
Dossier de Conception : PicExplorer

LANVIN Elyan - MARCAIS Thomas

RAMOLET Arthur - GUENVER Loïc

AYDIN Emre - FOUCAULT Antoine

14 mai 2014

©2014 Groupe C - Tous droits réservés

Sommaire

1	Introduction	3
1.1	Présentation du document	3
1.2	Contexte	3
1.3	Portée du document	3
2	Présentation du projet	3
2.1	Contexte	3
2.2	Objectifs	4
2.3	Critères d'acceptabilité du produit	4
3	Environnement et outils de développement	4
3.1	Technologies utilisées pour l'implémentation	4
3.2	Environnements de développements	5
3.3	Autres	5
4	Architecture du projet	6
4.1	Conception générale du projet	6
4.2	Diagramme de classe	7
4.3	Schéma de navigation	8
4.4	Utilisation de SQLite et modèle relationnel	9
5	Spécifications des interfaces	10
5.1	Accueil	10
5.2	Editeur de grille	11
5.3	Jeu	12
5.4	Statistiques	13
5.5	Connexion	13

1 Introduction

1.1 Présentation du document

Ce document est le dossier de conception du projet “CrosSPI”. Il a pour objectif de décrire précisément l’architecture du projet ainsi que les outils et technologies utilisés pour sa mise en application.

Ce document fait référence au cahier des charges du projet qui définit les besoins du client, les fonctionnalités nécessaires et 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 a pour but de définir le produit et de détailler sa conception afin qu’il soit à la fois conforme à la conception décidée par le chef de projet et cohérente par rapport aux attentes du client.

1.3 Portée du document

Il est destiné :

- Aux professeurs du département informatique nous évaluant sur ce projet.
- A l’équipe du projet lors de la phase d’implémentation.

Il servira de base :

- A l’évaluation du produit final.
- Au développement des différents produits.

2 Présentation du projet

2.1 Contexte

Il se déroulera dans le cadre d’un projet tuteuré lors du semestre 6 de 3ème année de licence SPI.

Il est réalisé par un groupe de 6 étudiants dont un chef de projet en charge de la coordination des membres du groupe et du travail ainsi que d’un documentaliste chargé de mettre en forme les différents livrables écrits à rendre.

2.2 Objectifs

Ce projet doit aboutir à la création d'une application dédiée au jeu de Picross, sur PC et pour tout système d'exploitation, destinée à tout type d'utilisateur.

Cette application sera divisée en deux parties majeures :

- Edition de grilles de Picross avec importation/exportation de grille.
- Jeu de Picross avec sauvegarde/chargement de parties et aide à la résolution.

2.3 Critères d'acceptabilité du produit

L'application doit répondre aux critères suivants :

- Validation du produit via une série de tests réalisés par notre groupe.
- Respect des contraintes client (choix technologiques, fonctionnalités de l'application).

3 Environnement et outils de développement

3.1 Technologies utilisées pour l'implémentation

Ruby (version 1.9.3)

Ruby est un langage de programmation libre et tout-objet (chaque élément est lié à un objet) dont les fichiers, en .rb, sont interprétés. Il a été créé en 1995 par Yukihiro Matsumoto. Nous utiliserons ce langage pour l'implémentation de l'application, il sera la base du programme. Dans notre projet, nous travaillons avec sa version 1.9.3 afin d'assurer la compatibilité avec les ordinateurs de l'Université et Hop3x, ce qui est une des contraintes techniques imposées par le client.

Librairie graphique Gtk 2

The Gimp ToolKit (GTK) est un ensemble de bibliothèques logicielles contenant des fonctions qui permettent la réalisation d'interfaces graphiques pour différentes langages de programmation tels que Ruby, Java, C++ ou encore PHP. De même que pour Ruby, nous utilisons la version 2 de GTK, antérieure à la plus récente, afin d'assurer la compatibilité avec les postes de l'Université.

