

Documentation technique

Alpha Game Builder

Résumé

Ce document a pour but de présenter la conception de l'architecture de l'Alpha Game Builder. Y sont présentés un diagramme global du projet, un diagramme détaillé, un diagramme de communication, la logique métier, le modèle de la base de données ainsi que le choix des technologies pour lesquelles nous avons optées.

L'Alpha Game Builder se décomposant en trois modules majeurs (la bibliothèque, les outils ainsi que la plateforme web), nous nous attacherons à développer un point de vue particulier pour chacune d'elles.

Description du document

Titre	2014_TD9_FR_alphagamebuilder
Date	31/03/2014
Auteur	Bruno QUILGHINI
e-mail	quilgh_a@epitech.eu
Sujet	Documentation technique
Version du modèle	1.0

Tableau des révisions

Date	Auteur	Section(s)	Commentaire
31/03/2014	quilgh_b	Toutes	Version de base
31/03/2014	quilgh_b	Toutes	Refonte complète du template du document
31/03/2014	quilgh_b	IV. a.	Rédaction commentaire explicatif
31/03/2014	quilgh_b	V. a.	Rédaction commentaire explicatif
31/03/2014	quilgh_b	VI.	Rédaction commentaire explicatif
31/03/2014	quilgh_b	VII.	Rédaction commentaire explicatif
31/03/2014	quilgh_b	VIII.	Rédaction commentaire explicatif
31/03/2014	quilgh_b	IX.	Ajustement des valeurs suite à la finalisation du projet
31/03/2014	quilgh_b	XI.	Ajustement suite à la finalisation du projet
31/03/2014	quilgh_b	XII.	Ajustement suite à la finalisation du projet
31/03/2014	quilgh_b	XIII.	Création
31/03/2014	quilgh_b	XIV.	Création

Table des matières

I.	Introduction	1
a.	<i>Rappel de l'EIP</i>	<i>1</i>
b.	<i>Description du projet</i>	<i>1</i>
II.	Représentation de l'architecture globale	2
III.	Architecture, buts et contraintes	3
a.	<i>Objectifs spécifiques ayant un impact sur l'architecture</i>	<i>3</i>
b.	<i>Contraintes fonctionnelles</i>	<i>3</i>
c.	<i>Contraintes non fonctionnelles</i>	<i>4</i>
IV.	Vue globale du projet	5
a.	<i>Use Cases principaux</i>	<i>5</i>
b.	<i>Uses Cases détaillés</i>	<i>5</i>
V.	Vue logique de l'application	7
a.	<i>Vue globale</i>	<i>7</i>
b.	<i>Composants principaux</i>	<i>8</i>
VI.	Vue processus	9
VII.	Vue déploiement	10
VIII.	Vue données	11
IX.	Taille et performance	13
X.	Qualité	14
XI.	API et interfaces	15
XII.	Liste des bugs techniques connus	16
XIII.	Coding style	17
XIV.	Releases notes	18

I. Introduction

Cette documentation utilisateur a été réalisée autour du projet Alpha Game Builder dans le cadre des Epitech Innovative Projects (EIP) de la promotion EPITECH 2014.

a. Rappel de l'EIP

L'EIP est le projet de fin de cursus des étudiants de l'EPITECH. Réalisé sur deux années, il permet aux étudiants de mettre en application l'ensemble des compétences acquises au sein de leur formation, tant au niveau technologique qu'au niveau de la gestion de projet. Réunissant un minimum de six étudiants, ce projet est réalisé avec une approche professionnelle.

L'EPITECH, l'école de l'innovation et de l'expertise informatique, se démarque des autres formations informatiques par son approche concrète des problématiques rencontrées par ses étudiants. Considérant que la plus-value d'un étudiant ne réside pas dans ses connaissances mais dans sa capacité à innover et à rechercher des solutions en situation réelle, la formation EPITECH jouit d'une forte popularité au sein des entreprises.

b. Description du projet

L'Alpha Game Builder a pour vocation de permettre à ses utilisateurs de réduire drastiquement leurs temps de développement aux moyens d'outils accessibles en ligne, se basant sur une bibliothèque HTML5/Javascript.

L'objectif du projet est d'encourager la création de jeux navigateurs PC 2D en HTML5 afin de créer une véritable plateforme communautaire entre développeurs et joueurs, à la fois pour apprendre, tester, jouer et partager avec les autres utilisateurs. La bibliothèque sert véritablement de support à la mise en place de la plateforme web, en ce sens.

