

Dossier de conception

B1N-HER0

Boiveau Quentin

Cartier Pierre

Delorme Corentin

Demaret Kevin

Guenver Loic

Savarre Amaury

Wei Tianqi

Table des matières

1) Introduction.....	3
1.1) Présentation du document.....	3
1.2) Contexte.....	3
1.3) Portée du document.....	3
2) Environnement et outils de développement.....	4
2.1) Technologies utilisés pour l'implémentation.....	4
2.2) Environnements de développements.....	5
2.3) Autre.....	5
3) Architecture du projet.....	6
3.1) Conception générale du projet.....	6
3.2) Diagramme de classe.....	7
3.3) Schéma de navigation.....	9
3.4) Modèle relationnel en SQLite.....	10
3.5) Connexion client serveur.....	12
3.6) Synchronisation.....	12
4) Spécification de l'interface et du thème.....	12

1) Introduction

1.1) Présentation du document

Ce document est le dossier de conception du projet "B1N-HER0" qui consiste en la réalisation d'un jeu de type Takuzu (ou Binero) en Ruby et GTK+. Ce document a pour objectif de décrire précisément l'architecture du projet ainsi que les outils et technologies utilisés pour le réaliser.

Ce document fait référence au cahier des charges du projet qui définit les besoins du client, les fonctionnalités, ainsi que les contraintes imposées du projet.

1.2) Contexte

Ce document explicite l'architecture, les renseignements techniques ainsi que les contraintes à prendre en compte lors du développement de l'application.

Il va définir la conception du produit afin qu'il soit conforme au cahier des charges et cohérent par rapport aux attentes du client.

1.3) Portée du document

Ce document est destiné :

- aux professeurs évaluant notre projet (également clients dans le cadre de ce projet),
- à l'équipe de développement lors de la phase d'implémentation.

Il servira également pour évaluer le produit final.

2) Environnement et outils de développement

2.1) Technologies utilisés pour l'implémentation

Ruby en version 1.9.3 et supérieure

Ruby est un langage de programmation interprété open source et tout objet. Il a été créé en 1995 par Yukihiro Matsumoto. Ce projet utilisera ce langage pour la base du programme. Dans notre projet, nous travaillons avec la version 1.9.3 et supérieure qui permet d'implémenter GTK+ en version 3.

Librairie graphique GTK+ 3

The Gimp ToolKit (GTK) est un ensemble de librairie multiplate-forme gratuite écrit en langage C pour créer des interfaces graphiques utilisateur. GTK+ est une librairie simple d'utilisation et compatible avec de nombreux langages tels que Ruby, Java, C++, Perl, Python ...

Nous utilisons la version 3 qui permet de faire des transitions avec une syntaxe CSS, gestion de la transparence ...

SQLite

Nous nous appuyons également la librairie SQLite écrite en C qui propose un moteur de gestion de base de données relationnelle accessible par le langage SQL. L'avantage de SQLite est son intégration directement en local à la différence d'un SGBD qui requiert une mise en place assez complexe au niveau serveur et client.

Il nous servira à enregistrer nos données après extinction du programme et à les récupérer lors du démarrage tel que les profils, sauvegardes, classements, options, statistiques ...

2.2) Environnements de développements

Sublime Text

Ce logiciel est un éditeur de texte écrit en Python, multiplate-forme et gratuit en utilisation non commerciale. Il supporte de nombreux modules et langages pour s'adapter à l'utilisateur. Il comprend également de nombreux raccourcis clavier et fonctionnalités permettant de coder plus rapidement.

GitHub

C'est une plate-forme de gestion de versions basée sur Git permettant de travailler en collaboration dans le cadre d'un projet. Il permet de gérer l'historique des versions de fichiers, les modifications apportées et permet de partager plus facilement les fichiers au sein de l'équipe de développement.

