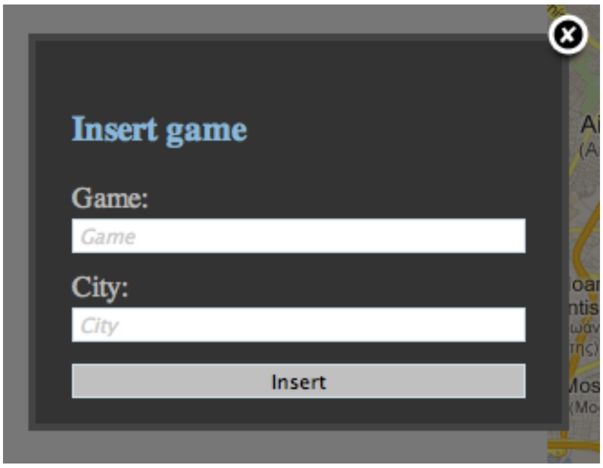


ILLUSTRATION 28. INSERTING A NEW GAME

4.5.2 Selecting another game

In this case, the application will load all the elements as I said in the point 4.4 but with the data of the selected game: in the title will appear the name of this game, the map will be centred in the city of that game and it will show its markers (deleting the pasts ones). Finally its table will be load and the user will be able to edit the game selected.

4.5.3 Editing the game

In the chapter 3 I explained how to edit all the elements of the game. Below, I'm going to explain how does this work.

Inserting new topics, elements or markers

When the user selects one of these options, a hidden window appears asking for the info of the new item. When the user inserts the info, it is added to the database and the editor is updated in order to show the new info. The flow diagram is the next:

The order to insert new data into the database is the next:

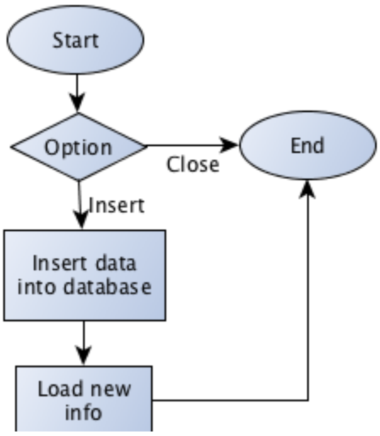


ILLUSTRATION 29. INSERTING DATA DIAGRAM

```
INSERT INTO Table (List of names) VALUES (List of values);
```

Editing topics, elements or markers

If the user decides to edit one of the existing items, a pop-up window with all the information of the item will appear with the chance to modify it or to remove it. In the first case, the item will be updated and the elements of the site will be reloaded in order to show the changes.

If the users decide to remove it, first, the item will be remove of the database and then the elements of the site will be reloaded. The other case is when the user decides to close the editing window. In this case, the editor keeps all the elements how they were.

The next diagram shows the flow diagram of the modifying mode:

In this case, we have to take in account that there are a lot of relationships into the database: elements belong to topics; topics and markers belong to games; scores join elements and markers, etc.

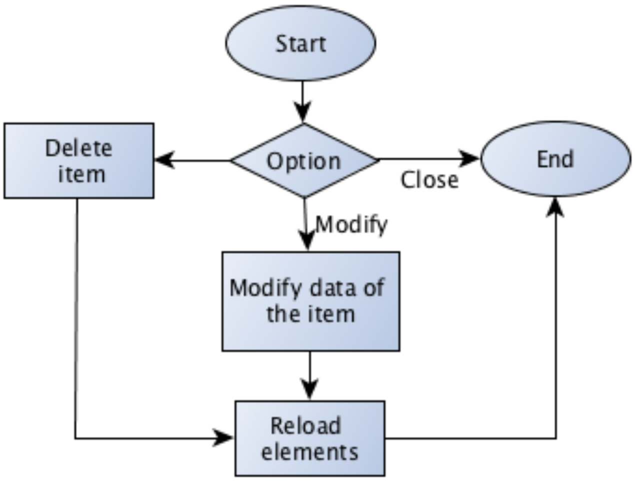


ILLUSTRATION 30. CHANGING ITEMS

For this reason we have to be very careful. If the editor is changing the name of the game, we must to update this information also in the Markers table and in the Topics table.

If he/she is modifying the topics, we must also to update the corresponding elements. In the case of modifying the markers, Score table and Elements table have to be updated too.

Finally I include the order to update data in a table and the order to delete data:

```
UPDATE Table Set Field = Value WHERE Options;
```

LISTING 5. UPDATING EXISTING DATA

```
DELETE FROM Table WHERE Options;
```

LISTING 6. DELETING DATA

4.6. Generating the JSON file

Once the game is completely edited the user has the option to export it to a JSON file and download it in order to play the game in CityScrabble.

There are many relationships into each game between all the fields. These relationships must be conserved in the JSON file in order to keep all the information. If we lose the smallest part of information, the game inside CityScrabble may seem completely different to the one we designed.

For this reason, the final JSON file it is a complex array of arrays that follows the next structure:

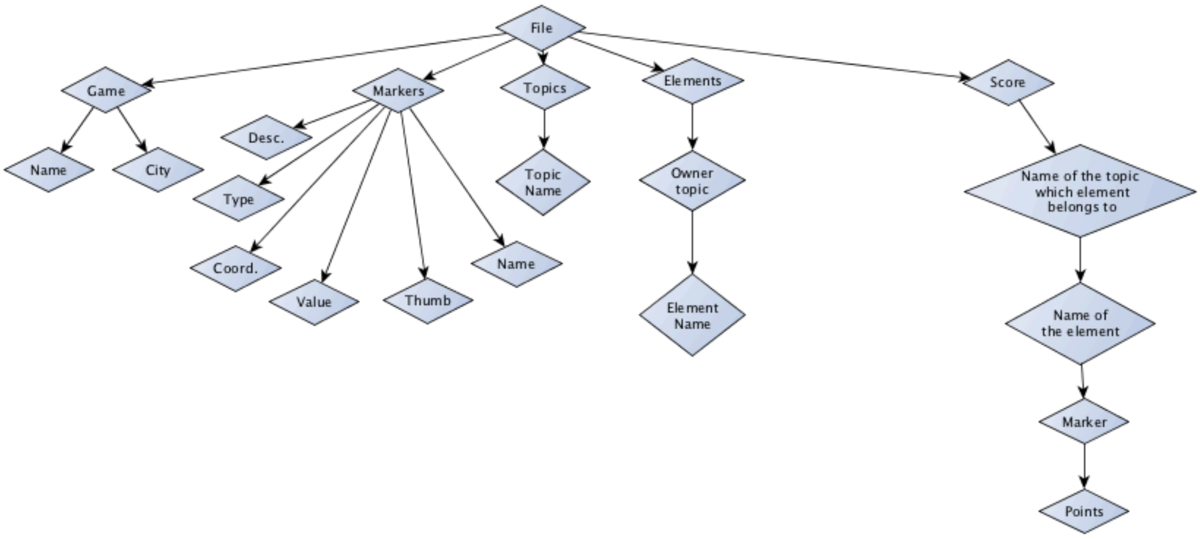


ILLUSTRATION 34. JSON STRUCTURE

4.7. The collaborative mode

The purpose of using the Google+ Hangouts it is to provide to the users the ability to create and edit games simultaneously (collaborating among themselves) through the Internet. This ability allows to a group of users to discuss the different parameters of the game while they are editing it (live edition) in the same application. Obviously, this advantage could help saving time or finding and repairing mistakes easily.

To implement this kind of application, Google offers the Google+ Hangouts API (explained in the chapter 2). This API is divided in 5 namespaces but for the GCE editor I only have used two: **gapi.hangout** and **gapi.hangout.data**.

The first one is only used in the application to establish the number of participants in the hangout and to know when the interface is ready. To know this info I used the next piece of code:

```
gapi.hangout.onApiReady.add(apiReady);
```

It detects when the interface is ready and starts the application.

In the editor, there are some events that I want in collaborative mode and some of them that I didn't want on this way (for example, the form that appears adding a new marker is more useful if only one person can edit it). For the collaborative mode events, I used the "gapi.hangout.data" namespace in the next way: there is a store of shared events where we insert the events that must be seen for all the participants.

```
gapi.hangout.data.submitDelta(key);
```

When the event is detected, a callback to the function that must be done is made and all the participants execute this function. If it is needed to remove the shared events, the next code has to be called:

```
var a = gapi.hangout.data.getKeys(events);

for(i=0;i<a.length;i++) {

       gapi.hangout.data.clearValue(a[i]);

}
```

5. RESULTS OF AN EVALUATION SESSION

This chapter will explain the evaluation session done for the game editor, the obtained results and their impact: things to improve, to add and to discard. The evaluation session was done on July the 6th of 2012 at the HCI laboratory of the University of Patras. It included a demographic survey and a usability test done by me.

In order to obtain the realest results and not to influence the respondents of the survey, they did it in pairs (to test the collaborative mode) in a separated room. The number of respondents was finally 10 and they were from different countries (Spain, Greece, Turkey, Pakistan, etc) in order to have a variety of different people. Moreover, except two od the respondents, they were people without programming knowledge and completely outside of the project trying with this to detect ambiguities that closer people to the project wouldn't detect.

Before the evaluation session, the respondents read carefully the user guide provided in the 6th chapter of this document in order to place them into the context and give them a light idea of what were they going to do and how to use the application.

5.1. Usability testings[xii]

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system. This is in contrast with usability inspection methods where experts use different methods to evaluate a user interface without involving users.

Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose. Examples of products that commonly benefit from usability testing are foods, consumer products, web sites or web applications, computer interfaces, documents, and devices. Usability testing measures the usability, or ease of use, of a specific object or set of objects, whereas general human-computer interaction studies attempt to formulate universal principles.

5.1.1 Goals of usability testing

Usability testing is a black-box testing technique. The aim is to observe people using the product to discover errors and areas of improvement. Usability testing generally involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response. The results of the first test can be treated as a baseline or control measurement; all subsequent tests can then be compared to the baseline to indicate improvement.

- **Performance.** How much time, and how many steps, are required for people to complete basic tasks? (For example, find something to buy, create a new account, and order the item.)
- **Accuracy.** How many mistakes did people make? (And were they fatal or recoverable with the right information?)
- ***Recall*.** How much does the person remember afterwards or after periods of non-use?
- ***Stickiness*.** How much time he/she spends
- ***Emotional response*.** How does the person feel about the tasks completed? Is the person confident, stressed? Would the user recommend this system to a friend?

To assess the usability of the system under usability testing, quantitative and/or qualitative Usability goals (also called usability requirements) have to be defined beforehand. If the results of the usability testing meet the Usability goals, the system can be considered as usable for the end-users whose representatives have tested it.

5.1.2 Methods

Hallway testing

Hallway testing (or Hall Intercept Testing) is a general methodology of usability testing. Rather than using an in-house, trained group of testers, just five to six random people are brought in to test the product, or service. The name of the technique refers to the fact that the testers should be random people who pass by in the hallway.

Hallway testing is particularly effective in the early stages of a new design when the designers are looking for "brick walls," problems so serious that users simply cannot advance. Anyone of normal intelligence other than designers and engineers can be used at this point. (Both designers and engineers immediately turn from being test subjects into being "expert reviewers." They are often too close to the project, so they already know how to accomplish the task, thereby missing ambiguities and false paths).

Remote usability testing

In a scenario where usability evaluators, developers and prospective users are located in different countries and time zones, conducting a traditional lab usability evaluation creates challenges both from the cost and logistical perspectives. These concerns led to research on remote usability evaluation, with the user and the evaluators separated over space and time. Remote testing, which facilitates evaluations being done in the context of the user's other tasks and technology can be either synchronous or asynchronous. Synchronous usability testing methodologies involve video conferencing or employ remote application sharing tools such as WebEx. The former involves real time one-on-one communication between the evaluator and the user, while the latter involves the evaluator and user working separately.

