

# **CAHIER DES CHARGES : PROJET JEU**

Mercredi 5 février 2014

**Groupe A**

L3 info SPI

## Sommaire

# 1 Présentation du projet

## 1.1 Contexte

Lors du second semestre de troisième année de licence SPI, il est demandé de réaliser un projet. Ce projet à thème imposé n'est autre que le développement d'un logiciel complexe demandant un travail en équipe.

Notre équipe de cinq étudiants aura donc du 24 janvier au 16 Mai 2014 pour mettre en œuvre les préceptes de Génie Logiciel vus au premier semestre afin de présenter un logiciel potentiellement commercialisable au client, ici représenté par les enseignants.

## 1.2 Objectifs

L'objectif de ce projet est la conception d'une application permettant la création et l'aide à la résolution de puzzles de type picross (henjie).

Le joueur pourra commencer des puzzles de différentes tailles ; le but étant de terminer le plus vite possible la partie.

## 1.3 Utilisateur cible

Ce logiciel n'est destiné qu'à un seul joueur à la fois. Cependant, il permet à de nombreux joueurs d'apposer leur score à la table des scores.

## 1.4 État de l'art

Les jeux déjà disponibles permettent différentes fonctionnalités :

- La création aléatoire et édition d'une grille de picross de diverses tailles,
- La création d'une grille à partir d'une image,
- La résolution du puzzle avec ou sans aide à la résolution,
- La sauvegarde et le chargement d'une partie,
- L'enregistrement et consultation des scores,
- Le partage de ses grilles déjà jouées,
- Cocher une case en cliquant dessus une fois, ce qui changera sa couleur,
- Marquer une case comme vide en cliquant dessus deux fois et en apposant une croix.

# 2 Contraintes initiales

## 2.1 Contraintes de jeu

- Le jeu est un picross, le joueur doit noircir les cases en fonction des informations données par les chiffres en tête de ligne et de colonne,

- Le jeu doit proposer plusieurs tailles de grille (5x5 10x10 15x15 25x25),
- L'application doit proposer un éditeur de grille.

## 2.2 Contraintes de conception

- Le langage de programmation est Ruby/GTK,
- Les paramètres de l'application doivent être externalisés afin de pouvoir être modifiés par le joueur (taille de la grille),
- Le logiciel doit être développé en Programmation Orientée Objet (POO),
- Le jeu doit supporter plusieurs langues,
- Le joueur doit pouvoir sauvegarder une partie pour la reprendre ultérieurement.

## 2.3 Contraintes temporelles

- Le client impose une réunion hebdomadaire entre les différents acteurs de notre groupe,
- Le projet final doit être présenté au client le 16 mai 2014.

## 2.4 Contraintes matérielles

- Le présent Cahier des Charges doit être remis au client le 16 mai 2014,
- Au terme de chaque réunion un compte rendu doit être remis au client.

# 3 Spécification du projet

## 3.1 Règles détaillées

### 3.1.1 Éléments constituant le jeu

**Grille** : La grille est constituée d'un carré dont le coté est un multiple de 5 cases. Il y a ainsi cinq tailles de grilles différentes, 5x5, 10x10, 15x15, 20x20 et 25x25.

**Case** : Les cases peuvent être noircies, en cliquant dessus à l'aide de la souris. Il est possible de décocher la case en cliquant dessus à nouveau. Il est également possible de cliquer sur une case à l'aide de la souris pour indiquer que la case restera vide, qu'elle ne sera pas noircie.

**Indications extérieures** : Les indications extérieures sont des suites de chiffres sur les bords gauche et supérieur, indiquant le nombre de case à noircir, respectivement dans les lignes et dans les colonnes.

**Chronomètre** : le chronomètre indique le temps qui s'est passé depuis que le joueur a noirci la première case.

### 3.1.2 Principe de fonctionnement du jeu

**But du jeu** Le but du jeu est de noircir toutes les cases devant être noircies dans le plus court laps de temps possible.

**Début d'une partie** Au début d'une partie, la grille est vierge, et le chronomètre est à 00:00.

**Déroulement du jeu.** Le jeu commence lorsque le joueur noircie une case en cliquant dessus, ce qui enclenche le chronomètre.

