

Legend of Walls

Le cahier des charges

● Chapitre 1 - Présentation des acteurs

PIRAT3 est un groupe issu de l'université Claude Bernard Lyon 1, composé de 3 étudiants en 2ème année de licence informatique :

RANDRIAMITANDRINA Finaritra p1925996

RAKOTONIAINA Kevin Onjatiana p1804715

RAFAMATANANTSOA Maya p1112685

Dans le cadre d'un TP projet de l'UE LIFAP4, ils ont choisi de développer le jeu Fanorona, un jeu de plateau de l'île de Madagascar (dont ils sont originaires).

Découverte de l'idée :

Un des membres du groupe a proposé comme type de jeu, le jeu de points. Ils se sont rappelés d'un en particulier auquel ils avaient chacun déjà joué en classe. En développant leur idée, ils se sont aperçus que le Fanorona était méconnu du public non originaire de Madagascar.

Le projet est à réaliser avec un minimum de 40h de travail. Il devra être abouti avant la soutenance du TP du 10 mai.

Le présent cahier des charges **prévisionnel** est composé :

- d'une description détaillée du produit
- d'une prévision détaillée du déroulement du développement
- d'un diagramme de Gantt
- d'un diagramme des classes

● Chapitre 2 - Description du produit

1. Histoire du jeu :

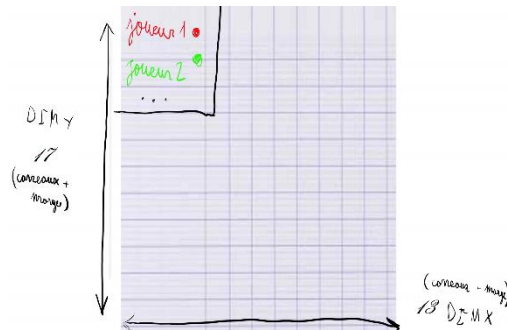
Fanorona est un jeu localement populaire. Le plateau est tracé sur une planche de bois ou une surface plane de pierre. Certains étudiants passent leur temps à jouer sur leur cahier pendant les interours. Comme ce jeu est très addictif, on aurait créé de toute pièce des interdictions pour empêcher les travailleurs de jouer au fanorona durant la saison des pluies.

On pourrait dire que c'est un Candy Crush malgache.

Ce jeu a tout pour être un parfait jeu mobile : on peut y jouer quand on veut et où on veut.

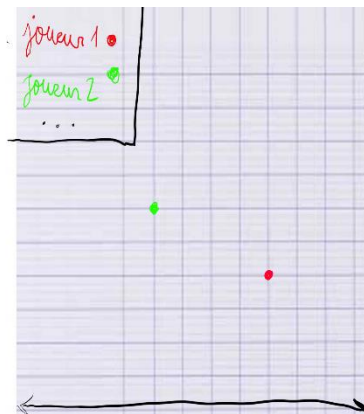
2. Principe et règles du jeu

- DIMENSION : 13 carreaux de largeur x 17 carreaux de longueur
- 2 joueurs



Les actions des joueurs :

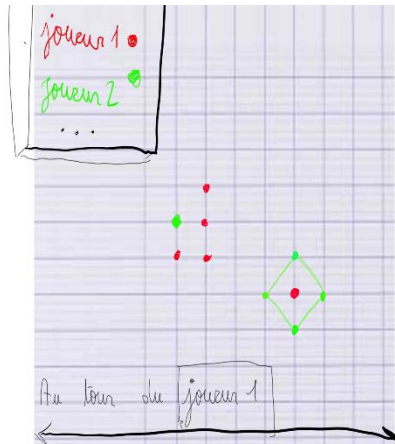
- Chaque joueur place un point *chacun son tour* sur le coin d'un carreau du cahier.