Asynchronous methodologies include automatic collection of user's click streams, user logs of critical incidents that occur while interacting with the application and subjective feedback on the interface by users. Similar to an in-lab study, an asynchronous remote usability test is task-based and the platforms allow you to capture clicks and task times. Hence, for many large companies this allows you to understand the WHY behind the visitors' intents when visiting a website or mobile site. Additionally, this style of user testing also provides an opportunity to segment feedback by demographic, attitudinal and behavioural type. The tests are carried out in the user's own environment (rather than labs) helping further simulate real-life scenario testing. This approach also provides a vehicle to easily solicit feedback from users in remote areas quickly and with lower organisational overheads.

Numerous tools are available to address the needs of both these approaches. WebEx and Go-to-meeting are the most commonly used technologies to conduct a synchronous remote usability test. However, synchronous remote testing may lack the immediacy and sense of "presence" desired to support a collaborative testing process. Moreover, managing inter-personal dynamics across cultural and linguistic barriers may require approaches sensitive to the cultures involved. Other disadvantages include having reduced control over the testing environment and the distractions and interruptions experienced by the participants' in their native environment. One of the newer methods developed for conducting a synchronous remote usability test is by using virtual worlds.

Expert review

Expert review is another general method of usability testing. As the name suggests, this method relies on bringing in experts with experience in the field (possibly from companies that specialize in usability testing) to evaluate the usability of a product.

Automated expert review

Similar to expert reviews, automated expert reviews provide usability testing but through the use of programs given rules for good design and heuristics. Though an automated review might not provide as much detail and insight as reviews from people, they can be finished more quickly and consistently. The idea of creating surrogate users for usability testing is an ambitious direction for the Artificial Intelligence community.

5.1.3 How many users to test?

In the early 1990s, Jakob Nielsen, at that time a researcher at Sun Microsystems, popularized the concept of using numerous small usability tests—typically with only five test subjects each—at various stages of the development process. His argument is that, once it is found that two or three people are totally confused by the home page, little is gained by watching more people suffer through the same flawed design. "Elaborate usability tests are a waste of resources. The best results come from testing no more than five users and running as many small tests as you can afford". Nielsen subsequently published his research and coined the term heuristic evaluation.

The claim of "Five users is enough" was later described by a mathematical model which states for the proportion of uncovered problems U

$$U = 1 - (1 - p)^n$$

where p is the probability of one subject identifying a specific problem and n the number of subjects (or test sessions). This model shows up as an asymptotic graph towards the number of real existing problems.

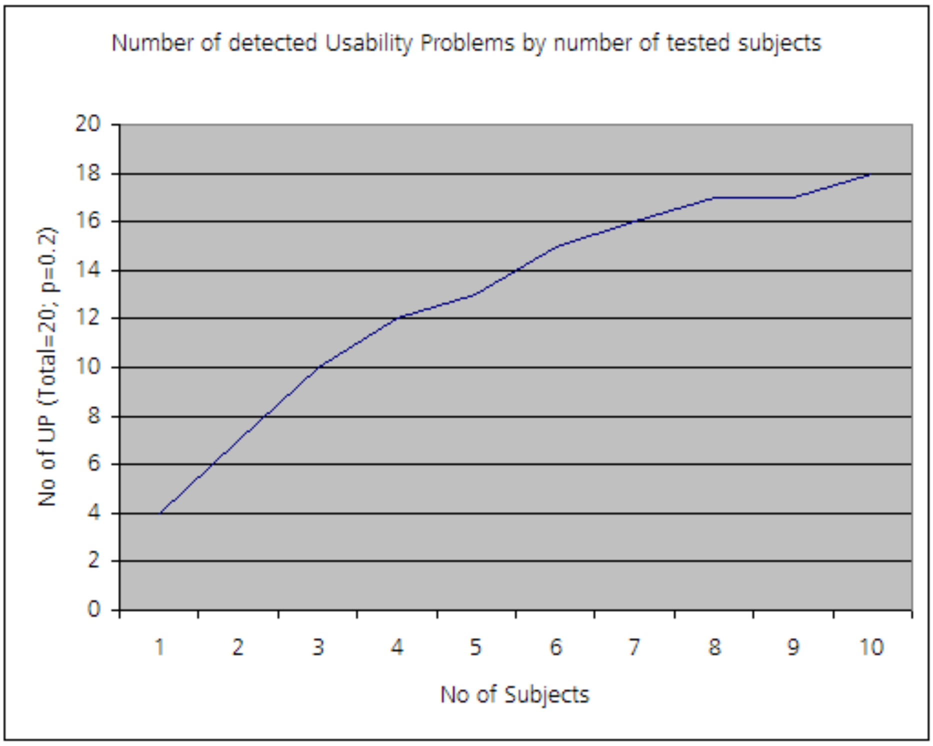


ILLUSTRATION 35. VIRZIS FORMULA

5.2. Testing results

The test took part on July the 6th of 2012 in the HCI laboratory. Ten people tested the application into the Google+ Hangouts: seven of them were erasmus students in the Patras University, two were greek students and one of the was a PHD candidate.

5.2.1 Demographic survey

This survey tries to show the level of knowledge of the technology used, languages they are fluent in and their knowledge of social networks.

