Technical documentation

Alpha Game Builder

EPITECH.
L'ECOLE DE L'INNOVATION ET DE
L'EXPERTISE INFORMATIQUE

Summary

This paper aims to present the architecture design of the Alpha Game Builder. It presents a comprehensive diagram of the project, a detailed diagram, communication diagram, business logic, the model of the database and the choice of technologies for which we optées.

Alpha Game Builder can be broken down into three major modules (library, tools and the web platform), we will endeavor to develop a unique perspective to each.

Document description

Title	2014_TD9_EN_alphagamebuilder
Date	31/03/2014
Author	Bruno QUILGHINI
e-mail	quilgh_a@epitech.eu
Subject	Technical Documentation
Version	1.0

Updates table

Date	Author	Section(s)	Comment
03/31/2014	quilgh_b	All	Basic version
03/31/2014	quilgh_b	All	Complete overhaul of the template document
03/31/2014	quilgh_b	IV. a.	Writing explanatory comment
03/31/2014	quilgh_b	V. a.	Writing explanatory comment
03/31/2014	quilgh_b	VI.	Writing explanatory comment
03/31/2014	quilgh_b	VII.	Writing explanatory comment
03/31/2014	quilgh_b	VIII.	Writing explanatory comment
03/31/2014	quilgh_b	IX.	Value adjustment following the completion of the project
03/31/2014	quilgh_b	XI.	Adjustment following the completion of the project
03/31/2014	quilgh_b	XII.	Adjustment following the completion of the project
03/31/2014	quilgh_b	XIII.	Creation
03/31/2014	quilgh_b	XIV.	Creation

Table of contents

I. Introduction

This user documentation was realized for the Alpha Game Builder project under the Innovative Projects Epitech (EIP) promotion EPITECH 2014.

a. The EIP

The EIP is the proposed end of the course students EPITECH. Carried out on two years, it allows students to apply all the skills acquired in their training, both technologically and in terms of project management. Involving a minimum of six students, this project is realized with a professional approach.

The EPITECH, school innovation and IT expertise, apart from other IT training for its practical approach to the problems faced by students. Whereas the appreciation of a student is not in knowledge but in its ability to innovate and find solutions in real situations, the training EPITECH enjoys strong popularity among businesses.

b. Project description

The Alpha Game Builder aims to allow users to drastically reduce development time using the online tools, based on a library HTML5/Javascript.

The project objective is to encourage the creation of 2D PC games browsers HTML5 to create a true community platform between developers and players, both to learn, test, play and share with other users. The library actually serves to support the implementation of the web platform, in this sense.

Free to use, it is the entrepreneurs without a team eager to simply produce a demo, young programmers to learn about the videogame programming and advanced developers to optimize their creation time.

II. Representation of the overall architecture

This first version of the architecture review revolves around the three main functional parts of the Alpha Game Builder namely:

- Library - Allows you to build the engine of the future play through preset functions
- Utilities - editor worlds and the generator and manager graphical elements allow the user to manage both its graphics data but also the creation of the universe
- The web platform - Allows the promotion of different games created from the Alpha Game Builder and facilitates access to these

We also present a number of perspectives:

- The global perspective: Will using a Use Case describes the different scenarios of the project.
- The logical point of view: In order to present the overall design of the solution.
- The process perspective: Decomposition into tasks via a sequence diagram.
- The deployment perspective: Browsing the physical structures necessary to deploy the project.
- The view of the database: Overview of the structure of the database.

III. Architecture, purpose and constraints

a. Objectives having a specific impact on architecture

The two major objectives of the Alpha Game Builder is to allow easy access to the programming of 2D video games using HTML5 and promote projects created from our solution.

In order to achieve the realization of these objectives, it was necessary to divide the project into three main entities, each providing a portion of our needs. The heart of the project could be likened to the library with the most features for the creation of a game, however, keeping in mind our initial objectives, it was necessary to develop utilities for facilitating access to this type of programming and also to implement a web platform designed to promote both our solution but also games produced therewith.

Architecture into three parts thus flows from the upstream reflection of user needs.

b. Functional constraints

Inherent in the Alpha Game Builder functional constraints are divided into three main parts, each corresponding to one of the above entities of the project.

Functional constraints related to the library:

- **The user needs to create different types of games.**

The library is built around a system of code generation / templates depending on the type of game that wants to create the user. It may well have pre-generated code for an arcade game that differs from a pre-generated code for a strategy game.

