

Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction | 2 |
| 1.1 | Présentation de l'équipe | 2 |
| 1.2 | Présentation du projet | 2 |
| 1.2.1 | Règles | 2 |
| 1.2.2 | Stratégie de résolution | 2 |
| 2 | Environnement et jeu | 3 |
| 2.1 | Représentation du monde | 3 |
| 2.1.1 | Géométrie du monde | 3 |
| 2.1.2 | Plateau de jeu | 3 |
| 2.2 | Pièces et joueurs | 4 |
| 2.2.1 | Pièces | 4 |
| 2.2.2 | Joueurs | 4 |
| 2.2.3 | Formations de départ | 4 |
| 2.3 | Boucle de jeu | 5 |
| 2.3.1 | Sélection des options | 5 |
| 2.3.2 | Déroulement d'une partie | 5 |
| 2.3.3 | Conditions de victoire | 5 |
| 3 | Architecture du projet | 6 |
| 3.1 | Relations | 6 |
| 3.1.1 | Voisinages | 6 |
| 3.1.1.1 | Déplacements | 6 |
| 3.1.1.2 | Chagements de terrain | 6 |
| 3.1.2 | Positions de départ | 6 |
| 3.1.3 | Capture | 6 |
| 3.2 | Inclusions et organisation du projet (Makefile?) | 6 |
| 3.3 | Tests | 6 |
| 3.3.1 | Structure des tests | 6 |
| 3.3.2 | Programmation par le test | 6 |
| 4 | Conclusion | 7 |
| 4.1 | Difficultés rencontrées | 7 |
| 4.2 | Points à améliorer | 7 |
| 4.3 | Ce que le projet nous a apporté | 7 |

Chapitre 1

Introduction

1.1 Présentation de l'équipe

1.2 Présentation du projet

1.2.1 Règles

1.2.2 Stratégie de résolution

Chapitre 2

Environnement et jeu

2.1 Représentation du monde

2.1.1 Géométrie du monde

La première chose à faire était de définir des constantes, structures et variables qui allaient nous servir tout au long du projet tel que :

Les directions :

NORTH
NEAST
NWEST
WEST

SOUTH
SWEST
SWESE
EAST

Le type de couleur de la case :

NO-COLOR WHITE BLACK

L'occupant de la case :

NO-SORT SIMPLE-PAWN

Cette catégorie est vouée à s'agrandir au vu de l'ajout de différentes sortes de pions par la suite.

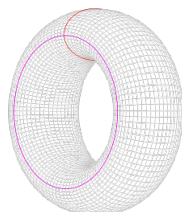
Egalement nous avons besoin des variables du monde :

taille : WORLD-SIZE longueur : WIDTH largeur : HEIGHT Nombre de cases Max : UNIT-MAN
nombre de joueurs : NB-PLAYERS

Avec ces variables on a créé la structure "world" qui nous permet d'assigner à chaque case du monde sa couleur et son occupation ce qui nous servira grandement dans la suite du jeu.

2.1.2 Plateau de jeu

Il faut définir un plateau de jeu pour l'utilisateur. Nous avons décidé de faire celui-ci de forme torique afin de rendre le jeu plus modulable. Il a donc fallu nous affranchir des effets de bord à l'aide du modulo. Ce choix a aussi été fait pour des raisons d'anticipation des futures modifications qui seront plus faciles en partant d'un modèle très généraliste tel que le tore.



Nous avons également pris par défaut des relations entre les cases de type hexagonale (figure 1.a) comme il nous l'était imposé en début de projet. Cependant dans le cadre d'un achèvement nous avons également implémenté un changement de terrain qui impact les relations faisant passer de case octogonale à des cases carré (figure 1.b) ou triangulaire (figure 1.c). Ces changements arrivent tout les x tours en fonction de la demande du joueurs.