Demographic and personal questions

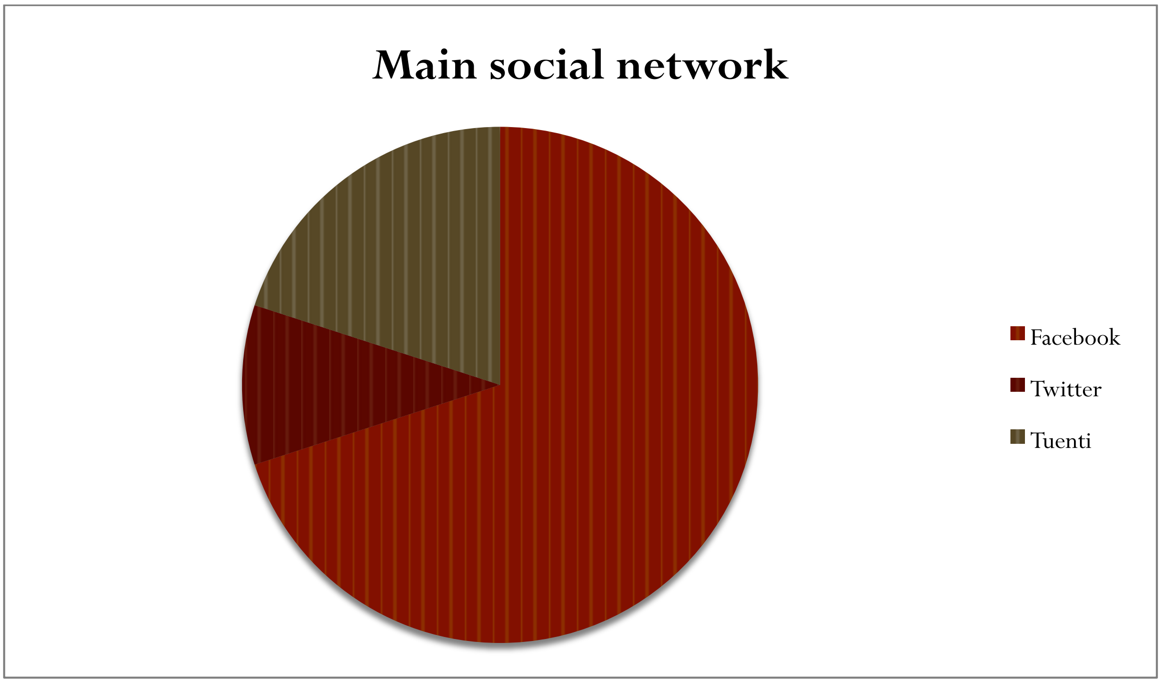
Seven of the respondents were erasmus students in the university of Patras (five spanish and two turkish). Two of them were computer-engineering students, three were aeronautic engineering students, and the other two were electrical engineering students.

Two of participants were greek students and their fields is Mathematics. And the last participant is a Pakistani student PHD candidate specialized in Phisics. Five of the respondents were male and five female and age range were between 20 and 29.

About the language, all the participants were fluent in english, five spoke spanish as their native language, two turkish, two greek and one pakistani but he was fluent also in greek.

Social networks

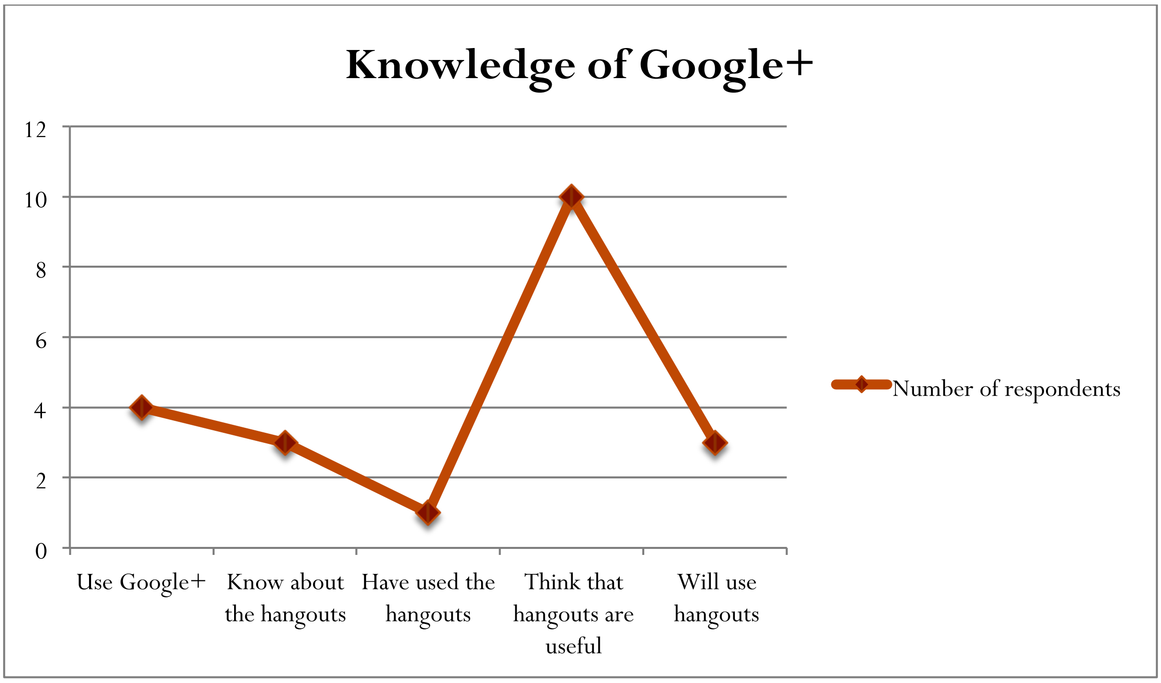
All the respondents feel very confortable using social networks and they use this technology everyday. All of them use Facebook as their main social network and they use it daily or almost daily, just 4 of them use Google+ once a week and 4. Seven of them have Twitter, but only 4 used it daily. About other social networks, 2 of them use Linkedin and 5 (the spanish) use Tuenti being this the main social network for 2 of them.



GRAPHIC 1 MAIN SOCIAL NETWORKS FOR THE RESPONDENTS

About the main use of their social networks, the half of them answered that they use social networks mainly to chat, 3 to post and the other 2 to view photos and comment them.

To the question "Did you know about the hangouts?", just three of them know of the hangouts and only one have used them one time. All of them answered that they think that hangouts are very useful but instead of this only three would use them usually. This result is a little bit disconcerting so I decided to ask them about it and usually they answered that they won't use the hangouts because they aren't planning to have Google+ account.



GRAPHIC 2 LEVEL OF KNOWLEDGE OF GOOGLE+

5.2.2 Application survey

After a test of the application where the participants must probe the editor in normal mode and in collaborative mode, they answered a survey. This survey tries to find how clear is the interface; the level of intuitiveness of the editor, how friendly is the layout, etc. in order to draw conclusions to improve the application, find bugs and correct them and what must be the next steps in the development of the application.