Utilisable gratuitement, il vise les entrepreneurs sans équipe désireux de pouvoir produire simplement une démo, les jeunes programmeurs de s'initier à la programmation vidéo-ludique et les développeurs confirmés d'optimiser leurs temps de création.

II. Représentation de l'architecture globale

Cette première version du bilan d'architecture s'articule autour des trois grandes parties fonctionnelles de l'Alpha Game Builder à savoir :

- **La bibliothèque** – Permet de construire le moteur du futur jeu au travers de fonctions préétablies
- **Les utilitaires** – L'éditeur de mondes et le générateur et gestionnaire d'éléments graphiques permettent à l'utilisateur de gérer à la fois ses données graphiques mais aussi la création de son univers
- **La plateforme web** – Permet la promotion des différents jeux créés à partir de l'Alpha Game Builder et facilite l'accès à ceux-ci

Nous présenterons également un certain nombre de points de vue :

- **Le point de vue global** : Permettra au moyen d'un Use Case de présenter les différents scénarios du projet.
- **Le point de vue logique** : A pour but de présenter le design global de la solution.
- **Le point de vue processus** : Décomposition en tâches via un diagramme de séquence.
- **Le point de vue déploiement** : Parcours des structures physiques nécessaires au déploiement du projet.
- **Le point de vue de la base de données** : Présentation de la structure de la base de données.

III. Architecture, but et contraintes

a. Objectifs spécifiques ayant un impact sur l'architecture

Les deux objectifs majeurs de l'Alpha Game Builder sont de permettre un accès simplifié à la programmation de jeux vidéo 2D via HTML5 et d'assurer la promotion des projets créés à partir de notre solution.

Afin de parvenir à la réalisation de ces deux objectifs, il était nécessaire de découper le projet en trois grandes entités assurant chacune une partie de nos besoins. Le cœur du projet pourrait s'apparenter à la bibliothèque comprenant la majorité des fonctionnalités pour la réalisation d'un jeu. Néanmoins, gardant en tête nos objectifs initiaux, il était nécessaire de concevoir des utilitaires permettant la facilitation d'accès à ce type de programmation et également de mettre en place une plateforme web destinée à la promotion à la fois de notre solution mais également des jeux produits avec celle-ci.

L'architecture en trois parties découle donc de la réflexion en amont des besoins utilisateurs.

b. Contraintes fonctionnelles

Les contraintes fonctionnelles inhérentes à l'Alpha Game Builder sont divisées en trois grandes parties correspondant chacune à une des entités susmentionnées du projet.

Contraintes fonctionnelles liées à la bibliothèque :

- **La nécessité utilisateur de pouvoir créer différents types de jeux.**

La bibliothèque s'articule autour d'un système de génération de code/templates en fonction du type de jeu que souhaite créer l'utilisateur. Il pourra ainsi avoir du code pré-généré pour un jeu d'arcade qui diffèrera d'un code pré-généré pour un jeu de stratégie.

- **La nécessité utilisateur de pouvoir créer des jeux multi-joueurs.**

Grâce à NodeJS, nous offrons à l'utilisateur une surcouche lui permettant de gérer facilement toute la problématique réseau au sein de son jeu.

- **La volonté utilisateur d'avoir le moins de code à produire possible.**

La génération de code en fonction du type de jeu à réaliser est un gain de temps non-négligeable à la fois pour le programmeur amateur et pour le programmeur professionnel.

- **La volonté utilisateur de pouvoir personnaliser au maximum son projet.**

Le code pré-généré que propose l'Alpha Game Builder pose les bases du type de jeu choisi. C'est ensuite au programmeur de créer ses propres règles et d'inclure ses modèles pour réaliser son prototype. De fait, chaque projet est différent à la fois graphiquement et fonctionnellement. Nous ne faisons que générer des bases communes nécessaires à la mise en place de structures plus élaborées.

Contraintes fonctionnelles liées aux utilitaires :

- **La volonté utilisateur de pouvoir accéder à des outils facilitant la création de son projet.**

Un des principaux enjeux de l'Alpha Game Builder est de permettre au programmeur de ne pas avoir à perpétuellement naviguer entre différents utilitaires. De fait, l'accès à un éditeur de mondes ainsi qu'à un gestionnaire de données graphiques font que l'utilisateur n'aura pas à chercher de solutions complémentaires pour la réalisation de son projet.