SQLite

SQLite est une bibliothèque écrite en C proposant un moteur de gestion de base de données relationnelle accessible par le langage SQL. A la différence d'un SGBD

“classique” (architecture client-serveur), SQLite est directement intégré, en local, au programme. Dans le cadre du projet, il nous servira à enregistrer nos données après extinction du programme et à les récupérer lors de son démarrage.

3.2 Environnements de développements

IDE Hop3x

Hop3x est un environnement de développement créé en Java et appartenant à l’Université du Maine. Il permet d’éditer et de compiler des programmes en C, Ruby ou encore Java et de sauvegarder ces projets en ligne.

GitHub

GitHub est une plate-forme de gestion de versions basée sur Git permettant de gérer les différentes versions des codes sources d’un logiciel au cours de son développement. Par exemple il permet d’ajouter des fonctionnalités en parallèle du projet jusqu’à sa version finale.

3.3 Autres

RubyDoc

Le Ruby Document Format est un langage de balisage que l’on peut mélanger avec le code source Ruby. Il permet de générer une documentation propre en fonction du code balisée.

4 Architecture du projet

4.1 Conception générale du projet

Afin de mieux structurer notre code et d'assurer une meilleure portabilité, nous avons choisi d'appliquer le design pattern Modèle Vue-Contrôleur. Son fonctionnement globale est détaillée dans le schéma ci-dessous :

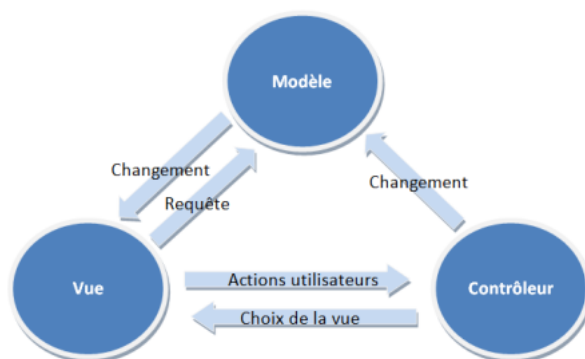


FIGURE 1 – Fonctionnement du modèle MVC

A chaque nouvelle interface, la classe Picross (voir Diagramme de classe ci-dessous) chargera le trio MVC concret associé. Par exemple pour l'interface "Editeur", nous aurons `VueEditeur`, `ModeleEditeur` et `ControleurEditeur`.