The layout

The questions about the layout try to find out how good is the general aspect of the application, if the size of the elements is suitable and how to do to improve the layout.

The general aspect of the layout has been evaluated between 1 and 10 being 1 the lowest (worst) and 10 the highest mark possible. The average mark has been 8.8 so it seems that people generally likes the layout.

About the size of the buttons, 100% of the respondents answered that it is appropriate and it shouldn't be changed, but when they are asked about the size of the map, 80% of them says that is not big enough and 90% of then answered that the infoBubble is very big and it take almost the whole map.

Moreover 70% of the interviewed answered that there were a lot of empty space that could be used to enlarge the size of the map.

Correctness

When the participants submitted the program to various basic tasks, almost all the times it worked as it was expected but two times it failed and this events allowed me to find two separate bugs that are fixed.

These bugs consisted in when names of the topics, elements or markers where written using double or simple brackets, the application didn't work as it was expected.

Usability

In order to evaluate the usability, the participants have been asked about how difficult did they find to create a new game, insert markers or add topics or elements. Also, they were asked about if they think if is it too difficult to find the information in the table and to identify the associations between markers and elements.

Three of them took more than 3 minutes to create a new game and in the survey the average difficulty obtained about this subject has been 6.5/10 what suggest that the way to create a new game should be changed.

The most difficult step to follow for the respondents was to place a new marker, six of them asked me how to do it because they didn't find the way to do it. In the survey, this subject obtains a mark of 7.2/10 of difficulty.

About create new topics or elements none of them found it hard and all of them answered that it was very intuitive.

Documentation

Before the test, all the participants were reported on what were they going to do, they received information about the application and its context and all of them received a copy of the user guide of the 6th chapter.

In the survey, all of them answered that they felt satisfied with the information received and that it was very useful when they were performing the text.

Scalability

Despite of the number of different programming languages used in the implementation, the code has been written in a module way separating in different files the languages in order to increase the scalability and easiness of add new functionalities or fix bugs.

Utility

Asked about the utility of the editor, they answered that it is a very good idea to have an editor to create games for this kind of application, because it provides the advantage to everybody of create their own games and releases the developers of the work to have to do something that can be done for the final or an intermediate user.

About the utility of the collaborative edition, they think that it could help to save time in the edition and to create better games.

5.3. Results of the test

Advantages

- **Intuitive in almost all the aspects.**
- **Correct,** except for small bugs**.**
- **Useful.**
- **Well documented.**
- **Scalable.**
- **Util.**

Disadvantages

- **Bugs.**
- **Hard to create a new game.**
- **Hard to place markers.**
- **Map very small.**
- **InfoBubbles very big.**
- Sometimes the information is moved due to the **display resolution.**

Future changes to enhance the user experience

1. Repair all the bugs found

2. Change the way to create a new game. The option new game will be removed from the game list and in its stead, it will be placed a new button on the top part of the application, close to the “Get Game” button.
3. Change the way to create markers. The button “New marker” will be removed and it will be added in the map a new button to drag and drop markers following the style of the Google Maps website, which is more intuitive.