Contraintes fonctionnelles liées à la plateforme web :

- **La nécessité utilisateur de pouvoir se procurer la bibliothèque en ligne.**

La plateforme web permettra en un clic de pouvoir télécharger l'ensemble de la bibliothèque.

- **La volonté utilisateur de pouvoir tester des jeux issus de notre solution.**

Faisant office de présentoir, la plateforme web offrira une présentation de chaque projet, classé par genre.

- **La volonté utilisateur de pouvoir promouvoir son projet.**

Chaque projet réalisé à partir de l'Alpha Game Builder pourra être soumis à la communauté via la plateforme web.

- **La volonté utilisateur d'avoir une communauté pour l'aider en cas de problème.**

Le fait de mettre en place une plateforme web, accompagnée d'un forum permettra à chaque utilisateur de pouvoir rapidement prendre contact avec les autres programmeurs utilisant l'Alpha Game Builder. L'équipe de réalisation du projet se tiendra également à la disposition des utilisateurs via un système de contact et à l'actualisation régulière d'une FAQ.

c. Contraintes non-fonctionnelles

Voici une liste des contraintes non-fonctionnelles reliées à l'Alpha Game Builder :

- **Compatibilité avec les différents explorateurs internet**

Il est nécessaire de faire en sorte que le projet de l'utilisateur soit compatible avec l'ensemble des explorateurs internet majeurs actuels ainsi qu'avec les versions les plus utilisées (I.E – Firefox – Chrome – Opéra – Safari).

- **Compatibilité avec les nouvelles spécifications HTML5**

Nécessité de se tenir informés des nouvelles spécifications du langage.

- **Support de différents langages**

Importance de pouvoir présenter le projet à la fois en français et en anglais.

- **Nécessité de se démarquer des autres solutions**

Dans la continuité de la création du projet, toujours rester alertes par rapport aux autres solutions afin de rester toujours innovants par rapport au marché.

IV. Vue globale du projet

a. Use cases principaux



Vue des principaux use cases de l'Alpha Game Builder

L'objectif du schéma des use cases principaux est d'illustrer l'ensemble des interactions entre les différentes entités interagissant au sein de notre projet.

Il existe deux types d'utilisateurs de l'Alpha Game Builder : **le joueur et le développeur**.

Le joueur n'interagit qu'avec la plateforme web. A travers elle, il peut essayer différents jeux et communiquer avec les utilisateurs connectés.

Le développeur a des interactions plus complexes au sein du projet. Sa première action sera de récupérer la bibliothèque via la plateforme web. Il peut ensuite accéder à la bibliothèque afin de créer un jeu au moyen des templates pré-générés par type de jeu. Une fois son code créé, il peut utiliser les outils afin de gérer l'ensemble des parties graphiques de son jeu (éditeur de mondes et découpeur de sprites). Une fois son jeu complet il pourra le promouvoir grâce à la plateforme web.