4.2 Diagramme de classe

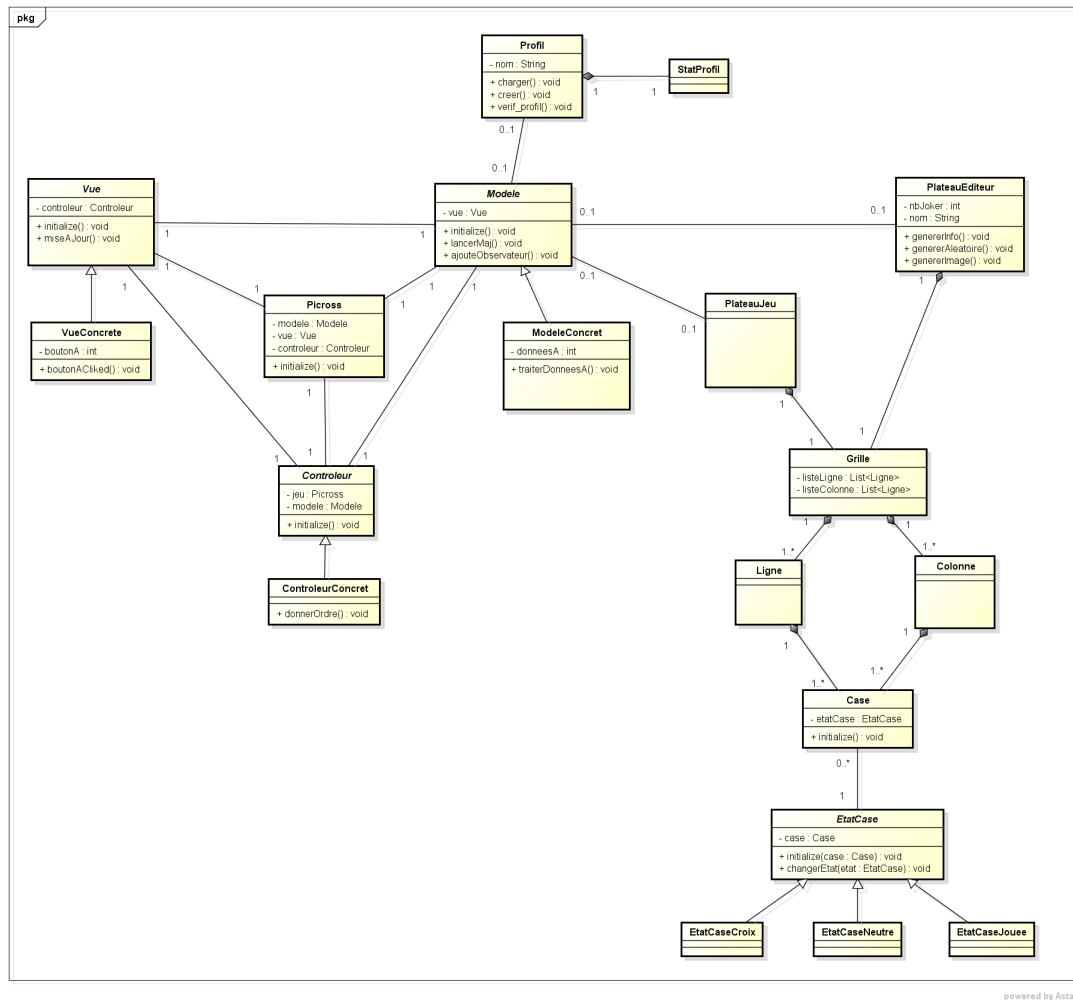


FIGURE 2 – Diagramme de classe de l'application

4.3 Schéma de navigation

Ci-dessous, le schéma de navigation de l'application, c'est-à-dire la représentation par un diagramme des liens d'accès entre les différentes interfaces de l'application :

