

# **DOSSIER DE CONCEPTION : PROJET JEU**

Mercredi 5 mars 2014

**Groupe A**

L3 info SPI

# Sommaire

<b>1</b>	<b>Présentation général</b>	<b>1</b>
1.1	Objectif du document . . . . .	1
1.2	Objectif du projet . . . . .	1
1.3	Documents de références . . . . .	1
<b>2</b>	<b>Modélisation du projet</b>	<b>2</b>
2.1	Les cas d'utilisations . . . . .	2
2.2	Contraintes de conception . . . . .	2
2.3	Le scénario du système . . . . .	2
2.4	Schéma de navigation . . . . .	2
<b>3</b>	<b>Modélisation de la Sauvegarde des données</b>	<b>3</b>
3.1	Dictionnaire de données . . . . .	3
3.2	Description des données sauvegardées et de leur stockage . . . . .	3
3.3	Mécanismes de Sauvegarde des Données . . . . .	3
<b>4</b>	<b>Interface Homme-Machine</b>	<b>4</b>
<b>5</b>	<b>Outils, normes et standards de programmation</b>	<b>5</b>
5.1	Langages utilisées . . . . .	5
5.2	Environnement et outils de développement . . . . .	5
5.3	Standards de programmation . . . . .	5

# **1 Présentation général**

## **1.1 Objectif du document**

Ce document est rédigé dans le cadre du projet de troisième année de licence SPI option informatique de l'Université du Maine. Le Dossier de Conception vient en complément du Cahier des Charges. Il a pour objectif de monter la modélisation du système à partir des besoins et contraintes exprimés dans le Cahier des Charges. Il informe également sur les outils et les normes utilisés pour la conception du produit.

## **1.2 Objectif du projet**

L'objectif de ce projet est de concevoir et de développer une application de Picross, ainsi que de rédiger les documents inhérent à la gestion et la finalisation d'un projet informatique. Les utilisateurs doivent pouvoir, bien évidemment, jouer, mais il doivent également pouvoir créer des grilles, modifier la taille des grilles, et charger les parties préalablement sauvegardées. Il doit également être possible de jouer sous son propre profil afin d'avoir accès aux options les plus poussées de l'application.

## **1.3 Documents de références**

Afin de réaliser le présent Dossier de Conception nous nous sommes appuyés sur le Cahier des Charges ainsi que sur les exemples que nous avons récupérés grâce à un ancien élève de DUT informatique.

## **2 Modélisation du projet**

### **2.1 Les cas d'utilisations**

Partie de Kévin

### **2.2 Contraintes de conception**

Partie de Kévin

### **2.3 Le scénario du système**

Partie de Kévin

### **2.4 Schéma de navigation**

Partie de Kévin

### 3 Modélisation de la Sauvegarde des données

#### 3.1 Dictionnaire de données

Nom de l'entité	Type	Code	Commentaires
Grille	Class Ruby Grille		une collection de case, jointe à un nom et des paramètres
Profil	Chaîne de caractère		le profil joueur, constitué uniquement du nom du profil
Scores	Tableau		un tableau à trois colonnes, comprenant le nom de la grille
Parties Sauvegardées	Class Ruby Partie		Partie en cours, sauvegardée avec tous ses paramètres

#### 3.2 Description des données sauvegardées et de leur stockage

L'ensemble des données sauvegardées, présentées brièvement ci-dessus, est ici décrite plus en détail.

**Grille** : les grilles sont les supports du jeu( cf Cahier des Charges 3.1.1). Pour permettre à plusieurs utilisateurs de jouer sur une même grille, il est nécessaire de sauvegarder ces objets. Les grilles sont constituées d'une collection de case, d'un nom, attribué à la création et de divers paramètres, stockés sous formes d'entiers. Elles sont stockées dans un même fichier, les unes après les autres, à leur création, qu'elle soit manuelle ou aléatoire. Ce fichier comporte trois grilles par défaut.

**Profil** : Les profils sont les noms que choisissent les utilisateurs(cf Cahier des Charges 3.1.2, qui les définissent dans le jeu, et qui les lient aux grilles créées, aux parties jouées et aux scores. Un profil est constitué d'une simple chaîne de caractères, choisie par l'utilisateur à la création du profil. Cette chaîne est sérialisée à la création, dans un tableau contenant l'ensemble des profils créés, le tout stocké sur un même fichier spécifique aux profils. Ce fichier comporte un profil "par défaut".

**Scores** : les scores sont des entiers, dépendant du temps réalisé pour finir la Grille. La donnée à sauvegardée "Scores" est un tableau à trois colonnes, comprenant le score lui même, le nom de la grille sur lequel il a été réalisé, et le nom du profil l'ayant réalisé. Ce tableau est stocké sur un fichier propre.

**Parties Sauvegardées** : elles correspondent à la sérialisation d'une instance de la classe Partie, contenant un certain nombre de paramètres et d'informations, comme la grille utilisée, l'utilisateur, le temps écoulé, etc. Ces paramètres sont conservés sur un fichier lors de la sérialisation. L'utilisateur peut ainsi quitter une partie, fermer le programme, et reprendre cette partie plus tard. Chaque Partie Sauvegardée est conservée sur un fichier propre.

#### 3.3 Mécanismes de Sauvegarde des Données

## 4 Interface Homme-Machine

Partie d'Anice

## 5 Outils, normes et standards de programmation

### 5.1 Langues utilisées

Ci dessous nous référençons la liste des différents langages utilisés dans le projet :

- Ruby : Langage de programmation orienté objet, utilisé en version 2.0.0 sur nos machines personnelles, avec une compatibilité sur les versions antérieures,

### 5.2 Environnement et outils de développement

Le développement sera effectué sur des ordinateurs suffisamment puissants auxquelles nous avons accès à l'Université du Maine, ainsi que sur nos machines personnelles.

- Version utilisé de Ruby : 2.0.0
- GTK : Bibliothèque graphique (The **GIMP Toolkit**) utilisé avec Ruby.

### 5.3 Standards de programmation

Durant le développement, nous respecterons les conventions de codages ci contre :

- Les commentaires : ils doivent être rédigés en français et permettent d'expliquer de façon claire le code,
- Le codage : à voir les standards de programmation
- La documentation : Ruby doc

La conclusion à pofiner