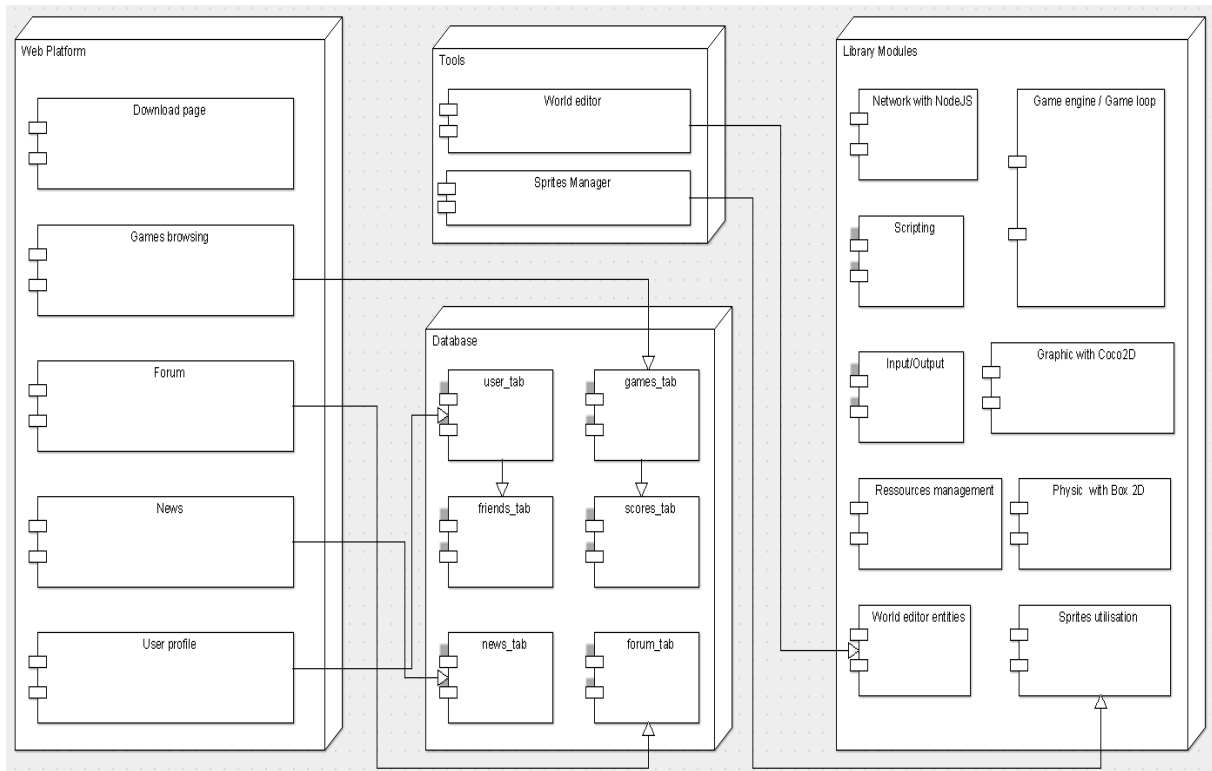
b. Use cases détaillés

L'Alpha Game Builder s'adresse à deux grands types d'utilisateurs : le programmeur et le joueur. Basiquement, le joueur souhaite tester les jeux créés via notre solution et éventuellement communiquer son ressenti avec les autres utilisateurs.

Le programmeur est également un joueur. En ce sens, il reprend les mêmes besoins que le joueur classique. Cependant, il a également la volonté de concevoir des jeux à partir de l'Alpha Game Builder. De fait, il va souhaiter télécharger la bibliothèque dans un premier temps puis l'utiliser afin de générer du code pour le concept de son jeu. Il aura également accès durant sa phase de développement aux différents outils mis à sa disposition notamment dans le domaine de la gestion des ressources graphiques. Enfin, il aura la possibilité de promouvoir son produit.

V. Vue logique de l'application

a. Vue globale



Vue globale de l'application Alpha Game Builder

L'objectif de la vue globale est de permettre de comprendre la division des grandes entités du projet ainsi que leurs composants.

Il existe quatre grandes entités au sein de l'application :

- La plateforme web
- Les outils
- Les bases de données
- Les bibliothèques de modules

Chacune de ces entités répond à un objectif précis. La plateforme web, principale interface mise à la disposition des utilisateurs, leur permet d'accéder à la page de téléchargement de la solution, aux news concernant les derniers jeux créés, à la consultation des différents jeux ou encore à la visualisation du profil des autres utilisateurs. La plateforme web est directement liée aux différentes bases de données.

Il existe deux outils au sein de l'Alpha Game Builder : l'éditeur de mondes et le découpeur de sprites. Tous deux sont reliés aux ressources qui leur sont inhérentes et qui sont stockées dans les bibliothèques de modules.

Les bases de données permettent le stockage des informations destinées au bon fonctionnement de la plateforme web. Elles ne sont d'ailleurs en contact qu'avec la plateforme web.

Enfin les bibliothèques de modules regroupent les différentes ressources nécessaires pour la production du projet tout en permettant le stockage des ressources liées aux outils.

b. Composants principaux

Les trois seuls composants déjà existants que nous utilisons sont mentionnés dans la bibliothèque dans le schéma ci-dessus. Ce sont :

- **NodeJS** pour le serveur web pour la facilité d'implémentation et le caractère innovant de cette technologie nouvelle.
- **Cocos2D** pour la gestion des données graphiques et s'abstraire de la gestion du moteur graphique. Cocos2D est la seule librairie existante ayant fait ses preuves.
- **Box2D** pour la gestion du moteur physique. Box2D est gratuite et est la seule librairie existante ayant fait ses preuves dans le domaine de la gestion physique.
- **BackboneJS** pour appliquer un modèle MVC en Javascript coté client.