**Le menu en cours de partie** En cours de partie, le joueur peut accéder à un menu spécial où se trouvent :

**Sauvegarde de la partie :** L'utilisateur peut sauvegarder sa partie en cours. Le joueur peut ensuite continuer à jouer ou quitter le jeu ;

**Quitter la partie :** Renvoie le joueur au menu principal. Il devra avoir pris le soin de sauvegarder ou non sa partie.

**L'interface joueur** Le joueur noircit des cases ou les rend de nouveau blanches en cliquant dessus à l'aide de la souris. Il peut également cliquer sur un bouton **Aide** ou sur le bouton **Menu**.

### 3.2 Besoins fonctionnels

FONCTIONNALITE	DESCRIPTION	M/m*
Gestion du jeu	Lancer une nouvelle partie	M
	Sauvegarder une partie (à tout instant)	M
	Charger une partie sauvegardée	M
	Supprimer une partie sauvegardée	M
	Quitter le jeu	M
	Choisir la taille de la grille (Facile, Moyen, Difficile)	M
	Choisir la langue	m
	Afficher les statistiques/classement (nom du joueur, nombre de partie jouées, victoire, temps, score)	M
	Afficher une aide	M
	Activer / Désactiver le son	m
Gestion du joueur	Choisir un pseudo pour le joueur	m
	Gérer ses parties	M
Interface du jeu	Afficher les informations de la grille	m
	Afficher le score	M
	Afficher le menu de jeu	M

\* M = Majeure / m = mineure

### 3.3 Interface avec l'utilisateur

#### 3.3.1 Interface Homme-Machine

Le jeu est une application pour ordinateur qui se lance dans une fenêtre. Le joueur regarde l'écran et interagit à la fois avec son clavier et sa souris.

#### 3.3.2 Les menus

**Le menu principal** Après le lancement de l'application, le joueur se trouve dans le menu principal où figure :

- **Lancement d'une nouvelle partie ;**
- **Charger une partie :** Ce menu permet de retrouver une partie à l'endroit où on l'avait laissé. Il permet aussi la suppression d'une sauvegarde ;
- **Le "Hall of Fame" :** On y trouve les meilleurs scores de tous les joueurs ayant utilisé cette application. Les 10 meilleurs scores et leurs statistiques associées sont visibles dans trois sous-menus différents pour chaque difficulté (novice, moyen, expert). Il y a trois statistiques représentant une partie : le nombre d'ennemis tués, la distance parcourue, (une case = un mètre), l'or total accumulé et le temps de jeu. Le score associé à une partie est synthétisé à partir de l'ensemble de ces statistiques ;
- **Langue :** Permet de changer la langue parmi celles disponibles ;
- **Aide :** Donne accès à une base d'aide permettant à l'utilisateur de prendre facilement le jeu en main ;
- **Quitter le jeu.**

### **3.3.3 Interface Graphique**

Voici un schéma général de l'interface graphique dont disposera le jeu :

## 4 Déroutement du projet

### 4.1 Livrables

Les livrables prévus sont :

- Le présent cahier des charges validé par le client ;
- Dossier d'analyse ;
- Manuel utilisateur ;
- Le jeu fonctionnel (livraison le jeudi 16 mai 2014).

### 4.2 Planning

Afin de mener à bien ce projet, il est mis à disposition des étudiants 16 séances de 3h (en plus du temps libre) pour permettre aux membres de l'équipe de se retrouver et au chef d'équipe de distribuer les tâches en respectant au mieux le planning mis en place au début du projet.

### 4.3 Équipe

Les membres de l'équipe sont Rémi TREMBLAIN (chef d'équipe), Erwan MARCAHND, Colas PICARD, Kévin CROILLERE et Anice KHOMANY.

### 4.4 Outils de développement

Voici les différents outils utilisés pour le projet :

- Le langage de programmation Ruby ;
- GTK pour Ruby pour le développement des interfaces graphiques ;
- YAML : outil qui permet la sérialisation de données, et qui sera donc utilisé pour la sauvegarde de données, telles que l'inventaire, la carte, la partie en cours.

## 5 Glossaire

**GTK** : Gimp ToolKit

**POO** : Programmation Orientée Objet

**YAML** : YAML ain't markup language (acronyme récursif)

---

Note : Les chiffres et pourcentages sont fournis à titre indicatif. Il pourront être modifiés pour une meilleure jouabilité.