- **The user needs to create multiplayer games.**

With NodeJS, we offer the user an overlay allowing him to easily manage all network problems in his game

- **The user will have the least amount of code to produce possible.**

Code generation based on the type of game is to achieve a gain non- negligible time for both the amateur and the professional programmer programmer.

- **The user will be able to fully customize the project.**

The pre-generated code that offers the Alpha Game Builder lays the foundation of the game type selected. It is then the programmer to create its own rules and to include its models to achieve its prototype. In fact, every project is different both graphically and functionally. We only generate common ground necessary for the implementation of more elaborate structures.

Functional constraints utilities:

- **The user will be able to access tools that facilitate the creation of the project.**

One of the main issues of the Alpha Game Builder is to allow the programmer does not have to constantly switch between different utilities . In fact , access to a publisher worlds as well as a manager of graphics data are that the user will not have to seek additional solutions for the realization of his project.

Functional constraints in web platform:

- **The user needs to be able to get the online library.**

The web platform will enable one-click downloading the entire library.

- **The user will be able to test games from our solution.**

Acting as a display, the web platform will provide an overview of each project , classified by genre.

- **The user will be able to promote his project.**

Each project from Alpha Game Builder will be submitted to the community via the web platform.

- **The user will have a community to help in case of problems.**

Putting up a web platform, along with a forum to allow each user to quickly make contact with other programmers using the Alpha Game Builder. The project team will also be held at the disposal of users via a contact system and regular updating of a FAQ.

c. Non-functional constraints

Here is a list of non-functional constraints related to the Alpha Game Builder:

- **Compatibility with various internet browsers**

It is necessary to ensure that the project is compatible with the user all current major web browsers as well as the most used versions (IE - Firefox - Chrome - Opera - Safari).

- **Compatibility with the new HTML5**

Need to keep abreast of new language specification.

- **Support for different languages**

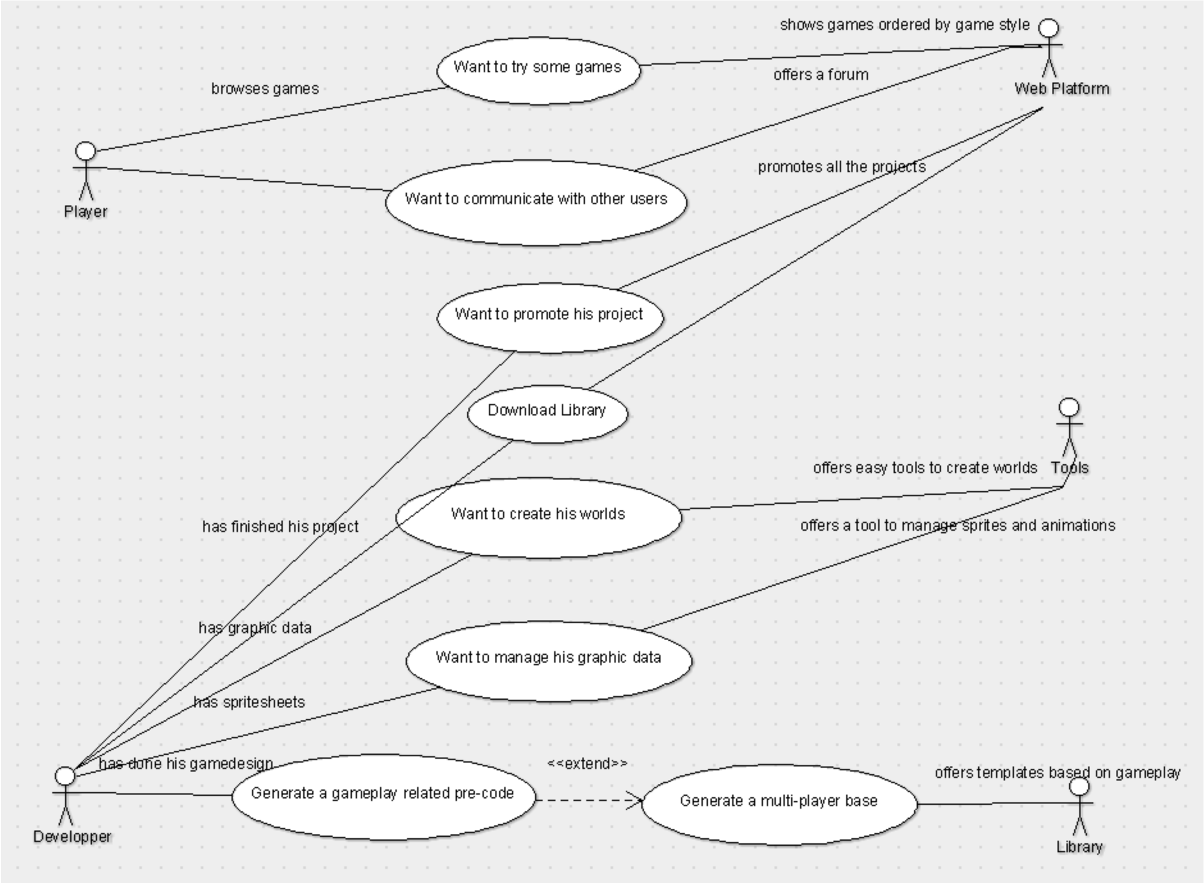
Need to present the project in both French and English.

