

CAHIER DES CHARGES : PROJET JEU

Mercredi 5 février 2014

Groupe A

L3 info SPI

Sommaire

1	Présentation du projet	1
1.1	Contexte	1
1.2	Objectifs	1
1.3	Utilisateur cible	1
1.4	État de l'art	1
2	Contraintes initiales	1
2.1	Contraintes de jeu	1
2.2	Contraintes de conception	2
2.3	Contraintes temporelles	2
2.4	Contraintes matérielles	2
3	Spécification du projet	2
3.1	Règles détaillées	2
3.1.1	Éléments constituant le jeu	2
3.1.2	Principe de fonctionnement du jeu	4
3.2	Besoins fonctionnels	7
3.3	Interface avec l'utilisateur	7
3.3.1	Interface Homme-Machine	7
3.3.2	Les menus	7
3.3.3	Interface Graphique	9
4	Déroulement du projet	10
4.1	Livrables	10
4.2	Planning	10
4.3	Équipe	10
4.4	Outils de développement	10
5	Glossaire	10

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

Le jeu permet différentes fonctionnalités :

- La création et édition d'une grille de picross,
- 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.

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

L'énergie Le joueur ainsi que les PNJ ennemis disposent d'une énergie. La difficulté du jeu est de gérer cette énergie au mieux. Pour le joueur, celle ci diminue lors des déplacements et lors des combats (les ennemis ne perdent pas d'énergie lors des déplacements). Si elle est inférieure ou égale à zéro, le joueur meurt.

Le repos Le joueur débute avec un nombre de repos adapté en fonction de la difficulté, il en gagne un nouveau tout les cinq niveaux. Un repos lui permet de regagner toute son énergie, il consomme dix tours. Si un ennemi déclenche un combat pendant le repos, alors seul une partie de l'énergie est récupérée.

Les PNJ ennemis et les combats Les combats ne requièrent pas d'interaction de la part du joueur. Ils se font toujours avec un ennemi à la fois et tiennent compte de l'énergie du joueur ainsi que de son équipement et de celle de l'ennemi. Lors d'un combat, l'énergie dont dispose l'ennemi est soustraite à celle du joueur. Le joueur peut donc remporter le combat s'il dispose de plus d'énergie que l'ennemi. Dans le cas inverse il meurt.

Certains ennemis se déplacent aléatoirement sur la carte tandis que d'autres ont pour but de pister le joueur s'il est à moins de 20 cases de lui. Ils connaissent donc la position du joueur et se

rapprochent de lui tour après tour. Lorsqu'un joueur se trouve sur la même case qu'un ennemi, le combat est obligatoire. Un ennemi a sur lui des objets et/ou de l'or. Plus l'ennemi est puissant et plus ce que l'on trouve sur lui sera intéressant mais il n'est pas possible pour le joueur de connaître ces objets avant l'issue du combat.

Voici la liste de base des PNJ ennemis :

- La chauve souris : énergie de base 30 ;
- Le loup : énergie de base 50 ;
- Le brigand : énergie de base 90 (pisteur) ;
- Le soldat : énergie de base 150 (pisteur) ;
- L'ogre : énergie de base 210.

Les ennemis sont générés avec un interval de tours régulier dépendant de la difficulté.

Les PNJ d'aide Les PNJ d'aide sont générés au début de la partie, immobiles et non attaquables. Lorsque le personnage rencontre un PNJ d'aide, celui-ci lui demande s'il veut faire une transaction (une seule transaction par tour). Si le personnage accepte, alors une liste de ce qui est proposé apparaît avec les prix correspondants. Le personnage sélectionne donc ce qu'il souhaite dans la limite de son capital d'or disponible et de son inventaire dans le cas d'un achat d'objet.

- **Les guérisseurs** : Les guérisseurs proposent au joueur qui vient les trouver un regain d'énergie en échange d'or. La quantité d'énergie proposée est un pourcentage de l'énergie maximale du joueur (+25% = 30 pièces d'or, +50% = 50 pièces d'or, +75% = 70 pièces d'or). Le joueur ne sera pas remboursé s'il paye pour plus d'énergie qu'il n'en a besoin.
- **Les commerçants** : Les commerçants permettent au joueur d'acheter ou de vendre des objets contre de l'or. Le joueur ne peut revendre que des objets qui se trouvent dans son inventaire : les objets avec lesquels il est équipé ne sont pas vendables. Lorsque le personnage rencontre un commerçant, il lui demande s'il veut acheter des objets ou non.