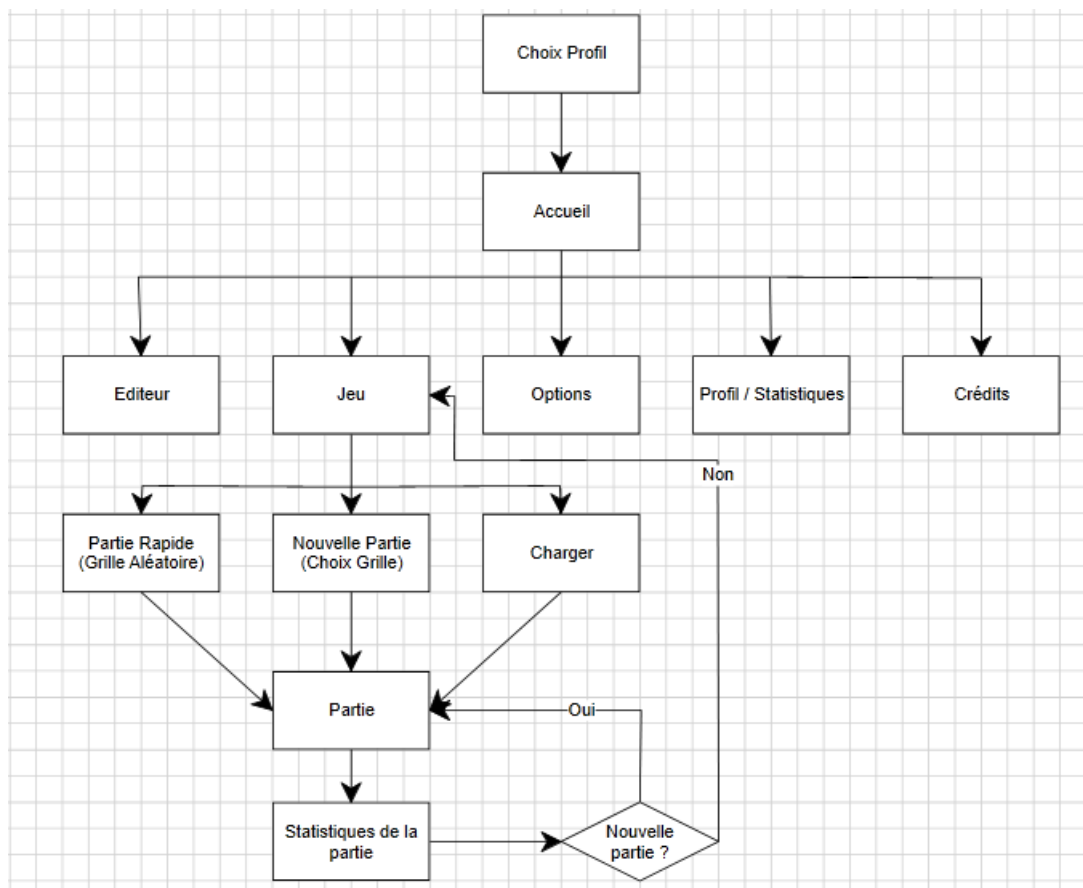


FIGURE 3 – Schéma de navigation de l'application

4.4 Utilisation de SQLite et modèle relationnel

Au lieu d'utiliser la sérialisation via YAML pour la persistance des données, nous avons préféré opter pour une base de données locale SQLite. Le modèle relationnel prévu sera le suivant :

