

Yannis Hutt 11408376  
Julien Cadier 11510421  
Randy Andriamaro 11512256

## Cahier des charges



# Sommaire

1. Présentation du projet
2. Description du projet
3. Contraintes
4. Déroulement du projet
5. Diagramme des classes

# 1.Présentation du projet

Dans le cadre pédagogique de l'unité d'enseignement de LIFAP4, nous devons, par groupe, concevoir une application ou un jeu vidéo en C++.

Le groupe qui va concevoir le jeu vidéo est composé Yannis Hutt, Julien Cadier, Randy Andriamaro

Le présent cahier des charges est composé :

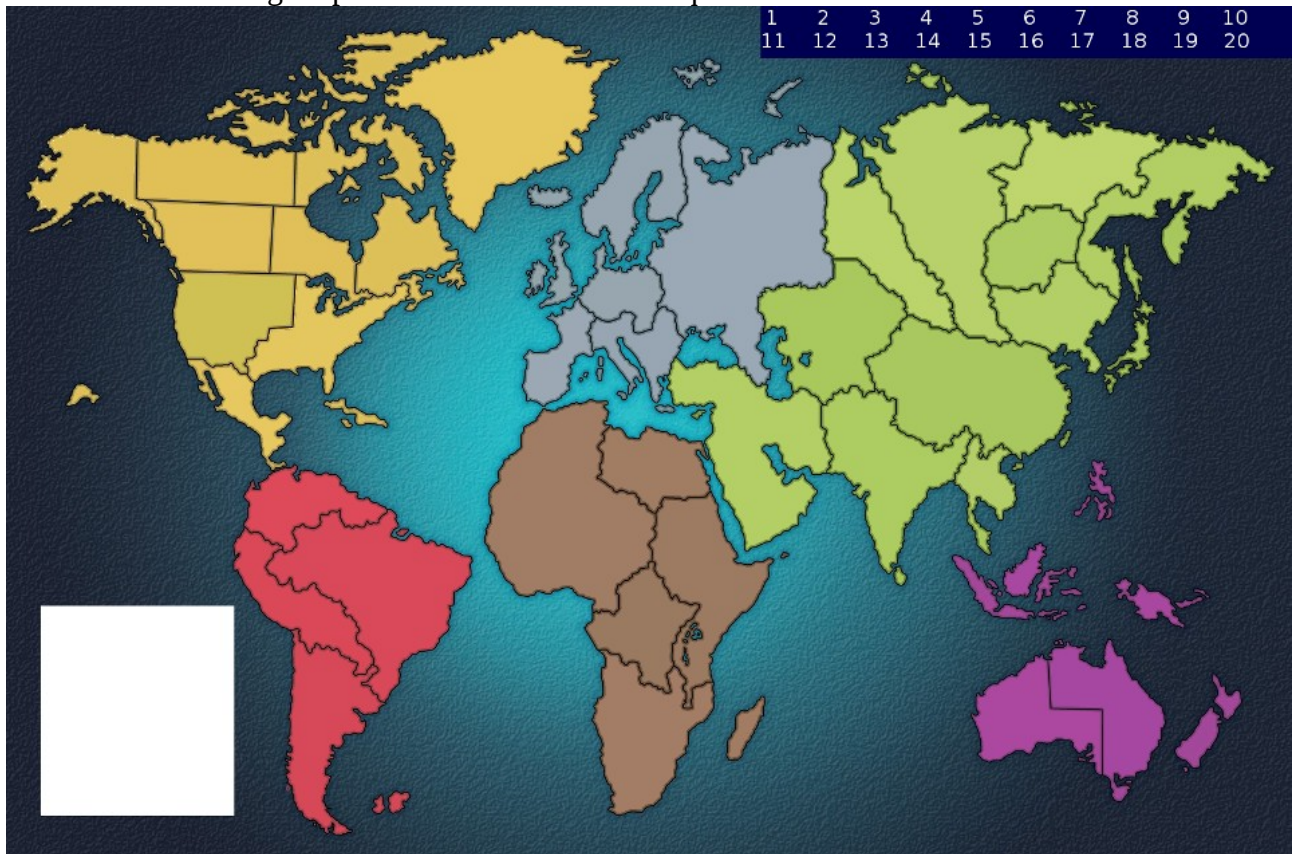
- D'une description détaillée du produit
- D'une prévision de temps pour le déroulement du projet
- Diagramme de Gantt
- Diagramme des classes

## 2.Description du projet

Le jeu est une variante du jeu de plateau Risk. Le jeu est basé sur la stratégie, le but du jeu est de conquérir les régions et les pays, et le but principal du jeu est basé sur la règle de la domination (conquérir toute la carte ou être le joueur le plus puissant ou influant).

Le Jeu se déroule sur une carte plus ou moins grande en fonction du nombre de joueurs (la partie peut être de 2 à 5 joueurs).