Trello

Ce site gratuit permet d'organiser facilement les tâches en différentes parties tels que l'interface graphique, documentation, gestion des données ... appelés boards. Il permet ensuite de séparer les tâches en différentes colonnes (par exemple à faire, en cours, fini) et de les glisser-déposer entre ces colonnes via la souris. Ensuite on peut affecter chaque tâche à des membres de l'équipe et mettre une date au plus tard.

2.3) Autre

RubyDoc

Le Ruby Document Format est un langage de balisage qui se met dans le code source Ruby afin de générer une documentation propre sous la forme d'une page HTML.

3) Architecture du projet

3.1) Conception générale du projet

Nous avons choisi d'appliquer le patron MVC (Modèle-Vue-Contrôleur), afin d'avoir une structure solide et modulable pour le projet. Son fonctionnement suit le schéma suivant.

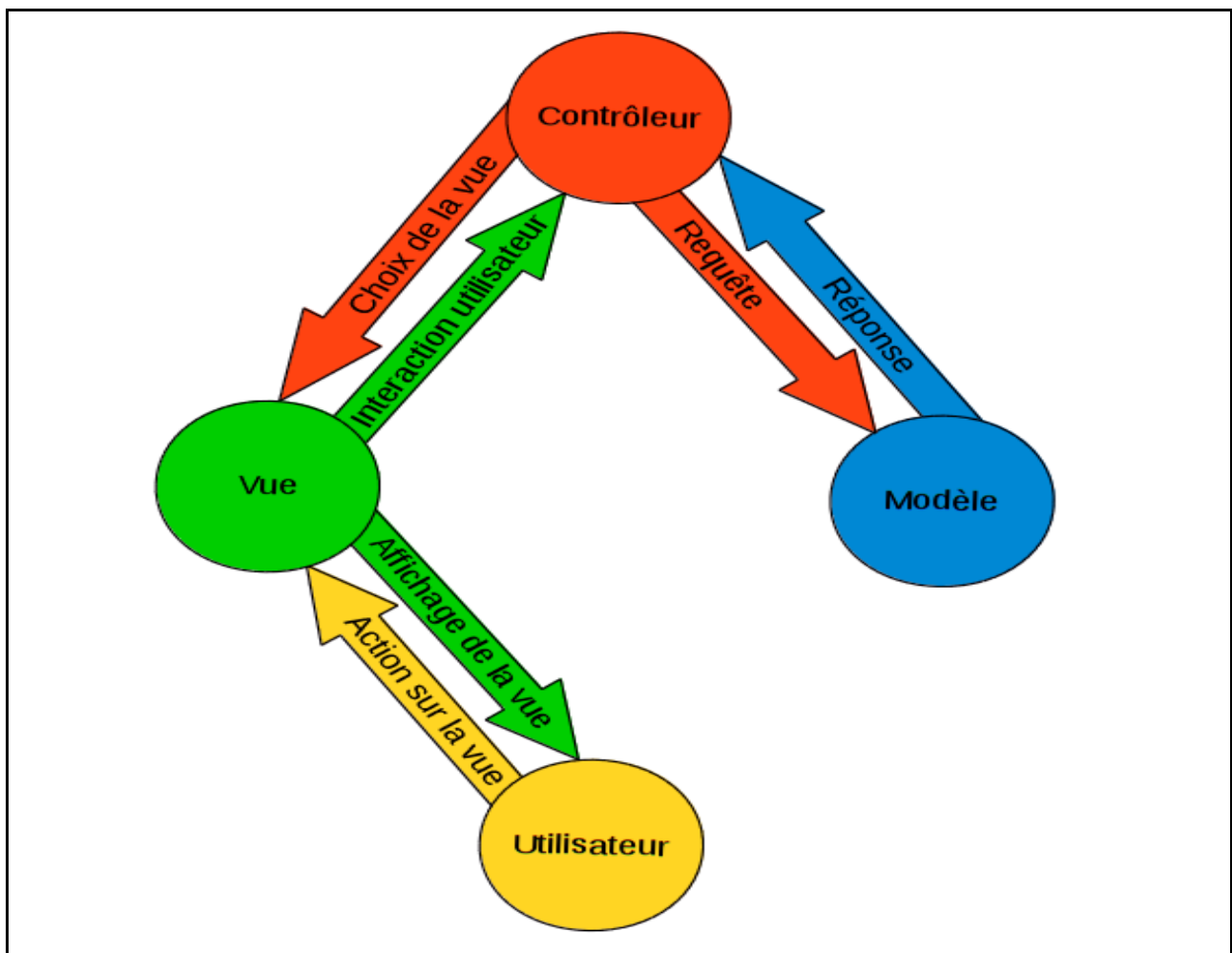


Illustration 1: Fonctionnement du modèle MVC