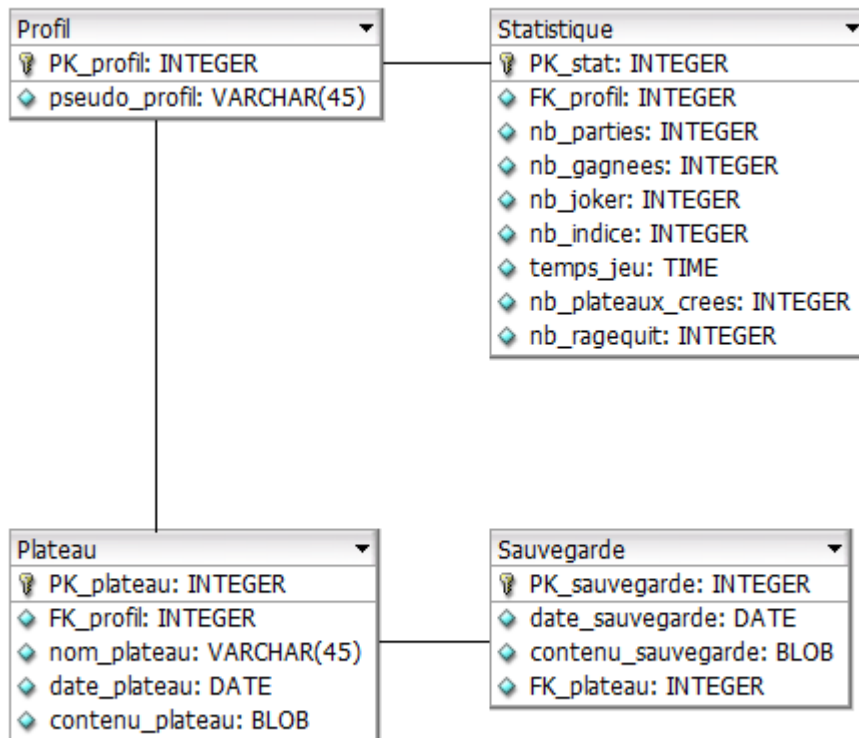


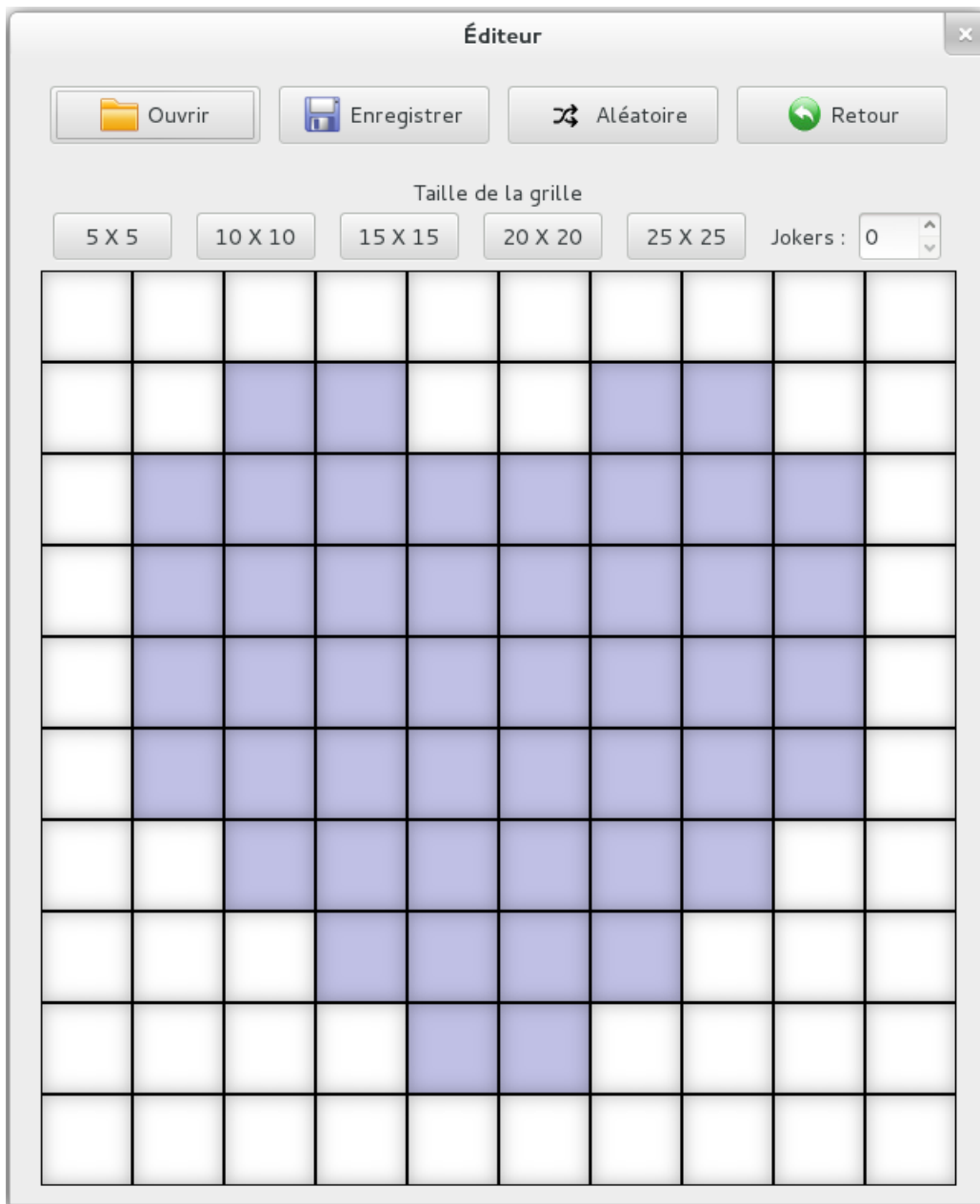
FIGURE 4 – Modèle Entité-Association de l'application