- **The need to stand out from other solutions**

In the continuity of project creation, always stay alert compared to other solutions in order to stay always innovative to the market.

IV. Overview of project

a. Main use cases


shows games ordered by game style
Want to try some games
browses games
offers a forum
Web Platform
Player
Want to communicate with other users
promotes all the projects
Want to promote his project
Download Library
offers easy tools to create worlds
Tools
Want to create his worlds
has finished his project
offers a tool to manage sprites and animations
has graphic data
Want to manage his graphic data
has spritesheets
has done his gamedesign
<<extend>>
offers templates based on gameplay
Generate a gameplay related pre-code
Generate a multi-player base
Developper
Library

Main view of the main use cases of the Alpha Game Builder

The objective of the scheme of the main use cases is to illustrate all the interactions between the different entities interacting within our project.

There are two types of users of the Alpha Game Builder: **player and developer**.

The player interacts with the web platform. Through it, he can try different games and communicate with users connected.

The developer has complex interactions within the project. His first action will be to recover the library via the web platform. It can then access the library to create a game using pre-generated code by game type. Once the code created templates they can use the tools to manage all the graphic parts of his game (editor chopper worlds and sprites). Once the full game it will be promoted through web platform.

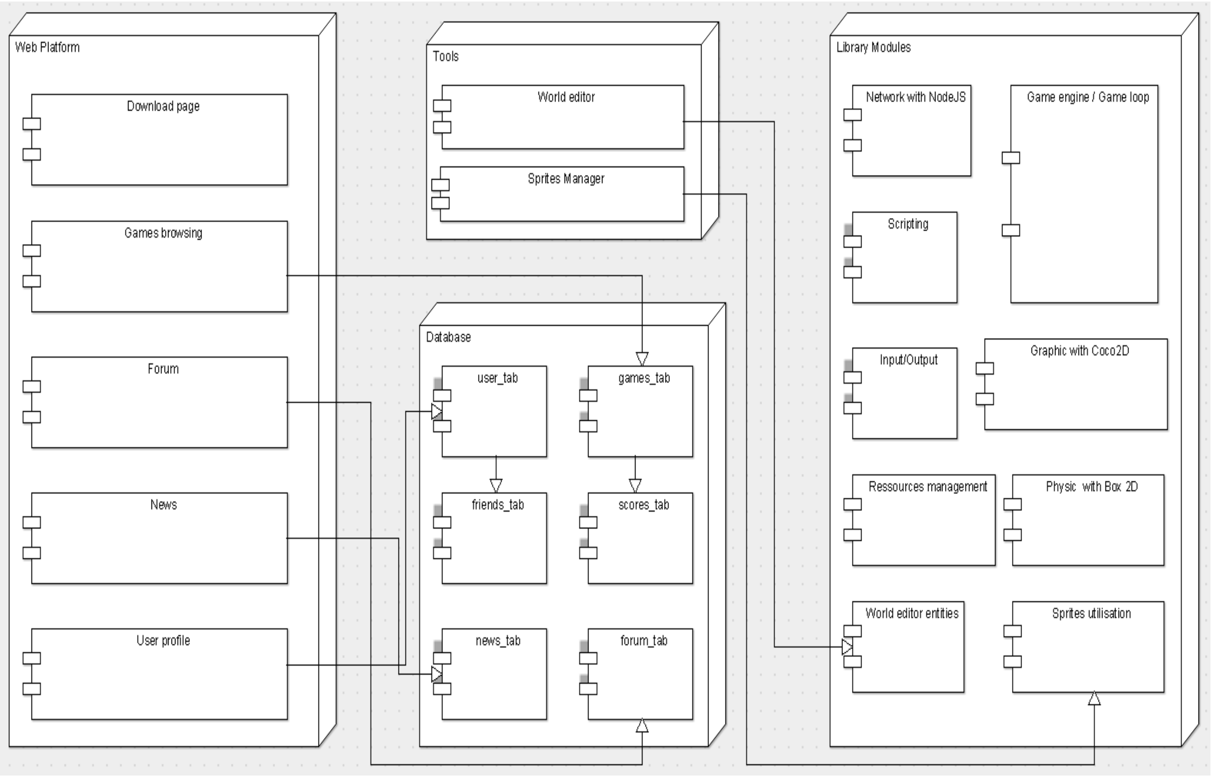
b. Detailed use cases

Alpha Game Builder is aimed at two types of users: the programmer and the player. Basically, the player wants to test the games created with our solution and possibly communicate his feelings with other users.

The programmer is also a player. In this sense, it takes the same needs as the classic player. However, he also has the desire to design games from the Alpha Game Builder. In fact, he will want to download the library first and then use it to generate code for the concept of his game will also have access during the development phase to the various tools at its disposal including in the field of managing graphics resources. Finally, he will be able to promote his product.

V. Logical view of the application

a. Global view



Global view of the Alpha Game Builder global view

The purpose of the overview is to understand the division of large project entities and their components.

There are four main entities within the application:

- The web platform
- Tools
- Databases
- The library modules

Each of these entities meets a specific goal. The web platform , the main interface available to users, allowing them to access the download page for the solution, the news about the latest games created in consultation with the various games or to view the profile of other users. The web platform is directly related to the different databases.

There are two tools in the Alpha Game Builder editor worlds and Chopper sprites. Both are connected to their inherent resources and which are stored in the library module .

Databases allow storage of information to the proper functioning of the web platform. They are also in contact with the web platform.

Finally libraries modules combine different resources to the production of the project while allowing the storage of tools related resources.

b. Main components

The only three existing components we use are listed in the library in the diagram above . These are :

• **NodeJS** to the web server for ease of implementation and the innovative nature of this new technology.

• **Cocos2D** for managing graphics and data abstraction for managing the graphics engine. Cocos2D is the only existing library proven .

• **Box2D** to manage the physics engine . Box2D is free and is the only existing library proven in the field of physical management .

• **BackboneJS** to apply a MVC client side Javascript .

Regarding the other components , they are fully developed and are cut into four large entities. The first of these is the Web platform that consists of:

- The download page
- The page path games
- The Forum
- The news page
- User profiles

The second entity is utilities. It is composed of:

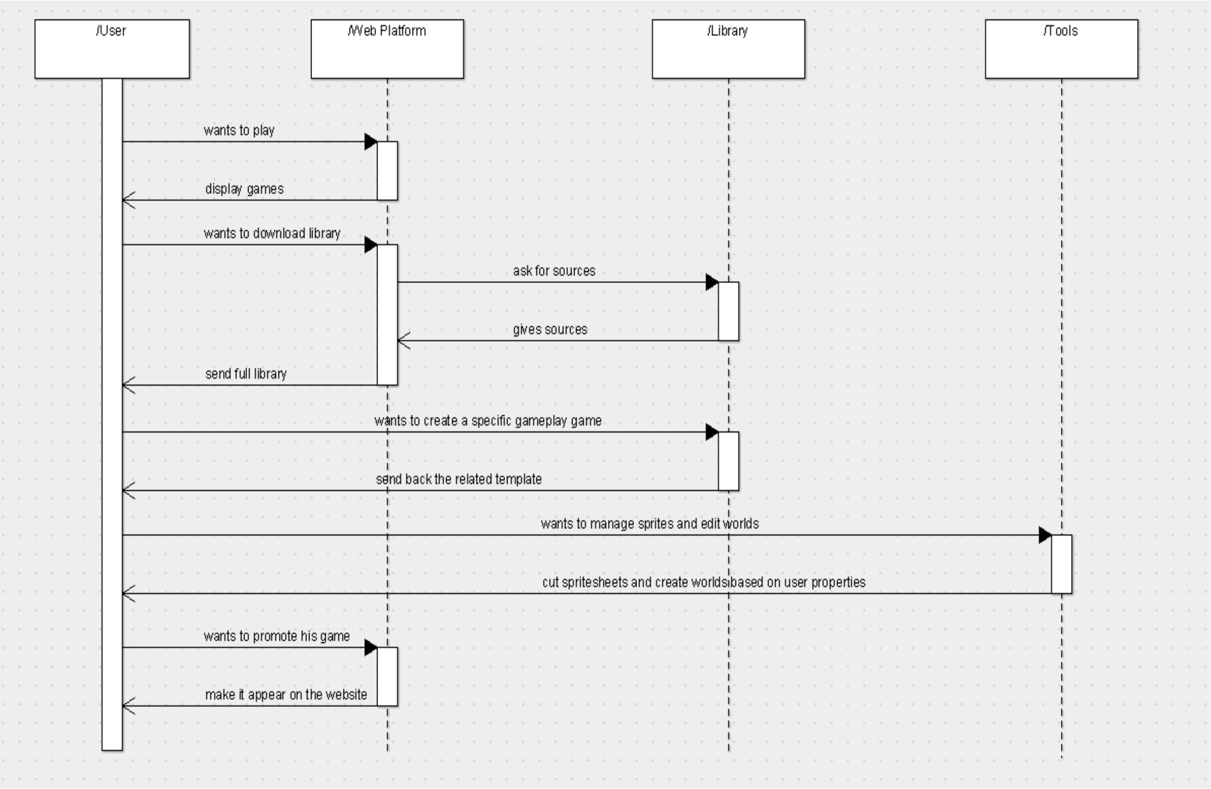
- The editor worlds
- The manager of graphics resources

The third entity is the library itself. Its components are:

- The script module
- management module I / O
- The module resource management
- The handling module graphics data generated by utility linked
- The management module entities creator of worlds

Finally, based on data presented in this document can be found .

VI. Process view


/User
/Web Platform
/Library
/Tools
wants to play
display games