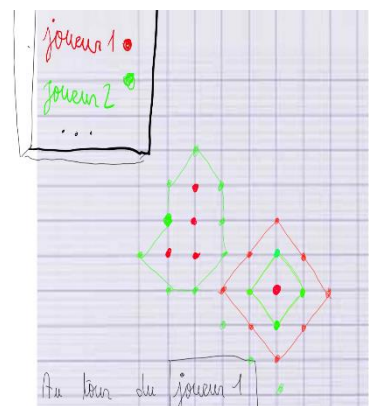
- Chaque joueur pourra tracer tous les murs qui compose une prison :



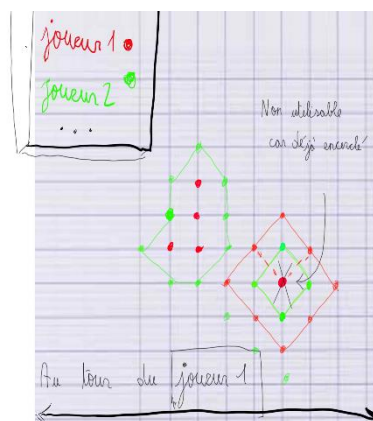
- Le but est d'encercler les points de l'adversaire c'est-à-dire les emprisonner en formant un mur de points :



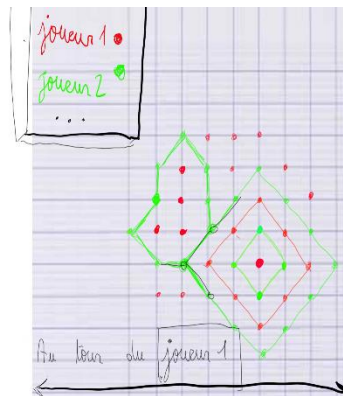
- Une prison adverse peut être encerclée à son tour :



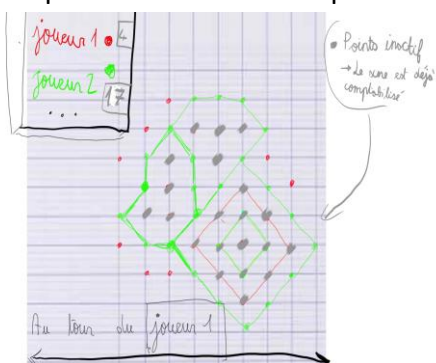
- Les points déjà emprisonnés seront inutilisables :



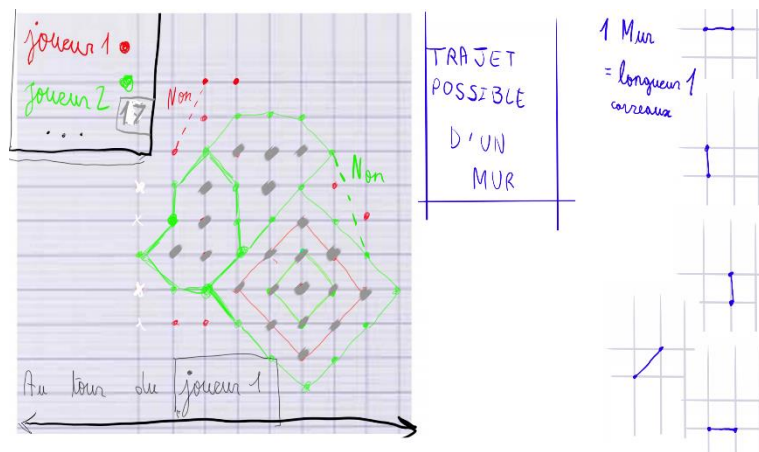
- Les murs d'une prison pourront être utilisés pour former une nouvelle prison :



- Le score équivaut au nombre de points adverses emprisonnés.