La carte est divisée par pays ensuite en régions, la défense de ces territoires doit être assurée avec soin, sans quoi un combat supposé facile peut se transformer en défaite. Or, la perte inattendue d'un territoire ou d'une région peut avoir de lourdes conséquences.



Chaque pays sera d'une couleur différente sur la carte ainsi que les joueurs pour permettre de différencier les possessions de chaque joueur sur la carte ainsi que le nombre d'unités.

Le jeu est différencié en différentes phases :

-Phase Renfort : Les renforts sont obtenus de 2 manières différentes, le joueur compte le nombre de territoires occupés par son armée, il divise ce nombre par 3 (arrondi en dessous, et reçoit autant de régiments que l'indique le résultat (au minimum, un joueur reçoit toujours trois régiments en renfort).

Exemple : le joueur occupe 17 territoires, il prend donc 5 régiments  $17/3=5$  (les 2 points restants ne compte pas)

Si au début de son tour de jeu, le joueur contrôle un ou plusieurs Pays, il obtient des renforts supplémentaires (pour contrôler un pays, il doit occuper tous les territoires du pays). Ces renforts sont octroyés suivant un barème qui dépendra de la taille du pays.

-Phase Combat : Une fois que tous les renforts ont été déployés, le joueur peut maintenant passer à l'attaque !

On ne peut attaquer un territoire ennemi que si la frontière entre les 2 régions est commune, ou si l'attaquant possède au moins 2 régiments sur son territoire. Avant d'attaquer un territoire le joueur doit désigner un territoire à attaquer ainsi que le nombre de régiments qu'il engage dans le combat (entre 1 et 3 régiments).

Pour la défense, le joueur attaqué peut lui aussi choisir sa défense (entre 1 et 2 régiments).

Le nombre de régiments choisis représente un nombre de dés à lancer (dé à 6 faces), la somme des dés de l'attaquant et la somme des dés du défenseur sont comparés et un gagnant est désigné pour cette manche.

-Phase Manœuvre :

Le joueur procède à cette action lorsqu'il estime que sa phase de combat est terminée pour la durée de ce tour de jeu. Il peut alors déplacer autant de régiments qu'il souhaite de l'un de ses territoires sur un territoire voisin lui appartenant, comme après un combat il doit toujours laisser au moins 1 régiments sur le territoire de départ.

Le joueur ne peut effectuer qu'une seule manœuvre. Puis c'est au joueur suivant de jouer.

### 3. Contraintes

-Le jeu est développé en C++ sur Linux.

-Les bibliothèques utilisées seront SDL2 et une bibliothèque texte pour le version alpha du projet.

-Le code respectera les standards et conventions d'écritures telle que l'indentation, nom de variables avec du sens ainsi que une arborescence des fichiers logique.

-Nous utiliserons un gestionnaire de fichier qui est Mercurial.

-La Documentation du code sera produite par Doxygen.

-Le diagramme des modules permet de montrer l'implantation du code et du projet pour permettre une vision d'ensemble.

-Un diagramme de Gantt permet de voir le temps nécessaire pour la conception de l'application et la répartition des tâches dans la durée.

### 4. Déroulement du projet

(nom des fichiers à préciser)

Tâche 0 : Définir le cahier des charges

Tâche 1 : Définir le diagramme des modules

Tâche 2 : Développement du prototype en mode texte  
(sous-Tâche)

- Tâche 2.1 : écriture et test du module Jeu (Jeu.h/Jeu.cpp)
- Tâche 2.2 : écriture et test du module Terrain (Terrain.h/Terrain.cpp)
- Tâche 2.3 : écriture et test du module Pays (Pays.h/Pays.cpp)
- Tâche 2.4 : écriture et test du module Région (Region.h/Region.cpp)
- Tâche 2.5 : écriture et test du module Combat (Combat.h/Combat.cpp)
- Tâche 2.6 : écriture et test du module Joueur (Joueur.h/Joueur.cpp)
- Tâche 2.7 : écriture et test du module JeuModeTxt (JeuModeTxt.h/JeuModeTxt.cpp)

Tâche 3: Développement du 2ème prototype en mode graphique (SDL2)

- Tâche 3.1 : affichage de sprites et tests de performance
- Tâche 3.2 : Développement d'une boucle de jeu en SDL2 (JeuModeSDL.h/JeuModeSDL.cpp) et intégration du jeu
- Tâche 3.3: Test et debug

Diagramme de Gantt

	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7	Semaine 8
Tâche 0								
Tâche 1								
Tâche 2.1								
Tâche 2.2								
Tâche 2.3								
Tâche 2.4								
Tâche 2.5								
Tâche 2.6								
Tâche 2.7								
Tâche 3.1								
Tâche 3.2								
Tâche 3.3								

Julien
Yannis
Randy

# 5.Diagramme des classes