wants to download library
ask for sources
gives sources
send full library
wants to create a specific gameplay game
send back the related template
wants to manage sprites and edit worlds
cut spritesheets and create worlds based on user properties
wants to promote his game
make it appear on the website

Process view of the Alpha Game Builder

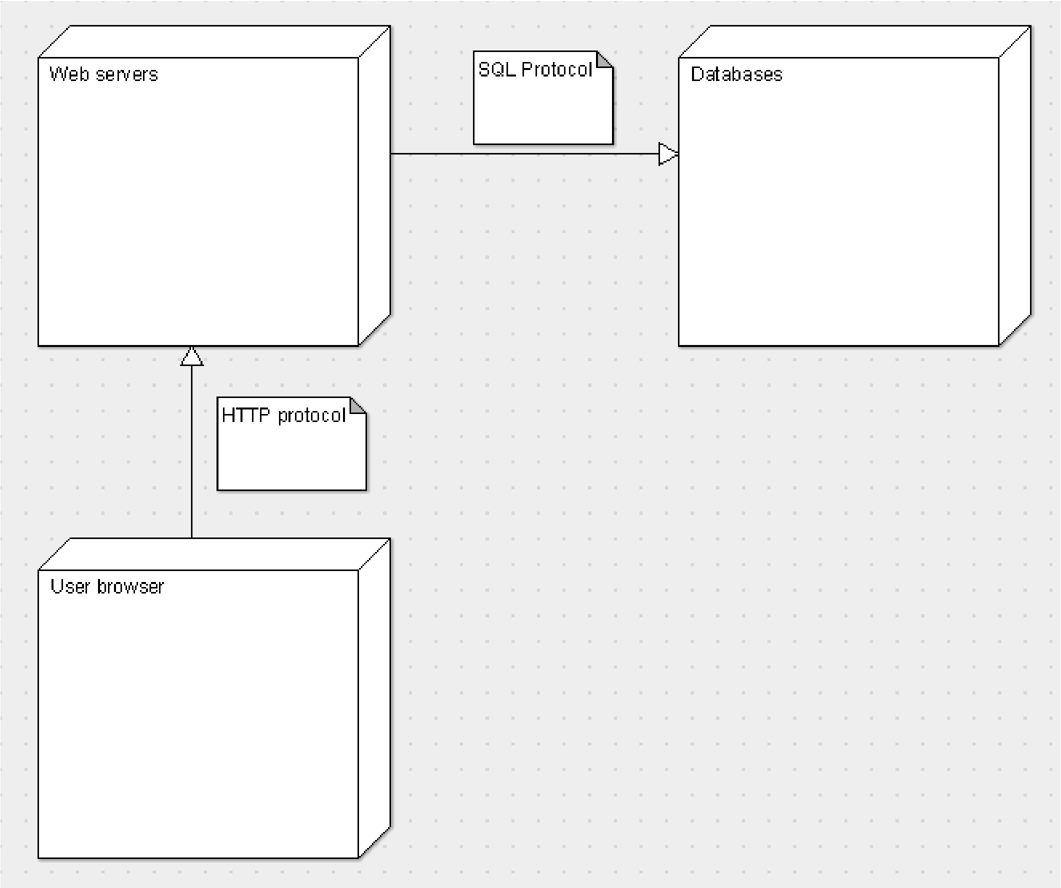
The objective of the process is to highlight the relationship between the different processes and tasks of the program.