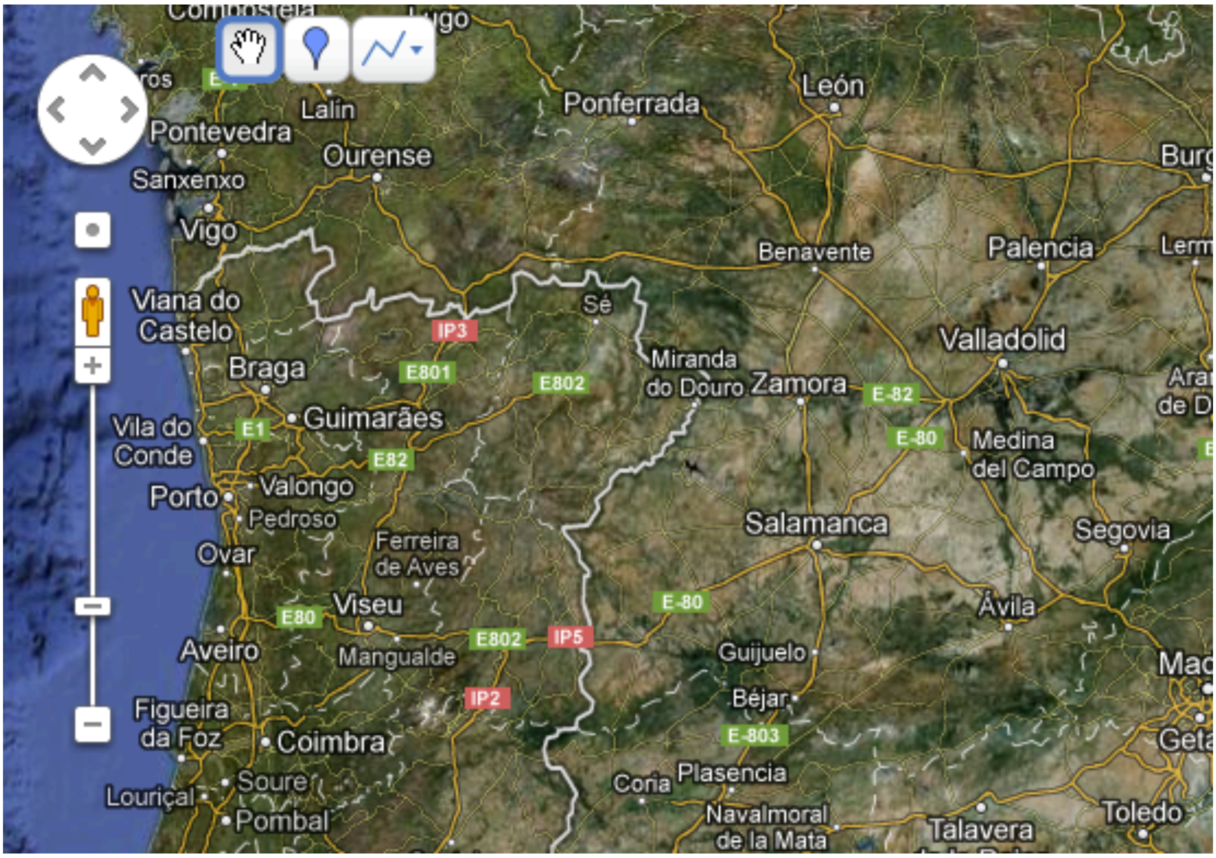


ILLUSTRATION 36. MAPS CREATOR FROM GOOGLE MAPS

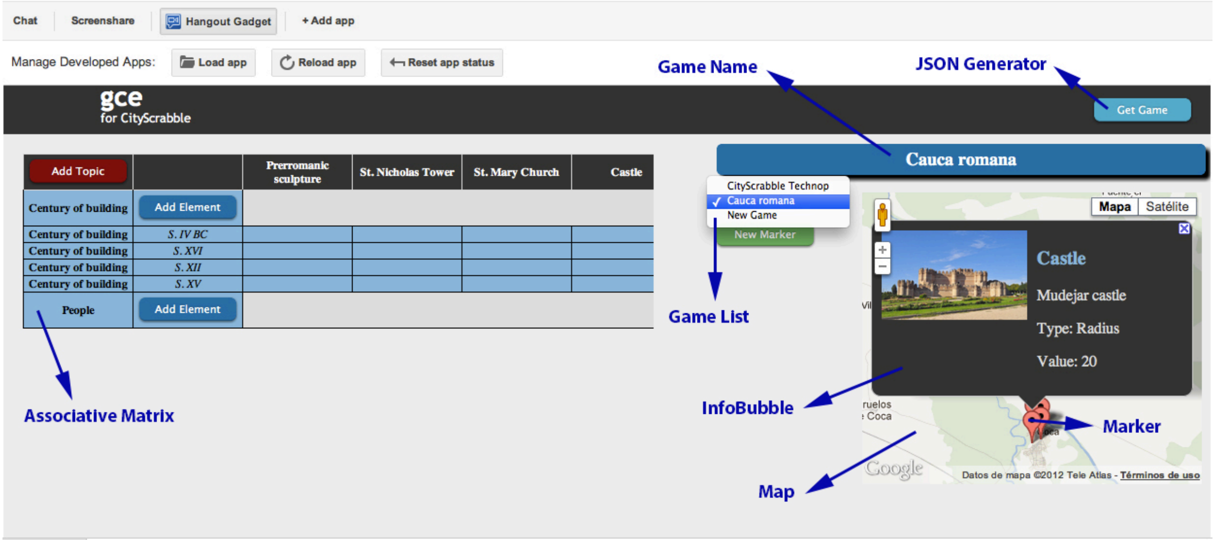
4. Enlarge the size of the map. The game list will be placed in the top of the site and the empty space will be used to enlarge the map.
5. Contract the infoBubbles. The size of the infoBubbles will be contracted and as the info that they content.
6. Repair the problems with screen resolutions different of 1280x800.
7. Implement a login system in order to detect whose user is the owner of which games.

6. USER GUIDE

DESIGN AND DEVELOPING OF A GAME EDITOR FOR CITYSCRABBLE APP

This chapter is a manual for the final user in order to let him know how to create, edit and delete games. Starting on the basis that the user knows how CityScrabble works and the kind of game he will develop, I will explain step by step all the phases creating, editing and deleting.

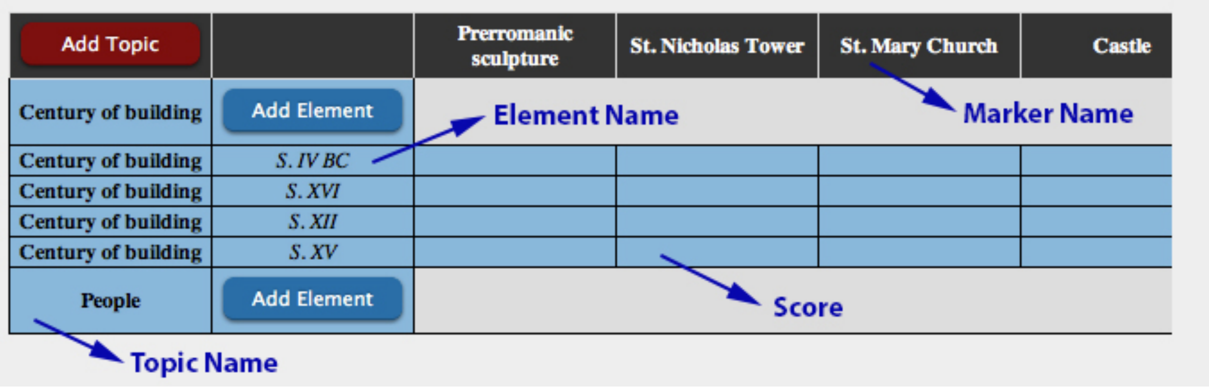
First of all, there are included two figures describing the interface:



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
Century of building	S. IV BC				
Century of building	S. XVI				
Century of building	S. XII				
Century of building	S. XV				
People	Add Element				

Castle
Mudejar castle
Type: Radius
Value: 20

ILLUSTRATION 37. GENERAL VIEW



Add Topic		Prerromanic sculpture	St. Nicholas Tower	St. Mary Church	Castle
Century of building	Add Element				
Century of building	S. IV BC				
Century of building	S. XVI				
Century of building	S. XII				
Century of building	S. XV				
People	Add Element				

ILLUSTRATION 38. MATRIX DESCRIPTION

6.1. Creating a new game

The process to create a game from the starting is the next:

1. Create the game.
2. Add all the markers of the game, which will save the positions of all the important location in the game.
3. Add all the topics that the game will have.
4. Add the elements of each topic.
5. Establish relationships between markers and elements adding the score.
6. Click on "Get Game" to get the JSON file.