3.2) Diagramme de classe

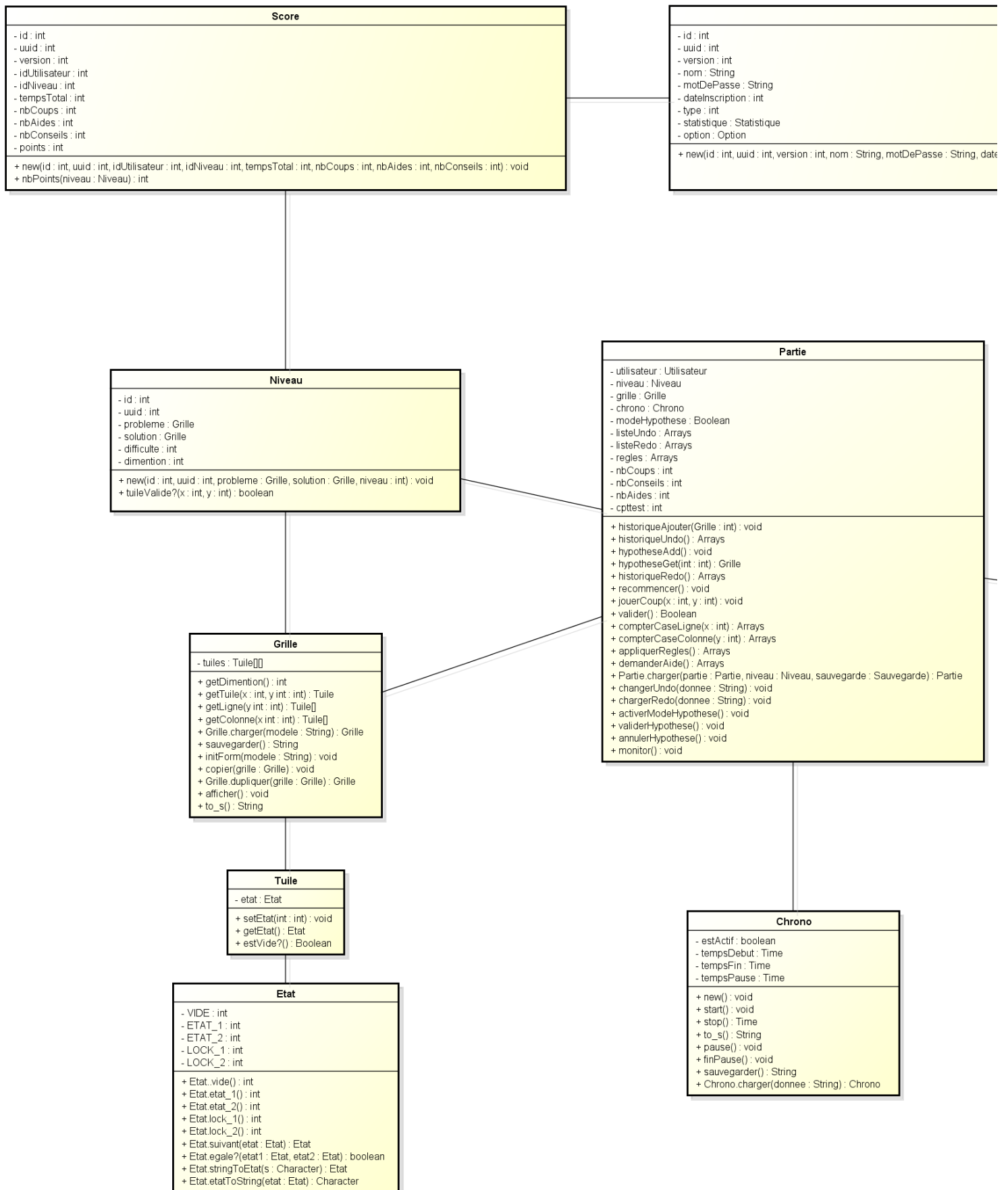




Illustration 2: Diagramme de classe

3.3) Schéma de navigation

Ce schéma représente, par des liens, les accès entre les différents menus et interfaces de l'application.

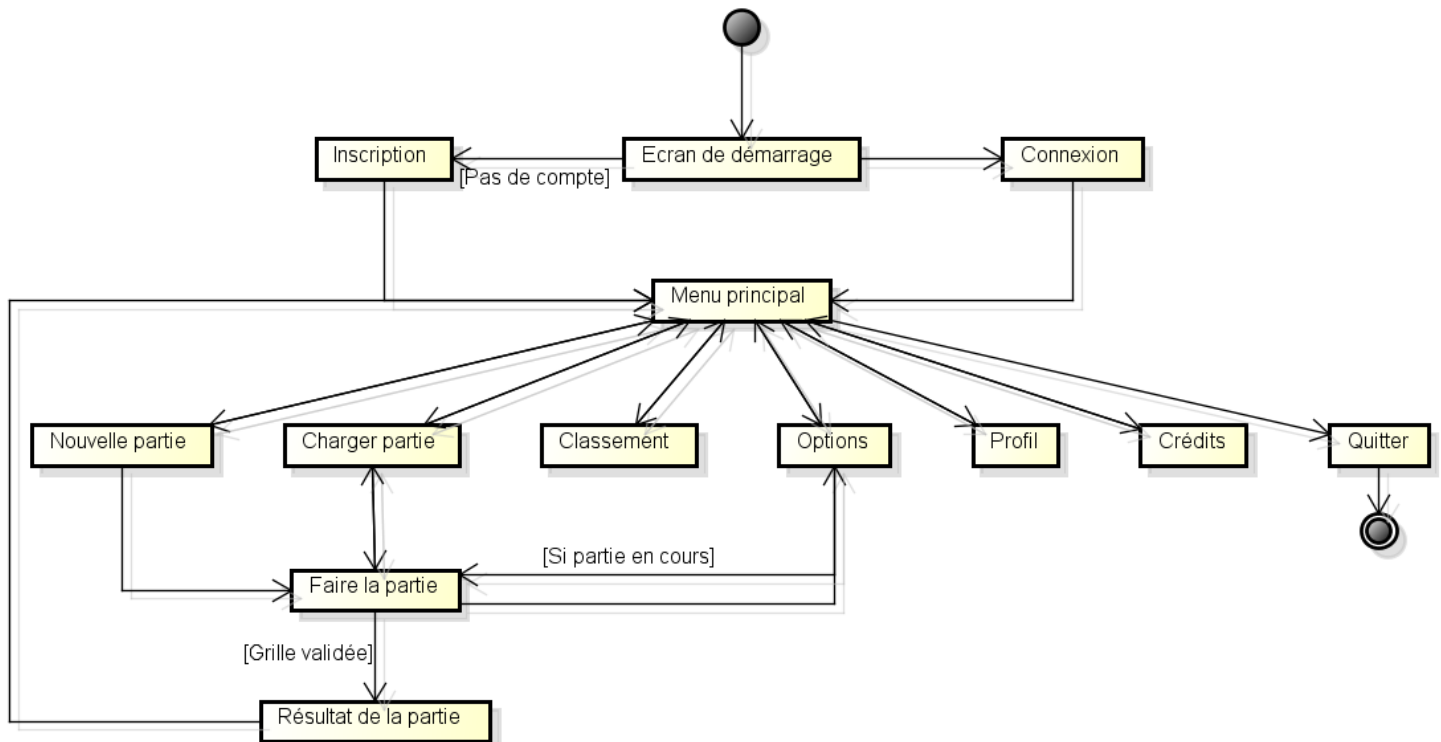


Illustration 3: Schéma de navigation

3.4) Modèle relationnel en SQLite

Modèle de la base client

La base de donnée locale, hébergée par le client, est utilisée lorsque le jeu est en mode hors ligne.