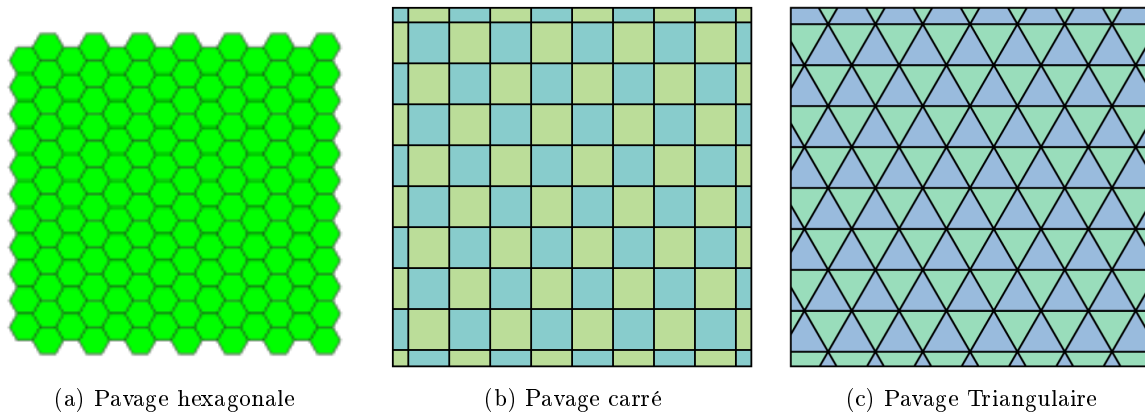


FIGURE 2.1 – Pavages disponible dans notre jeu

2.2 Pièces et joueurs

2.2.1 Pièces

Pour les pièces premièrement nous avons choisi de simple pions, pouvant se déplacer d'une seule case seulement si celle-ci est vide. Par la suite on a incorporé au jeu d'autres pièces le rendant plus intéressant. Chaque pièce a ses spécialités.

C'est pour cela que nous avons créé une structure qui rend les pièces modulable. Chaque pièce appartenant à cette structure est défini par :

- Son joueur : 1 ou 2
- Son nombre de déplacement : choisis en amont par le maître de la partie
- Sa couleur : White ou Black
- Son type : Pion, Tour, etc
- Sa position : $[0; \text{UNIT-MAX}-1]$

Le Pion : (insérer image)

Le pion est la pièce basique du jeu ayant un seul mouvement dans la direction de son choix. Cependant il est possible d'en faire une dame en augmentant son nombre de mouvement disponible dans les paramètres afin qu'il puisse se déplacer à la manière d'une dame aux échecs.

La Tour : (insérer image)

Elle peut comme aux échecs se déplacer seulement en direction des points cardinaux. Nous avons décidé de limiter ses déplacements au $\text{Max}(\text{longueur du plateau}, \text{largeur du plateau})$ car tant donné l'apparence torique de notre plateau de jeu ses mouvements pourraient être infinis.

L'éléphant : (insérer image)

Il se déplace uniquement suivant les 4 directions cardinales et possède dans le cas de base deux déplacements.

Le Roi premier : (insérer image)

Il se téléporte directement sur une case première si celle-ci n'est pas occupé par un autre pion.

2.2.2 Joueurs

2.2.3 Formations de départ

Nous avons mis en place deux formations de départs différentes, une première classique qui ressemble aux échecs en prenant en considération la forme torique de notre plateau.

La deuxième elle est plus anarchique et présente le plateau sous forme de champs de bataille avec deux

énorme blocs de pièces séparé par des tranchés.

Ces deux type de formation son modulable dans les pieces qui les composes car il est possible de choisir le nombre et le type de pieces "spécials" de la formation le reste est complété par des pions.

2.3 Boucle de jeu

2.3.1 Sélection des options

2.3.2 Déroulement d'une partie

2.3.3 Conditions de victoire

Chapitre 3

Architecture du projet

3.1 Relations

3.1.1 Voisinages

3.1.1.1 Déplacements

3.1.1.2 Chagements de terrain

3.1.2 Positions de départ

3.1.3 Capture

3.2 Inclusions et organisation du projet (Makefile ?)

3.3 Tests

3.3.1 Structure des tests

3.3.2 Programmation par le test

Chapitre 4

Conclusion

4.1 Difficultés rencontrées

4.2 Points à améliorer

4.3 Ce que le projet nous a apporté

- Environnement de jeu (variable pions etc)
 - Structure et Variables globales :
 - Plateau :
 - Pieces :
- Joueurs
 - Les joueurs sont définis par une structure qui contient un certain nombre d'élément :
 - Un index qui différencie les joueurs entre eux
 - Le nombre de pion qu'il possède
 - Le nombre de ses pions capturés
 - Une liste de ses pions
 - Une couleur
- Boucle de jeu
 - Déroulement de la partie
 - Conditions de victoire
- Architecture
 - Relation de jeu
 - Relations de voisinage
 - Relation d'initialisation de partie
 - Interaction entre les joueurs (prisons)
 - Interaction entre le monde et les joueurs
 - Inclusion
 - Makefile
 - Organisation des fichiers projets
 - Organisation des fichiers de test
 - Complexité
- Tests
 - Structures Tests
 - Notre utilisation des tests pour le bon fonctionnement du projet
- Difficultés rencontrées
 - Problème de structure
 - Difficulté d'affichage
 - Aléatoire
- Conclusion
 - Points à améliorer
 - Les test
 - Travailler l'indépendance
 - Ce que le projet nous a apporté
 - Découverte de GitHub
 - Découverte du Makefile
 - Progression en C