6.1.1 How to create the game

If there aren't games saved in the database the menu to create the new game will appear automatically. In other case, follow the next instructions:

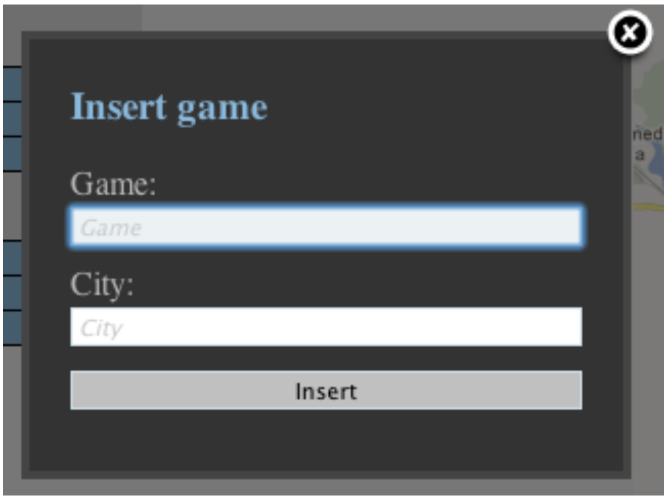
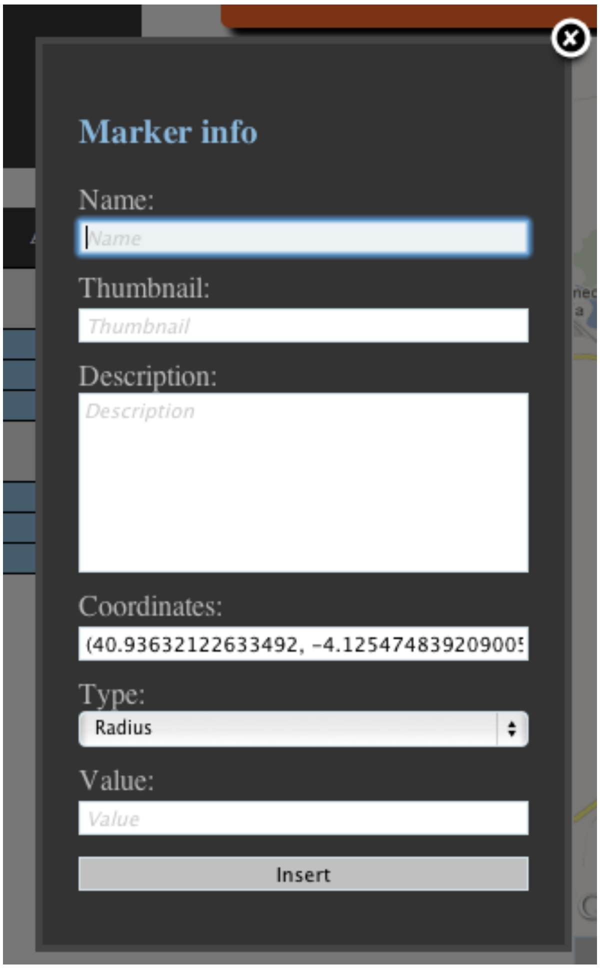


ILLUSTRATION 39. CREATING A NEW GAME

1. Click on the game list.
2. Select the option "New game".
3. It will appear a window like the left one. Please, fill in the fields and click "Insert"
4. The game is created. Let's edit it!

6.1.2 How to add markers



Coordinates: (40.93632122633492, -4.12547483920900

Type: Radius

Illustration 40. Adding markers

1. Click on “New Marker”
2. Click in the exact place of the map where we want the marker.
3. A window like the left one will appear. Fill in the fields and press “Insert”.
4. Done

Now the marker is visible in the map, clicking on it will appear all its information. Moreover, the table will appear.

6.1.3 How to add topics

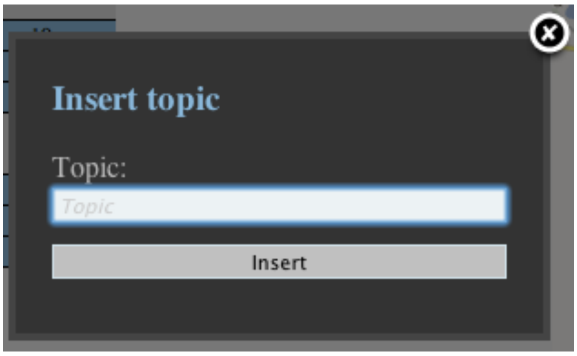


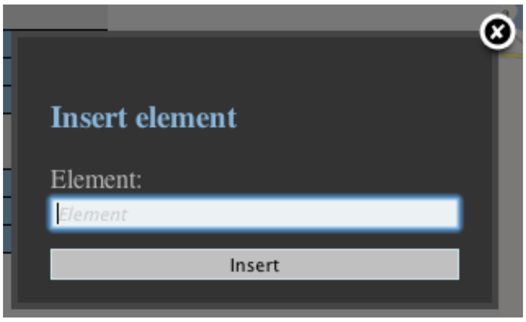
Illustration 41. Adding topics

1. Click on “Add topic” in the table.

2. A window like the left one will appear. Fill in the field and click “Insert”

3. Done. The topic will appear inside the table.

6.1.4 How to add elements


Insert element

Element:

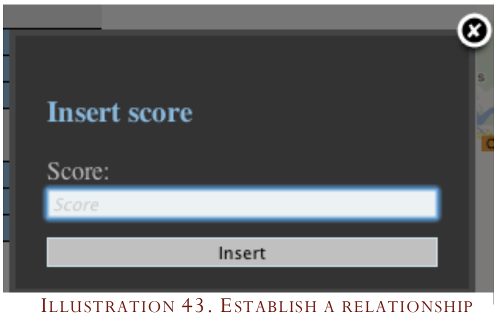
ILLUSTRATION 42. ADDING ELEMENTS

1. Click on “Add Element”.

2. A window like the left one will appear. Fill in the field and click “Insert”.

3. Done. The element will appear inside the table.

6.1.5 Establishing relationships


Insert score

Score:

ILLUSTRATION 43. ESTABLISH A RELATIONSHIP

1. Click on the cell of the relation.

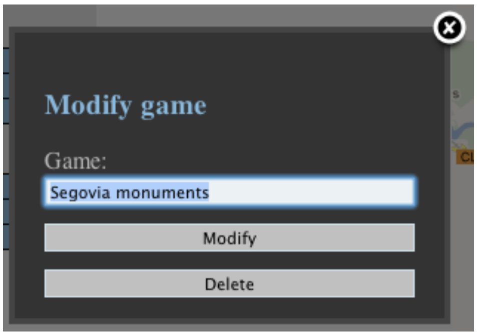
2. A window like the left one will appear. Fill in the field and click “Insert”.