In the diagram above five singular cases can be distinguished:

- The user wants to see or play games. It then passes directly through the web platform.
- The user wants to retrieve the sources of the library. He asks the web platform which returns sources.
- The user wants to create a style of play in particular. It then suffices to recover the corresponding template in the library sources.
- The user wants to manage external resources of his game at this time can turn to tools such as the creator of worlds or chopper sprites.
- The user wants to promote his game. In this case, the web platform can present it publicly.

All of these cases define the interactions between the four entities of the Alpha Game Builder user, the web platform, the library and tools.

VII. Deployment view



Deployment view of the Alpha Game Builder

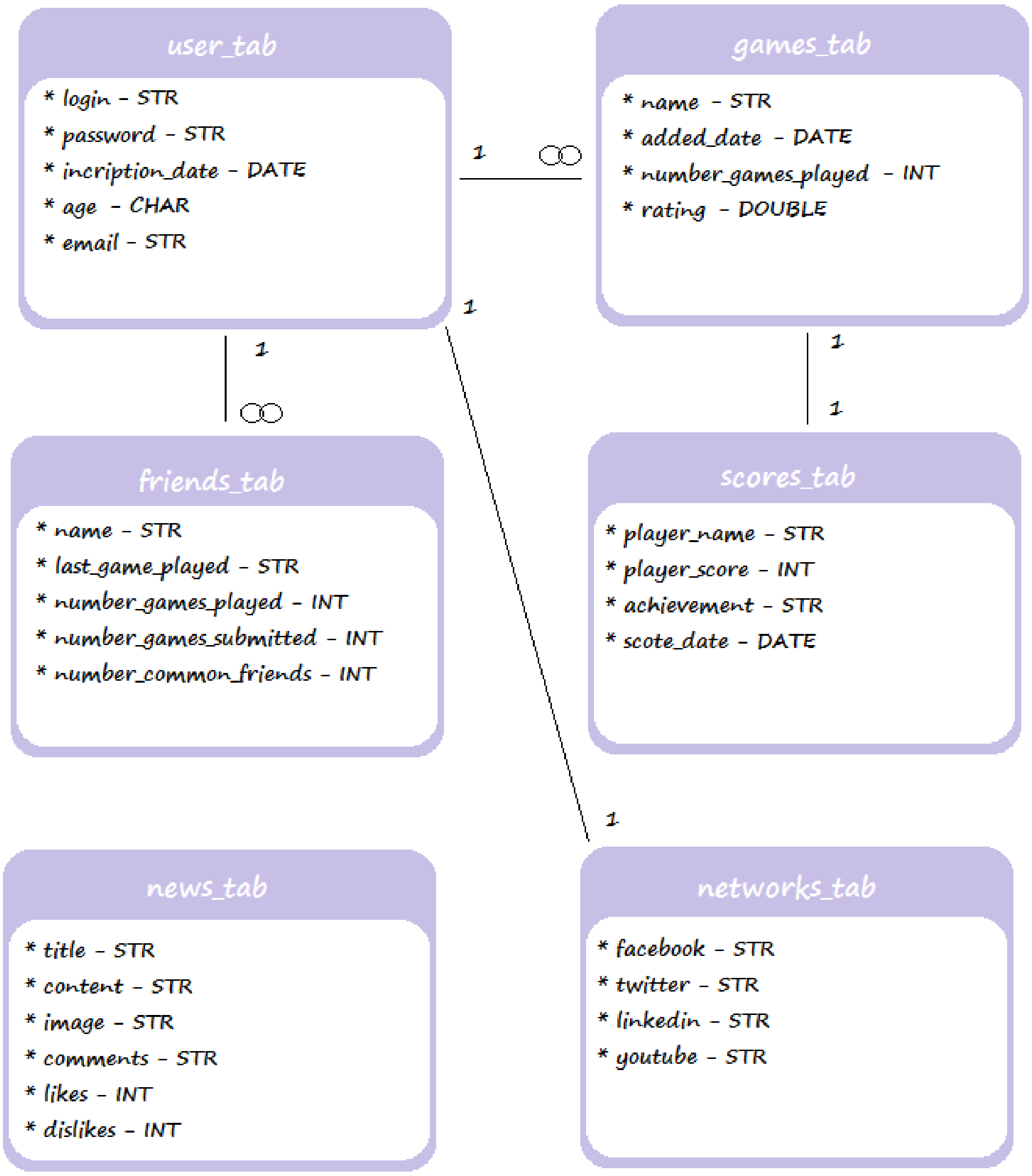
Given the model we use, the deployment view is very simple. There are only three major players involved. The user via the internet explorer will interact with the web server itself will fetch the information in the databases.