Pour ce qui est des autres composants, ils sont entièrement créés et sont découpés au sein de quatre grandes entités. La première d'entre elles est la **Plateforme Web** qui est composée de :

- La page de téléchargement
- La page de parcours des jeux
- Le forum
- La page de news
- Les profils utilisateurs

La seconde entité correspond aux **utilitaires**. Elle est composée de :

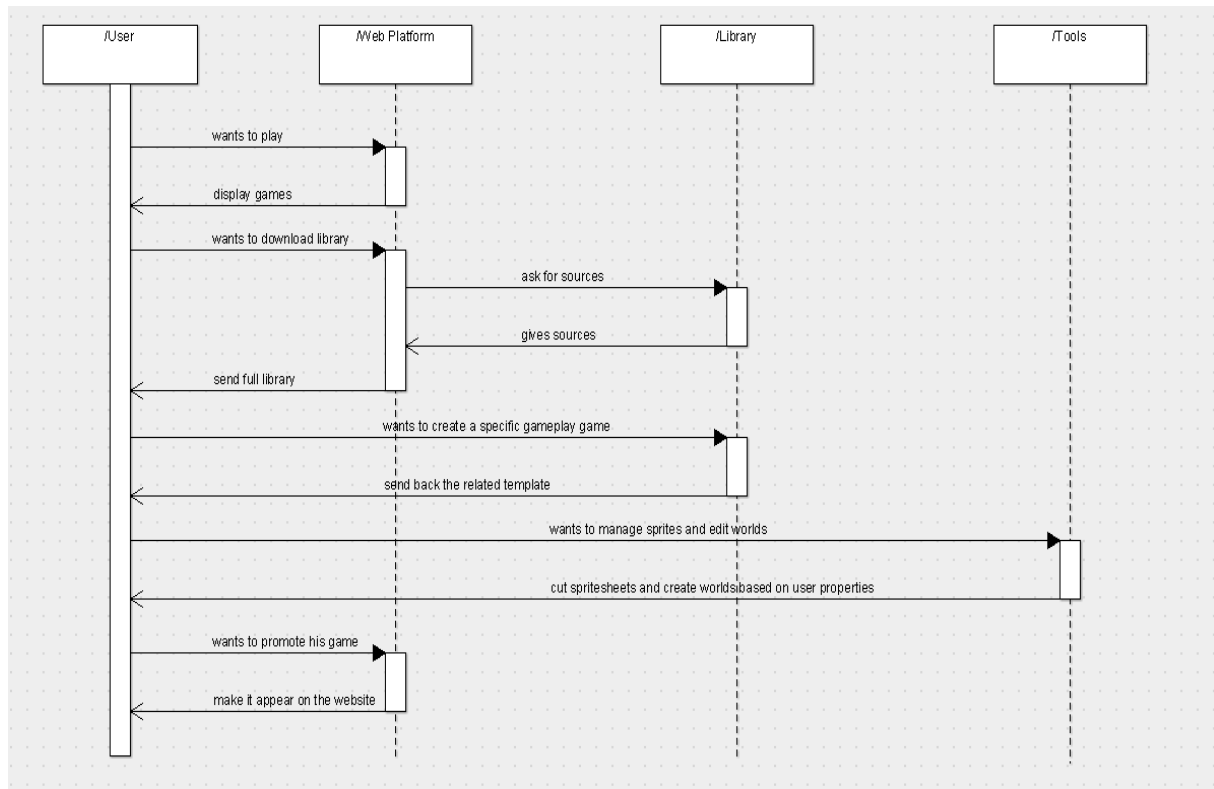
- L'éditeur de mondes
- Le gestionnaire de ressources graphiques

La troisième entité est la **bibliothèque** en elle-même. Ses composants sont :

- Le module de script
- Le module de gestion d'entrées/sorties
- Le module de gestion des ressources
- Le module de manipulation des données graphiques générées par l'utilitaire lié
- Le module de gestion des entités du créateur de mondes

Pour finir, on trouve la base de données présentées plus bas dans ce document.