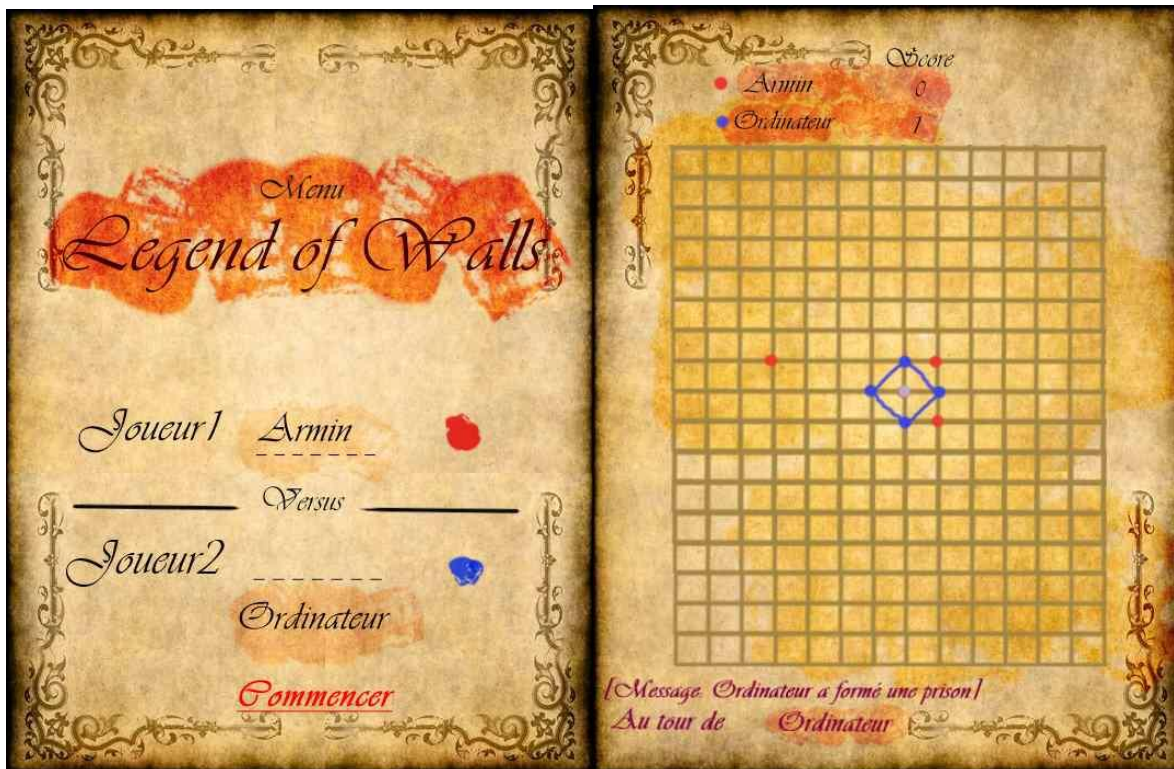
- Les points emprisonnés deviennent gris après le comptage du score.
- Si un joueur réussit à encercler un ou plusieurs points de l'adversaire, il a droit à un tour supplémentaire.
- Une prison ne pourra être formée que grâce à la reliure de plusieurs points *consécutifs*. (Pas de saut de coin)
- Par défaut le jeu s'arrête quand la grille est pleine



3. Résultats visés

Un jeu, en C/C++ sous linux, permettant à deux joueurs (Joueur1 VS Joueur2 ou Joueur VS Ordinateur) de s'affronter sur une interface graphique du Fanorona.

4. Détail de l'interface et des niveaux du jeu



Les messages affichés en dessous de la grille :

[Message : Joueur... a formé une prison

Joueur..., tu as gagné !

Joueur..., tu ne peux pas former ce mur

Joueur..., tu ne peux pas utiliser ce point]

● Chapitre 3 - Contraintes

- Archive à rendre avant la soutenance du TP le 10 mai
- Le jeu sera développé en C/C++ sous Linux
- Les bibliothèques utilisées seront SDL2 et une bibliothèque texte pour une version alpha
- Le code respectera le standard suivant : code indenté, variable ayant du sens, ...
- Le code sera géré et archivé sur GitLab
- La documentation du code sera produite par Doxygen
- Un diagramme des classes permettra d'avoir une vision de haut niveau de l'implantation
- L'équipe de développement utilisera les outils de debug et de profiling : gdb et valgrind

- **Chapitre 4 - Déroulement du projet**

Tâche 0 : rédiger le cahier des charges

Membres impliqués : Tous

Durée : 9 jours

Tâche 1 : définir le diagramme des classes

Membres impliqués : Tous

Tâche 2 : développement d'un 1er prototype en mode texte

Tâche 2.1 : écriture et test du module Point (Point.h/ .cpp)

Membres impliqués : Tous

Le Point est défini par une position, une couleur,
un état (mort (gris), vivant (couleur du joueur) et un score.