3. Done. The score will appear in the cell.

6.2. Editing an existing game

First of all, select the game to edit in the game list. It will load.

6.2.1 Modifying the name of the game


Modify game

Game: Segovia monuments

ILLUSTRATION 44. MODIFYING THE GAME NAME

1. Click on the game title.

2. A window like the left one will appear. Change the name and click “Modify”.

3. Done. The change will be shown.

6.2.2 Modifying a marker

1. Click on the name of the marker in the table.

2. A window like the shown before for the markers will appear. Modify the fields and click “Modify”.

3. Done. The changes will be shown.

6.2.3 Modifying a topic

1. Click on the name of the topic in the table.

2. A window like the shown before for the topics will appear. Modify the field and click “Modify”.

3. Done. The changes will be shown.

6.2.4 Modifying an element

1. Click on the name of the element in the table.

2. A window like the shown before for the elements will appear. Modify the field and click “Modify”.

3. Done. The changes will be shown.

6.2.5 Modifying the score

1. Click on the cell of the relationship.

2. A window like the shown before for the score will appear. Modify the field and click “Insert”.

3. Done. The changes will be shown.

To delete any item follow the first step in any of the firsts 4 cases and click “Delete”. That item will be removed.

Conclusions

The idea of developing this application, started with a previous design, made by Anastasios Maroudas. The purpose of that application was the same, but the interface wasn't completely satisfactory or easy to understand for the final users.

After the work presented in this document it is valid to say that the prototype of the game editor is working appropriately and the main objectives have been achieved. There is a lot of work to do to achieve a final editor able to create scenarios with the complete characteristics of CityScrabble.

The storage of the data generated in the hangouts, the collaborative mode of editing, the generation of the games as a JSON string, etc. are implemented and working efficiently.

I have tried to use the most modern technologies available to do the application updating the code with every update of the API's. Also, I have tried to simplify the JavaScript programming using jQuery and to maximize the quickness of response of the stored data using the simplest orders to recover the information.

To implement the collaborative mode, it has been used the Google+ Hangout API. And the people who tested the application were very comfortable using this mode of editing, anyway there is a long work to do and the improving of this mode of editing must evolutionate.

One way for this evolution is to extend the location of the scenarios inside the buildings with the new functionalities of Google Maps which allows to map the indoor of buildings.

Finally, we can conclude that the objectives have been achieved and now it exists an application with a powerful evolution potential and because of this, there is a lot of work to do.

References

[i] http://en.wikipedia.org/wiki/Location-based_game

[ii] http://www.wisegeek.com/what-is-a-web-server.htm

[iii] http://searchsqlserver.techtarget.com/definition/database-management-system

[iv] http://www.webopedia.com/TERM/P/programming_language.html

[v] http://en.wikipedia.org/wiki/Cascading_Style_Sheets#Advantages

[vi] http://en.wikipedia.org/wiki/JQuery

[vii] http://www.ericmmartin.com/projects/simplemodal/

[viii] http://www.json.org/

[ix] http://en.wikipedia.org/wiki/XML#Pull_parsing

[x] http://en.wikipedia.org/wiki/Ajax_(programming)

[xi] http://www.w3.org/DOM/#what

[xii] http://en.wikipedia.org/wiki/Usability_testing

DOCUMENTS FOR THE EVALUATION SESSION

The purpose of this evaluation session is to test the Gamescape Content Editor made for CityScrabble games. CityScrabble is a kind of Scrabble game where several players can play in a real location. This kind of game is called location-based. The game can be located in all around the world. Each player will be able to choose a topic about which he plays in every moment, and about this topic he must answer a group of hints (called elements in the editor). If the player answers correctly a hint, it will add the score value of the hint to his score and all the players will be informed about it in real time.

INSTRUCTIONS FOR THE FIRST PART

1. Read carefully the provided user guide.
2. Start a new Hangout session.
3. Create a game with the next parameters (fill in the required information as you want):
   a. At least 3 markers
   b. At least 2 topics
   c. At least 2 elements per topic
   d. Fill in the score cells as your election
4. Change the name of the game.
5. Change the parameters of a marker.
6. Select a different game.

INSTRUCTIONS FOR THE SECOND PART (COLLABORATIVE)

1. Find a partner to work with.
2. One of you will start a new Hangout session (main editor) and the other one will join (joined editor).
3. The main editor will select the game “Cauca romana”.
4. The joined editor will click over a marker in the map.
5. The main editor will add a new topic called “Used as”
6. The main editor will add the next elements to the new topic:
   a. High school
   b. Music school
7. The joined editor will add the next element to the new topic:
   a. Graveyard
8. The joined editor will fill the scores as follows:

	Castle	**St. Nicholas Tower**	**St. Mary Church**
High school	50	0	0
Music school	0	0	50
Graveyard	20	60	25

SURVEY

Point the next questions between 1 (lowest), 10 (highest):

	1	2	3	4	5	6	7	8	9	10
How **difficult** is to create a 'New Game'?										
How **intuitive** is to put a new marker?										
How **clear** is the information in the table?										
How **intuitive** is to create Topics or Elements?										
Evaluate the layout (aspect of the editor) (10 best)										
Evaluate the **utility** of collaborative edition										

Answer Yes or No:

	Yes	No
Do you think that there is a lot of empty space?		
Is the size of the buttons appropriate?		
Is the map big enough?		
Is the InfoBubble too big?		
Do you like the general aspect?		
Do you find useful a game editor for CityScrabble?		
Do you think that collaborative edition saves time?		
Is it easy to find information inside the application?		
Was the user guide easily understandable and useful?		

What things do you think that can be improved? :