Les objets du jeu et l'inventaire Des objets et de l'or sont disséminés sur la carte. Les ennemis vaincus donnent également de l'or et/ou des objets. Enfin les commerçants peuvent vendre des objets contre de l'or et inversement.

Voici les différentes catégories d'objets du jeu :

- **Les objets défensifs/offensifs** : Parmi ces objets figurent des boucliers et des épées. Ces objets se caractérisent par le pourcentage de protection/attaque qu'ils procurent et permettent de réduire la perte d'énergie liée à un ennemi lors d'un combat. Par exemple, si le joueur s'est revêtu d'une armure de pourcentage 20% avant de combattre un ennemi d'énergie 200, il ne perdra que 80% (= 160) de l'énergie de l'ennemi. Ces objets ne sont valables que pour un seul combat.

- **Les bottes** : Les bottes permettent de réduire le coût en énergie d'un déplacement d'une certaine quantité (variable selon les bottes) et pour un nombre fixé de déplacements (également variable). Il est impossible de les retirer une fois que le joueur s'en est équipé.
- **La nourriture** : Ces objets permettent de regagner de l'énergie. Ils sont consommés en une seule fois. Selon le type de nourriture consommée, le joueur gagnera plus ou moins d'énergie :
 - Le bout de pain augmentera de 70 l'énergie ;
 - Le brocoli vapeur augmentera de 100 l'énergie ;
 - Le steak de dinde augmentera de 200 l'énergie ;
 - Le pavé d'émeu augmentera de 500 l'énergie.
 - Le poulet rôti augmentera de 800 l'énergie.

Le joueur dispose de trois emplacements d'équipements : pieds pour les bottes, main gauche pour le bouclier et main droite pour l'épée.

Les différentes caractéristiques d'un objet (regain d'énergie, pourcentage de protection, nombre d'utilisations pour les bottes) lui confèrent une valeur. Plus l'objet a de valeur et plus il sera cher, difficile à trouver sur la carte et présent dans l'inventaire d'ennemis puissants.

Le joueur dispose d'un inventaire limité à quinze objets, pour stocker des objets qu'il ne veut pas utiliser immédiatement. Les objets d'un ennemi vaincu se retrouvent directement dans l'inventaire du joueur sauf si ce dernier est plein, auquel cas le joueur devra choisir quels objets du monstre ou de son inventaire il souhaite jeter. Un objet jeté de l'inventaire est détruit. Si un joueur passe sur une case où se trouve un objet, il peut décider de le ramasser ou non (et également de jeter des objets de son inventaire si besoin).

L'or ne prend pas de place dans l'inventaire. Par conséquent, l'or d'un ennemi vaincu ou ramassé s'ajoutera automatiquement à la cagnotte du joueur.

La carte du jeu Le monde dans lequel le joueur évolue est fini mais torique, évitant ainsi de limiter les déplacements aux bords de la carte. Le jeu est en deux dimensions et vu du dessus. La carte est divisée en cases carrées qui sont d'un certain type de terrains. Selon le type de terrains, un déplacement consommera plus ou moins d'énergie. Certains types de terrains sont infranchissables. Voici la liste des types de terrains :

- Eau : infranchissable ;
- Plaine : coûte 10 d'énergie pour s'y déplacer ;
- Forêt : coûte 20 d'énergie pour s'y déplacer ;
- Désert : coûte 30 d'énergie pour s'y déplacer ;
- Montagne : coûte 50 d'énergie pour s'y déplacer.

3.1.2 Principe de fonctionnement du jeu

But du jeu Le but général pour le joueur est de se déplacer le plus longtemps possible dans cet environnement. Le jeu se termine quand le joueur meurt, c'est à dire quand il n'a plus d'énergie. Le but est d'obtenir le meilleur score possible.

Début d'une partie Au début d'une nouvelle partie, le personnage se trouve au centre de la carte. Il dispose d'une quantité d'énergie liée à la difficulté. Il n'a aucun objet et aucune pièce d'or. Le logiciel fournit au joueur une interface lui affichant les seules actions possibles sous forme de bouton à cliquer

Le jeu au tour par tour Le déroulement du jeu se fait tour par tour. Une action pour un tour correspond à un déplacement (dans les quatre directions nord, sud, est et ouest sauf cas de cases inaccessibles) plus une interaction dans la case où se trouve le joueur à la fin de ce déplacement : il n'est possible d'interagir qu'avec les éléments présents sur la même case que le personnage. Lors d'un tour, le joueur peut décider de rester sur place (ce qui compte pour un déplacement nul). Il pourra ensuite agir sur la case sur laquelle il est resté avant la fin du tour. Une fois le tour du joueur fini, le monde évolue (mouvement des ennemis, génération de nouveaux ennemis et/ou objets sur la carte).