5 Spécifications des interfaces

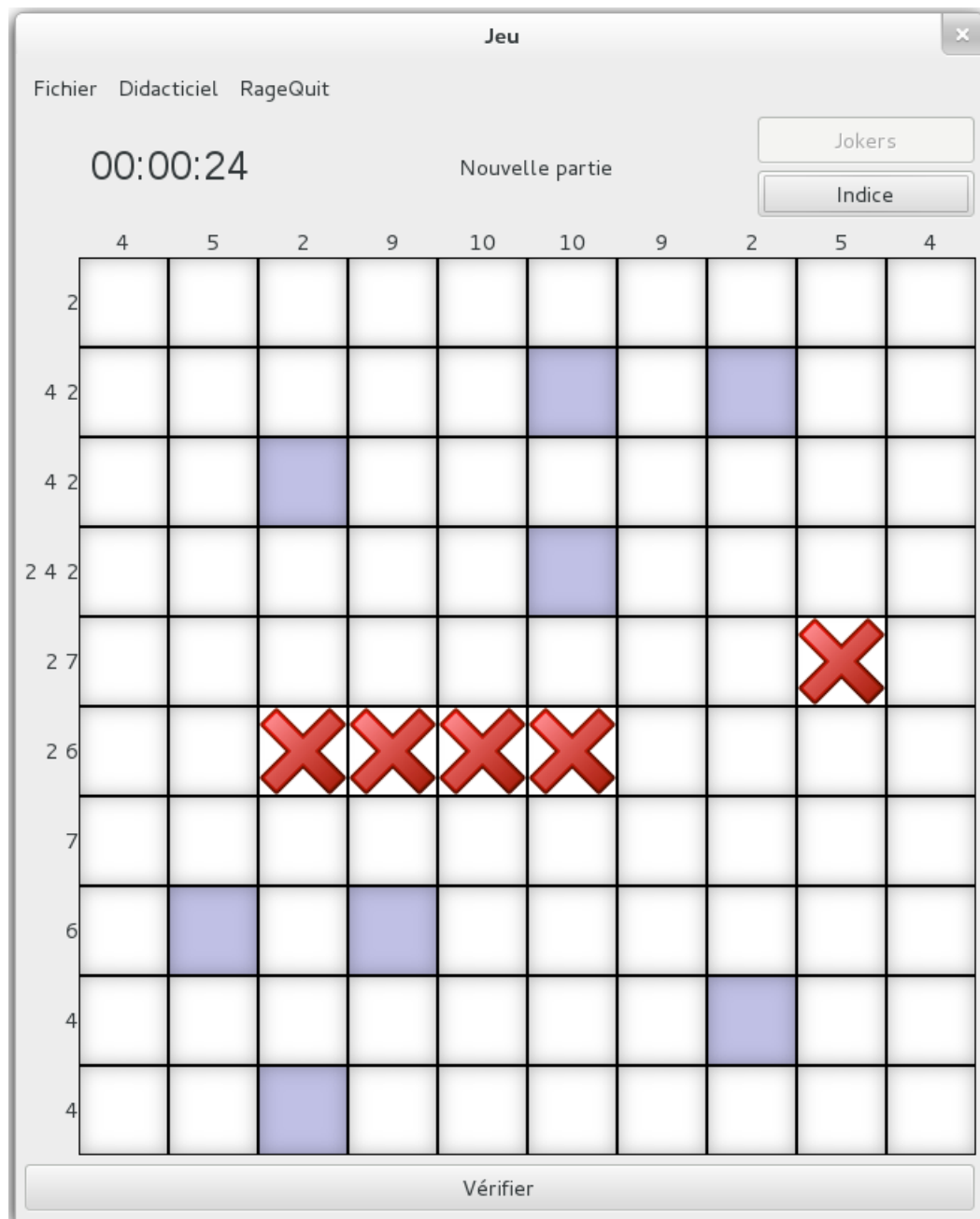
5.1 Accueil



5.2 Editeur de grille



5.3 Jeu



5.4 Statistiques

Profil/Classement

Statistiques et classement de Route

Joueur	Taux de réussite	Temps joué	Jokers utilisés	Indices utilisés	Nombre de clics	Nombre de ragequi
Profil Teste	0%	00h00m00s	2	2	39	0
Profil Testg	0%	00h00m00s	0	0	0	0
Route	0%	00h00m00s	0	0	34	0

Retour à l'accueil Renommer mon profil Réinitialiser mes statistiques

5.5 Connexion

Connexion

Supprimer Route ▼ Connecter

Ajouter