VI. Vue processus



Vue processus de l'Alpha Game Builder

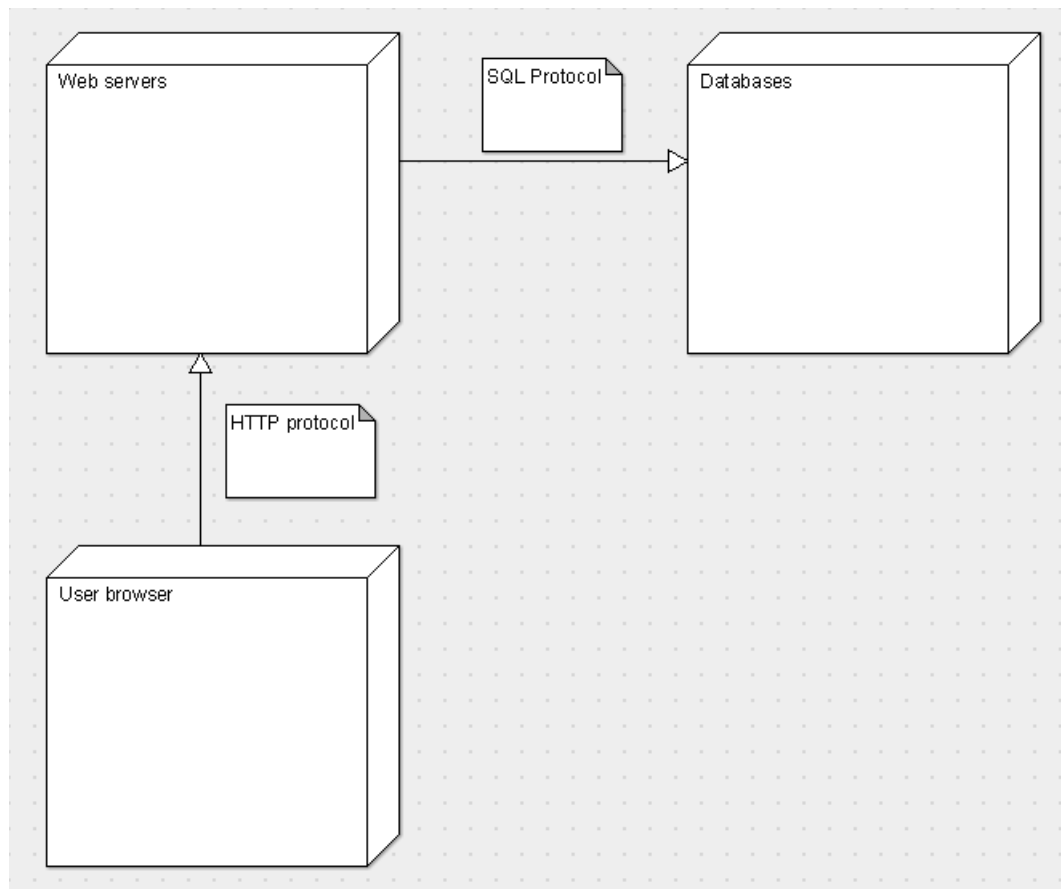
L'objectif de la vue processus est de mettre en avant la relation entre les différents processus et tâches du programme.

Dans le diagramme ci-dessus on distingue cinq cas singuliers :

- L'utilisateur souhaite jouer ou consulter des jeux. Il passe alors directement par la plateforme web.
- L'utilisateur souhaite récupérer les sources de la bibliothèque. Il la demande à la plateforme web qui lui renvoie les sources.
- L'utilisateur souhaite créer un style de jeu en particulier. Il lui suffit alors de récupérer le template correspondant dans les sources de la bibliothèque.
- L'utilisateur souhaite pouvoir gérer les ressources externes de son jeu. Il peut à ce moment se retourner vers les outils tels que le créateur de mondes ou le découpeur de sprites.
- L'utilisateur souhaite promouvoir son jeu. Dans ce cas, la plateforme web peut le présenter de manière publique.

L'ensemble de ces cas définit les interactions entre les quatre grandes entités de l'Alpha Game Builder : l'utilisateur, la plateforme web, la bibliothèque et les outils.