Se reposer (utilisation d'un repos) permet au joueur de récupérer 10% de la totalité de son énergie à chaque tour pendant 10 tours sauf si ce dernier est attaqué durant son sommeil, auquel cas le repos s'arrête et un combat s'engage.

Lors d'une interaction avec un PNJ d'aide, chaque transaction compte pour un tour : pour acheter trois objets, il faudra rester sur place trois fois de suite. Le joueur ne se déplaçant pas, il ne consomme (ni ne gagne) aucune énergie.

Dans le cas où plusieurs éléments sont présents sur une même case, le joueur peut décider d'interagir avec, dans l'ordre qu'il souhaite, sachant que chaque action équivaut à un tour en restant sur place (ramassage de plusieurs objets, interaction avec un/des PNJ d'aide). Si des ennemis se trouvent parmi ces éléments, la règle est différente : les ennemis doivent être traités en priorité et en un seul tour, ce qui signifie que l'énergie soustraite à celle du joueur correspond à la somme des énergies de tous les ennemis présents sur la même case et qu'il est impossible au joueur d'utiliser des équipements ou de la nourriture entre chacun des combats (seulement pour le premier). Si le joueur survit, il peut ensuite interagir avec les éléments non offensifs restants (s'il y en a) comme vu précédemment.

Les niveaux Lorsque le joueur remporte un combat il acquiert de l'expérience. Elle lui permet de devenir plus puissant afin de combattre des ennemis qui étaient auparavant hors de portée du joueur. Le joueur se voit attribuer un bonus de 1,2 fois son énergie, ainsi qu'un repos tout les cinq niveaux.

À chaque passage de niveau, des ennemis et des objets supplémentaires apparaissent, leur nombre dépendant de la difficulté. Le niveau des ennemis ajoutés est calculé en fonction de celui du joueur.

Lancement d'une partie Au lancement d'une partie, on propose au joueur d'entrer son pseudonyme et de choisir une difficulté entre novice, moyen et expert. C'est la difficulté d'une partie qui détermine les proportions des différents éléments du jeu :

- **Novice :**

- Quantités PNJ :

- * Ennemis : 50 au départ et 5 à chaque nouvelle génération ;

- * Aides : 50 ;
- Pourcentage Terrain : 80 % de la dépense énergétique du terrain ;
- Quantité d'objet : 100 répartis dans toute la carte ;
- Dimensions carte : 500 cases * 500 cases ;
- Energie initiale du joueur : 1000 ;
- Nombre de repos initial du joueur : 15 ;
- Nombre de tours entre chaque génération d'ennemis : 15.

- **Moyen :**

- Quantités de PNJ :
 - * Ennemis : 100 au départ et 10 à chaque nouvelle génération ;
 - * Aides : 25 ;
- Pourcentage Terrain : 100 % de la dépense énergétique du terrain ;
- Quantité d'objet : 75 répartis dans toute la carte ;
- Dimensions carte : 1000 cases * 1000 cases ;
- Energie initiale du joueur : 750 ;
- Nombre de repos initial du joueur : 10 ;
- Nombre de tours entre chaque génération d'ennemis : 10.

- **Expert :**

- Quantités de PNJ :
 - * Ennemis : 200 au départ et 200 à chaque nouvelle génération ;
 - * Aides : 25 ;
- Pourcentage Terrain : 120 % de la dépense énergétique du terrain ;
- Quantité d'objet : 50 répartis dans toute la carte ;
- Dimensions carte : 2000 cases * 2000 cases ;
- Energie initiale du joueur : 500 ;
- Nombre de repos initial du joueur : 5 ;
- Nombre de tours entre chaque génération d'ennemis : 5.

Ensuite une nouvelle partie commence.

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 ;
- **Son :** Active ou désactive le son ;
- **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 peut exécuter toutes les actions via l'interface graphique, il pourra également utiliser le pavé directionnel du clavier pour se déplacer et la touche échap pour accéder au menu. Un bouton lui permet d'accéder à l'inventaire. Une console affiche les interactions effectués, objets trouvés et utilisés. Il voit également une barre représentant son expérience et une barre représentant son énergie, avec leur valeur affichée au dessus, ainsi que les objets dont il est équipé, les repos dont il dispose et l'or de sa cagnotte. D'autre part si le même méchant est présent plusieurs fois sur la même case, un nombre apparaît à côté de son image pour spécifier le nombre.

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

PNJ : Personnage Non Joueur

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.