

## Manuel d'utilisation du RISK



Année 2017/2018

Semestre 2

Professeur Encadrant : Rafaël Angarita

SUY Van-Kevin  
JOLY Guillaume  
SAEZ Alicia

## SOMMAIRE

1. Introduction du jeu .....	Pages 3 et 4
2. UML.....	Page 5
3. Description des classes.....	Pages 6 à 9
4. Problèmes rencontrés.....	Pages 10 et 11

## 1. Introduction du jeu

Afin de lancer le jeu, il faut que le main soit ouvert sur Eclipse, soit la classe “Maître\_du\_jeu”. La page de présentation s’ouvre :



Il suffit de sélectionner le nombre de joueur et de cliquer sur jouer. Un temps de latence est nécessaire pour calculer la position des futurs pions. Après quelques secondes d’attendre le plateau de jeu apparaît :



### Phase d'initialisation

Il faut appuyer sur le bouton “ dessiner ” en bas à droite pour afficher les pions. Ainsi les pions seront placés aléatoirement sur le plateau. Toutes les zones seront occupées.

**Tour n°1** : Chaque joueurs posent leur pions à tour de rôle uniquement sur les zones qui leur ont été attribué initialement.

Image des différents pions :



Cavalier



Soldat



Tank

Le soldat vaut une unité tandis que le cavalier et le tank valent respectivement 3 et 7 unités. Chaque joueurs est caractérisé par une couleur de pion.

### Phase de déplacement et d'attaque

A l'heure d'aujourd'hui ( Mardi 5 juin 2018 ), nous ne pouvons pas déplacer les pions pour procéder aux déplacements et aux attaques.

Nous envisageons d'implémenter un bouton déplacer. Il faudra cliquer sur la zone ou l'unité voulant être déplacé se situe, puis cliquer sur déplacer, sélectionner le type et le nombre de chaque unités voulant être déplacer et finir en cliquant sur la zone d'arrivée.