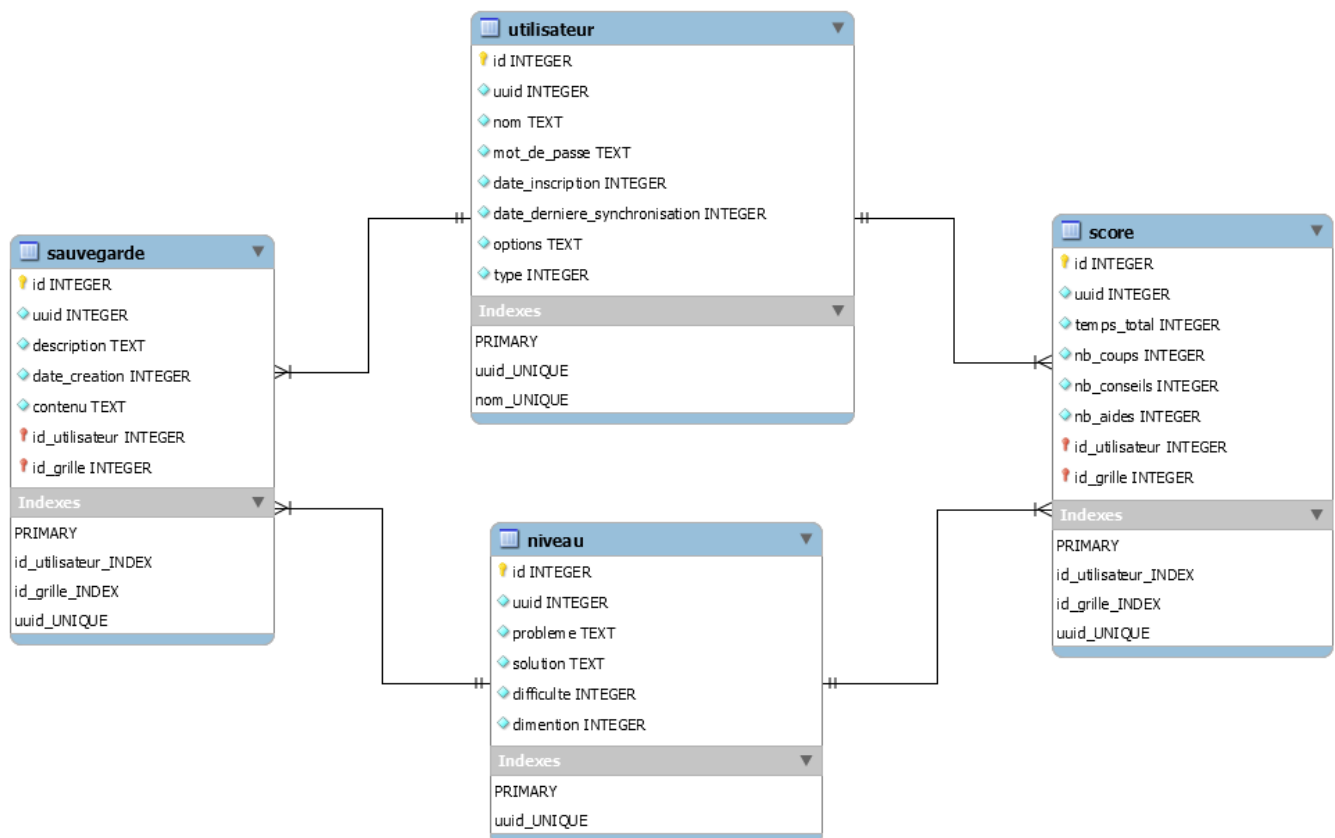


Illustration 4: MEA Client

Modèle de la base serveur

La base de donnée en ligne, hébergée sur un serveur, est utilisée lorsque le jeu est en mode en ligne.