The interest of keeping a simple model is to allow the development team to manage a minimum of protocols. Focusing on improving them, it leads to more effective communication of the various entities of the solution.

The objective of minimizing the exchange of information is also gaining performance. More protocols and the number of entities are reduced, the faster management.

In the structure presented above, the web browser of the user communicates directly with the web server through a http protocol , while the web server has its data in the database using a SQL protocol.

VIII. Data view



Data view of the Alpha Game Builder

The structure of the database of the Alpha Game Builder remained small despite the size of the project. We found it necessary to retain access to all data to allow optimal management of the development team simplicity.

By taking a small structure database, it is easier to centralize important information and interact with the different poles up the team.

There are relationships between tables that are at the heart of how the Alpha Game Builder:

- The user_tab representing a user can have an unlimited number of games_tab representing them different games created.
- Each set (games_tab) is linked to a scoring table (scores_tab).

- Each user (user_tab) can have an unlimited number of friends (friends_tab).
- The user (user_tab) can register their social networks within the networks_tab.
- The news_tab is independent.

IX. Size and performance

When the output of the Alpha Game Builder, we strive number 50 developers and 500 regular players to ensure the heart of community. Never having had the opportunity to meet many users, we can now and after our unit tests, estimate that our solution can fit up to 1000 simultaneous players.

We estimate the number of queries on the site averages 50 requests / minutes / user for a user browsing the site. We estimate the number of simultaneous connected to 100 maximum. This would give a maximum of 2000 queries / minute.
The size of each request should not exceed 800 bytes.

The size of the database, it should not exceed 70 MB

Finally, the response time in case of overcrowding should be 0.1 second maximum.

X. Quality

Qualitatively, we rely on an analysis of three main criteria namely: scalability, reliability and portability.

Regarding scalability, it is not expected to add new content to that provided for the project output. The issue of scalability does not arise in the case of Alpha Game Builder.

In terms of reliability, we plan to maintain servers and databases throughout the life of the project. We will also regularly update the documentation based on frequent user questions. The team will present on forums related to the web platform for a facilitated communication with users.

Finally, the web server will be portable to Windows, MacOS and Linux and compatible databases with MySQL, PostgreSQL and SQLite.

However, it may be noted that these sensitivities are impacting in any way the security of the solution or user data.

XI. API and interfaces

At the launch of the final version of the Alpha Game Builder, we have created or interfaces or APIs.

Indeed, the objective of the project is to propose a solution directly usable via the tools located on the web platform. The creation of API and interfaces becomes superfluous.

XII. Well-known bugs

All bugs appeared during development have been solved for the final version of the Alpha Game Builder.

XIII. Coding style

The whole product code on the Alpha Game Builder follows the coding style proposed by google and presented in detail on the following link:

https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml

In addition, each complex function is commented to facilitate reading by a third party.

By adopting a strict and documented existing standard, we hope to facilitate the handling and improve the readability of our solution.

XIV. Releases notes

Our project has been made available to the public at the time of its completion under the EIP module.

So there are no release notes since the only version presented is version 1.0.