Le Point est statique

Tâche 2.2 : écriture et test du module Grille (Grille.h/ .cpp)

Membres impliqués : Tous

Ce module pourra charger une grille depuis un fichier. Le format d'un
fichier de données respectera la syntaxe suivante :

Tâche 2.3 : écriture et test du module Mur (Mur.h / .cpp)

Membres impliqués : Tous

Ce module permettra grâce au get/ set d'avoir accès en mode public
pour que le joueur puisse faire le tracé.

Tâche 2.4 : écriture et test du module Prison (Prison.h / .cpp)

Membres impliqués : Tous

Ce module permettra de modifier l'état d'un point

Tâche 2.5 : écriture et test du module Jeu (Jeu.h / .cpp)

Membres impliqués : Tous

Ce module propose toutes les fonctions pour gérer une étape du jeu :

- Reçoit en entrée un clic de l'utilisateur

- Création et mise en place d'un Point en utilisant le module Point et Grille
- Création d'un mur en utilisant les modules Point, Mur et Grille
- Mise à jour du score
- Test de fin de partie

Tâche 2.6 : écriture du module JeuModeTexte(JeuModeTexte.h / .cpp)

Membres impliqués : Tous

Récupère les données de Grille, Point et Prison et les affiche au format texte.

Tâche 2.7 : écriture et test du module Menu (Menu.h / .cpp)

Membres impliqués : Tous

Ce module permettra de choisir le pseudo/couleur du joueur et si on veut jouer contre l'ordinateur ou un deuxième joueur.

Tâche 2.8 : mise en place des modules et tests

Membres impliqués : Tous

Tâche 3 : développement d'un 2e prototype en mode graphique (SDL2)

Membres impliqués : Tous

Tâche 3.1 : exploration et compréhension de SDL2

Tâche 3.2 : Affichage d'images et tests de performance

Tâche 3.3 : Développement d'une boucle de jeu en SDL2 dans le module

JeuModeGraphique (JeuModeGraphique.h / .cpp) et intégration au reste du jeu

Tâche 3.4 : Test et debug

Tâche 4 : Développement d'un 3eme prototype avec ajout en mode graphique avec la musique et gestion des scores.

Tâche 5 : développement d'un 4e prototype avec ajout de l'option jouer contre l'ordinateur

● **Chapitre 5 - Diagramme de Gantt**

| | Semaine 1 | Semaine 2 | Semaine 3 | Semaine 4 | Semaine 5 | Semaine 6 | Semaine 7 | Semaine 8 | Semaine 9 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Tâche 0 | X | | | | | | | | |
| Tâche 1 | X | | | | | | | | |
| Tâche 2.1 | | X | | | | | | | |
| Tâche 2.2 | | X | | | | | | | |
| Tâche 2.3 | | X | | | | | | | |
| Tâche 2.4 | | X | X | | | | | | |
| Tâche 2.5 | | | X | | | | | | |
| Tâche 2.6 | | | | X | | | | | |
| Tâche 2.7 | | | | X | X | | | | |

| | | | | | | | | | |
|--------------|--|--|--|---|---|---|---|---|---|
| Tâche 2.8 | | | | X | | | | | |
| Tâche 3.1 | | | | | X | X | X | | |
| Tâche 3.2 | | | | | | X | X | | |
| Tâche 3.3 | | | | | | X | X | | |
| Tâche 3.4 | | | | | | | X | X | |
| Tâche 4 | | | | | | | X | X | X |
| Tâche 5 | | | | | | | X | X | X |

• Chapitre 6 - Diagramme des classes