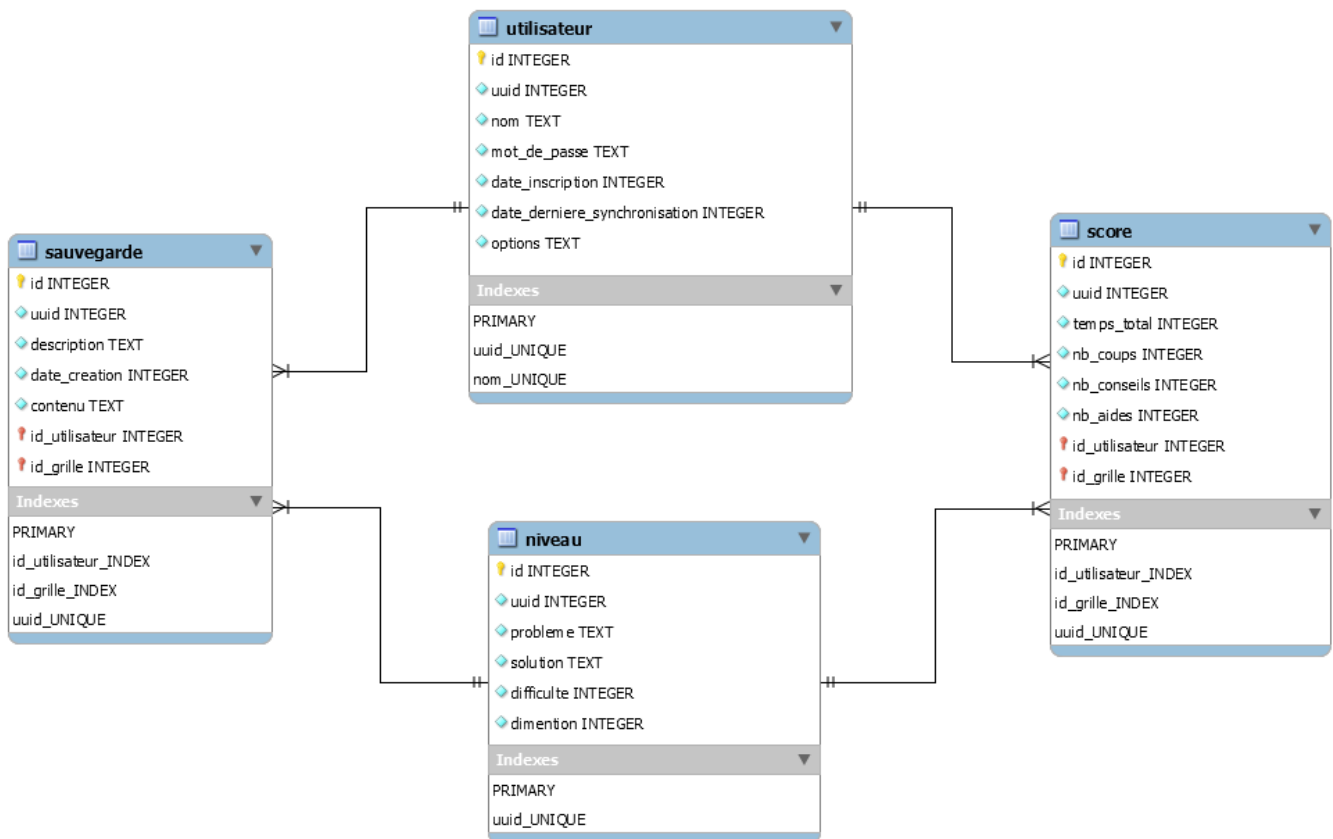


Illustration 5: MEA Serveur

3.5) Connexion client serveur

La communication client/serveur est réalisé en TCP/IP sur le port 10101. Les données échangées sont des objets ruby sérialisés. Les connexions ne sont pas persistantes : pour chaque demande, une connexion TCP est ouverte, puis fermé après réception de la réponse du serveur. La majorité des calculs sont effectuées sur les clients et le serveur fait principalement office de stockage réseau.