VII. Vue déploiement



Vue déploiement de l'Alpha Game Builder

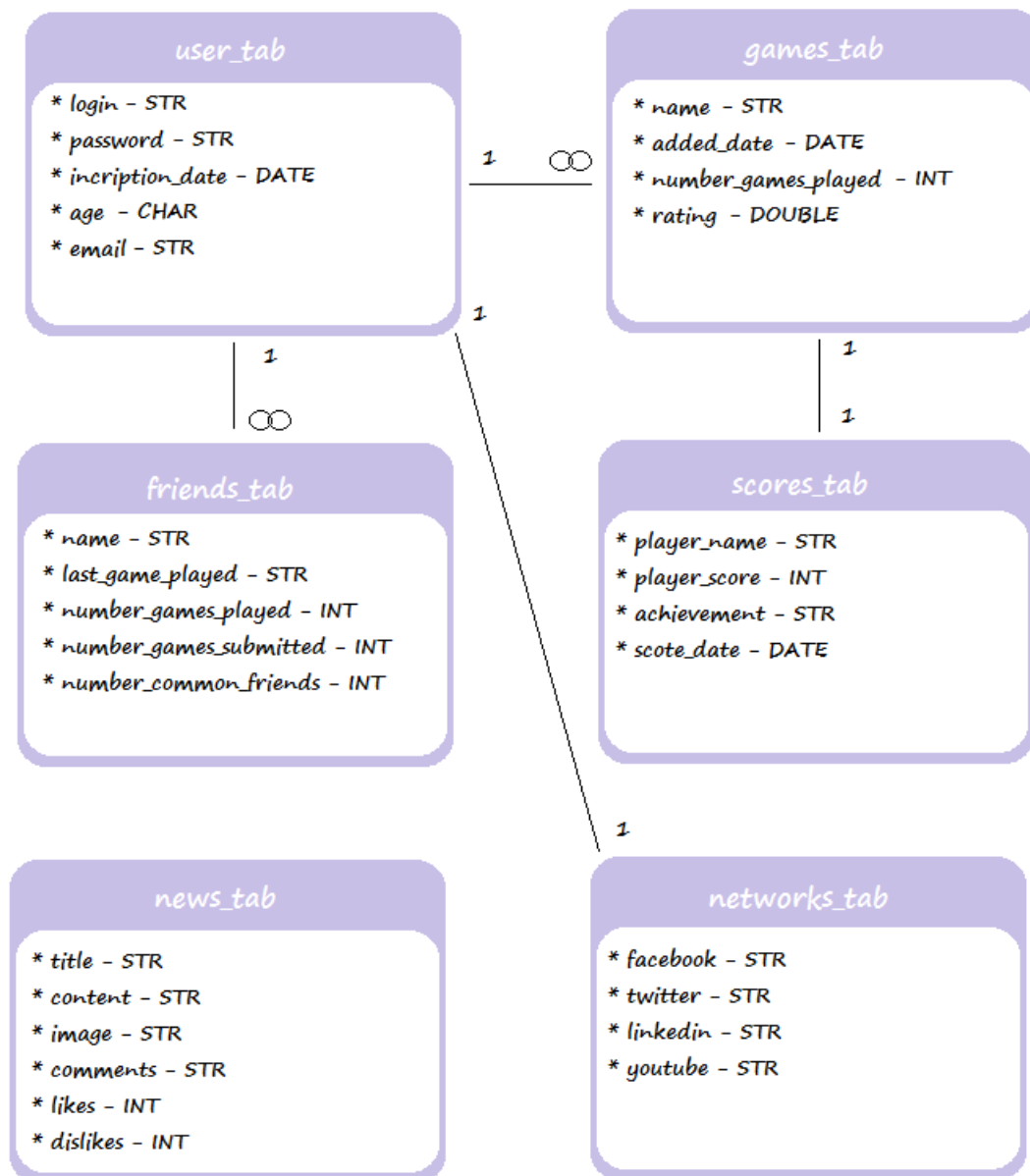
Compte tenu du modèle que nous utilisons, la vue de déploiement est très simple. Il n'y a que trois grands acteurs qui interviennent. L'utilisateur via son explorateur internet va interagir avec le serveur web qui lui-même ira chercher les informations dans les bases de données.

L'intérêt de garder un modèle simple est de permettre à l'équipe de développement de gérer un minimum de protocoles. En se concentrant sur l'amélioration de ceux-ci, on débouche sur une communication plus efficace des différentes entités de la solution.

L'objectif de minimiser les échanges d'informations est également de gagner en performance. Plus les protocoles et le nombre d'entités sont réduits, plus rapide est leur gestion.