3.6) Synchronisation

La principale fonctionnalité permise par le serveur est la synchronisation de données. La synchronisation des données s'opère uniquement sur les données d'un utilisateur et non sur l'ensemble des utilisateur. En d'autres terme, il permet à un utilisateur d'avoir accès à ses données de n'importe quelle machine disposant de l'application, en lui donnant la possibilité d'envoyer ses données personnelles sur un serveur.

La synchronisation des données se décompose en quatre étapes qui sont :

- inscription
- authentification
- transfert de données à la connexion
- transfert de données à la déconnexion

Le transfert de données consiste à comparer les différences entre les données présentes sur le serveur et celles présentes sur le client. Les données étant persistées par le biais du SGBD sqlite, la comparaison des tuples se fait grâce à un attribut de version et un identifiant unique composé par le serveur.

4) Spécification de l'interface et du thème

Nous avons choisi une interface épurée, le plus ergonomique possible tout en conservant un maximum de fonctionnalités pour le rendre accessible à tous publics.

Pour le thème, il a été conçu autour du nom du jeu. Un super héros et des petits robots de couleurs ont été sélectionnés pour illustrer la partie.

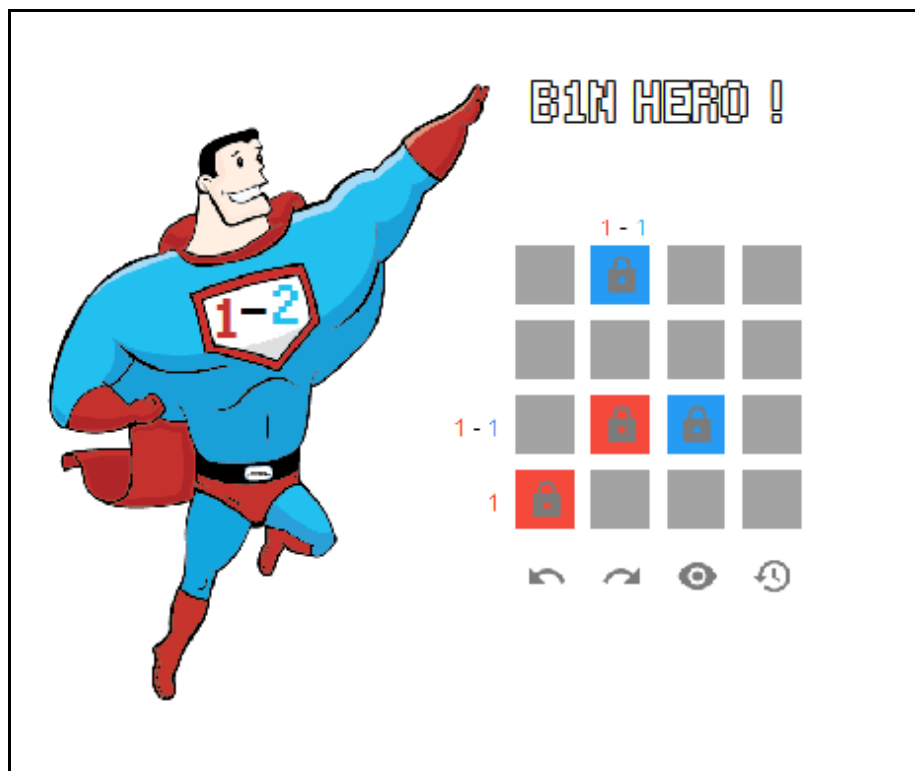


Illustration 6: Pré-rendu de l'interface en jeu



Illustration 7: Couleurs de base pour les tuiles