Dans la structure présentée ci-dessus, le navigateur web de l'utilisateur communique directement avec le serveur web grâce à un protocole http, tandis que le serveur web puise ses données dans les bases de données au moyen d'un protocole SQL.

VIII. Vue données



Vue de la structure de la base de données de l'Alpha Game Builder

La structure de la base de données de l'Alpha Game Builder est restée légère malgré la taille du projet. Nous avons jugé nécessaire de conserver une simplicité d'accès à l'ensemble des données afin de permettre une gestion optimale de l'équipe de développement.

En adoptant une petite structure de base de données, il est plus facile de centraliser les informations importantes et d'interagir entre les différents pôles composant l'équipe.

Il existe des relations entre les tables qui sont au cœur du fonctionnement de l'Alpha Game Builder :

- La user_tab représentant un utilisateur peut avoir un nombre illimité de games_tab représentant eux les différents jeux créés.
- Chaque jeu (games_tab) est lié à un tableau de scores (scores_tab).
- Chaque utilisateur (user_tab) peut avoir un nombre illimité d'amis (friends_tab).
- L'utilisateur (user_tab) peut enregistrer ses réseaux sociaux au sein de la networks_tab.
- La news_tab est indépendante.

IX. Taille et performance

Lors de la sortie de l'Alpha Game Builder, nous visions un nombre de 50 développeurs et de 500 joueurs réguliers afin d'assurer un cœur de communauté. N'ayant jamais eu l'occasion de réunir autant d'utilisateurs, nous pouvons aujourd'hui et d'après nos tests unitaires, estimer que notre solution peut faire tenir jusqu'à 1000 joueurs simultanés.

Nous estimons le nombre de requêtes moyennes sur le site à 50 requêtes /minutes/utilisateur pour un utilisateur parcourant le site. Nous estimons le nombre de connectés simultanés à 100 maximum. Cela donnerait un nombre maximal de 2000 requêtes/minute.

La taille de chaque requête ne devrait pas excéder 800 octets.

La taille de la base de données, elle, ne devrait pas excéder 70 Mo.

Pour terminer, le temps de réponse lors d'une action utilisateur devrait donc être de 0.1 seconde au maximum en cas de forte affluence.

X. Qualité

Au niveau qualitatif, nous nous basons sur une analyse de trois critères principaux à savoir : l'extensibilité, la fiabilité ainsi que la portabilité.

Concernant l'extensibilité, il n'est pas prévu d'ajouter de nouveau contenu à celui prévu pour la sortie du projet. La question de l'extensibilité ne se pose donc pas dans le cas de l'Alpha Game Builder.

Au niveau de la fiabilité, nous prévoyons d'assurer la maintenance des serveurs ainsi que des bases de données durant toute la durée de vie du projet. Nous mettrons également régulièrement à jour la documentation en fonction des questions fréquentes des utilisateurs. L'équipe sera présente sur les forums liés à la plateforme web pour une communication facilitée avec les utilisateurs.

Enfin, le serveur web sera portable sur Windows, MacOS et Linux et les bases de données compatibles avec MySQL, PostgreSQL et SQLite.

En revanche, on peut noter que ces points sensibles n'impactent en rien la sécurité de la solution ou des données utilisateurs.

XI. API et interfaces

A l'heure du lancement de la version finale de l'Alpha Game Builder, nous n'avons créé ni interfaces ni API.

En effet, l'objectif du projet est de proposer une solution directement utilisable via les outils situés sur la plateforme web. La création d'API et d'interfaces devient alors superflue.

XII. Liste des bugs connus

L'ensemble des bugs apparus durant le développement a été solutionné pour la version finale de l'Alpha Game Builder.

XIII. Coding style

L'ensemble du code produit sur l'Alpha Game Builder suit le coding style proposé par google et présenté en détail sur le lien suivant :

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

De plus, chaque fonction complexe est commentée afin d'en faciliter la lecture par un tiers.

En adoptant une norme stricte, existante et documentée, nous espérons pouvoir faciliter la prise en main et améliorer la lisibilité de notre solution.

XIV. Releases notes

Notre projet n'a été mis à la disposition du public qu'au moment de sa finalisation dans le cadre du module EIP.

Il n'y a donc pas de releases notes puisque la seule version présentée est la version 1.0.