## 2. UML

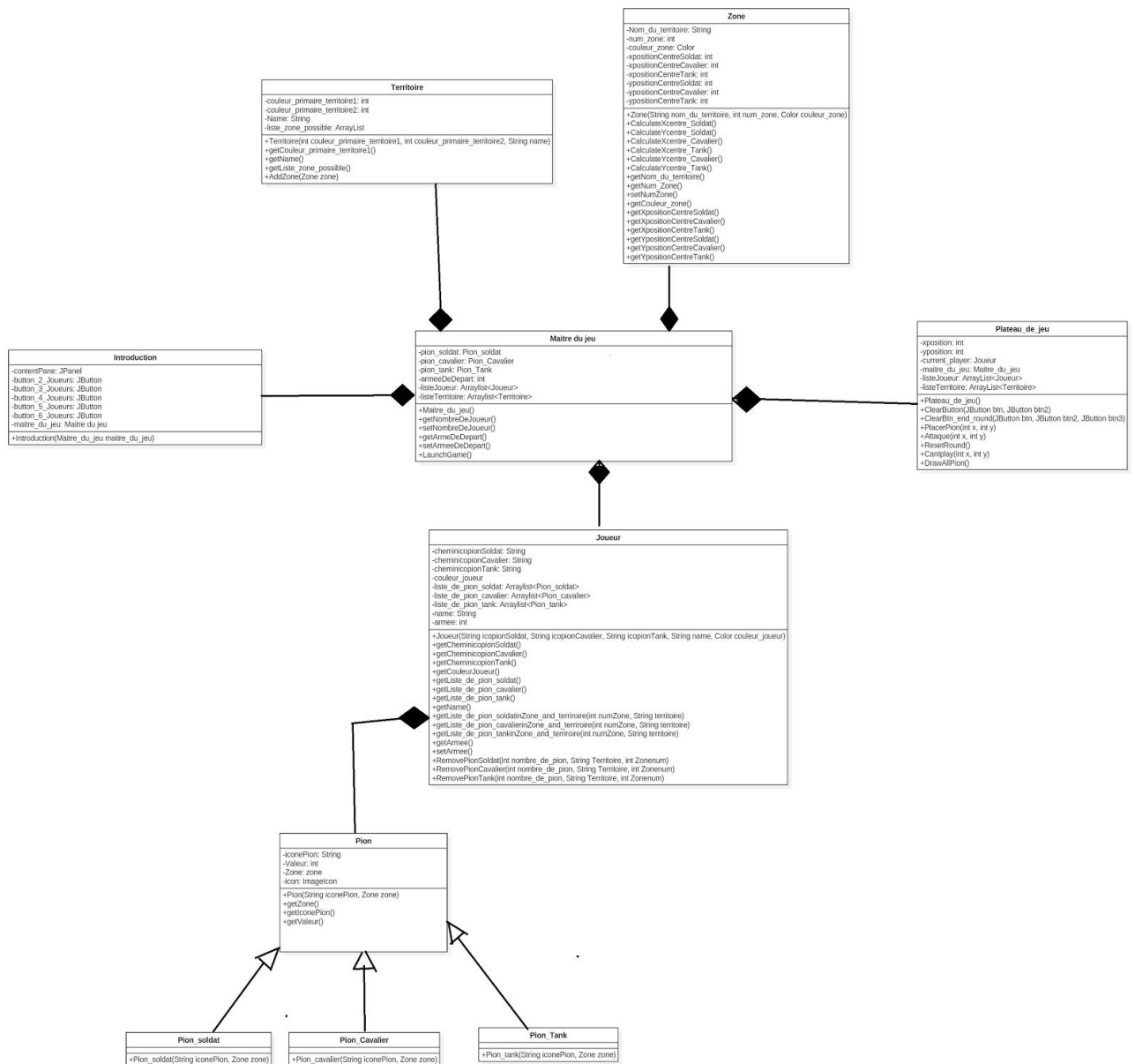


Diagramme UML avec les Getters et les Setters

### 3. Description des classes

#### Classe Introduction

Cette classe Introduction permet de choisir le nombre de joueur allant de 2 à 6.

Deux possibilité s'offraient à nous pour configurer cette page. Nous pouvions soit cliquer sur le nombre de joueur et lancer automatiquement le jeu, soit sélectionner le nombre de joueurs et ensuite valider ce choix.

Nous avons opté pour la deuxième configuration afin d'éviter de lancer une partie si le choix du nombre de joueurs est erroné à cause d'une erreur de l'utilisateur.

En cliquant sur le bouton "Jouer", la partie se lance et le nombre de joueurs est retourné et sauvegardé.

#### Classe principal maître du jeu (MAIN)

Cette classe permet d'une part de lancer la classe "Introduction" et d'autre part, la classe "Plateau de jeu". Elle définit qui joue et qui a gagné.

C'est ici que les pions sont créés afin de récupérer les attributs et les méthodes dans les classes correspondantes. Les informations nécessaires sont : le chemin de l'image et la zone)

On initialise le nombre de joueurs, c'est-à-dire que nous créons des objets joueur via la classe "Joueur"

On initialise les territoires, c'est-à-dire que nous créons des objets territoire via la classe "Territoire".

On initialise les zones, c'est-à-dire que nous créons des objets territoire via la classe "Zone".

Trois listes sont par la suite créés :

- Une liste Joueur : 6 joueurs sont ici créés par défaut, ils ne seront pas forcément joués.
- Une liste Joueurs Possibles : Reprend le nombre exact de joueur initialisés précédemment. Cette liste reprend la liste Joueur et ne valide que le nombre de joueurs nécessaires.
- Une liste Territoire : Elle permet d'assigner une couleur pour chaque territoire en RGB. Les zones qui sont aux nombres de 7, seront des dégradés de la couleur du territoire dans lequel ils se situent. Nous créons un dégradé en modifiant le nombre d'une des composante de la couleur. Les territoires allant de 1 à 7 seront du plus claires au plus foncées.

Elle permet aussi de créer le nombre de pions exactes selon le nombre de joueur sélectionné, et de les répartir aléatoirement sur le plateau de jeu. On s'est aussi assuré

qu'un seul pion est placé dans une zone. Soit il est impossible lors de l'initialisation d'avoir deux joueurs différents sur la même zone.

### **Classe Plateau De Jeu**

La carte du monde à été réalisée sur gimp et paint. La carte du monde était initialement vide, nous l'avons personnalisé avec gimp pour la délimitation des territoires et nous avons ensuite utilisé paint pour les couleurs.

Cette classe permet d'afficher le plateau de jeu ( = carte du monde ) mais aussi des éléments supplémentaires :

- 3 boutons de pions imagés ( soldat, cavalier, canon) permettant de placer ses différents pions sur le jeu.
- 1 bouton "Fin de tour" permettant de valider ses pions et de donner la main au joueurs suivant.
- 1 bouton "Effacer" pour effacer les pions qui n'apparaissent pas.
- 1 bouton "Dessiner" pour re-dessiner les pions
- Création d'un tableau récapitulatif pour chaque joueur. C'est un tableau personnel et privé avec le nombre de pions qu'ils ont et dans quelle zones. Les autres joueurs ne peuvent pas accéder au tableau des autres.
- Ajout et affichage des pions des différents joueurs. Afin de reconnaître les pions sur le jeu, chaque joueurs possèdent une couleur de pions bien distinctes.
- Afficher un message pour savoir qui doit jouer et ses possibilités.
- Trois Jauges permettant de sélectionner le nombre de pions à déplacer afin de gérer les déplacements.
- La classe plateau gère aussi les déplacements et les attaques des soldats

### **Classe Territoire**

Cette classe permet de définir l'objet "Territoire" possédant 3 attributs : 2 couleurs primaires fixes et un nom. Elle contient initialement une liste zone vide qui est par la suite rempli dans la classe Maître du Jeu.

### **Classe Zone :**

Cette classe permet de définir l'objet "Zone" possédant 3 paramètres :

- Nom du territoire
- Numéro de la zone
- Couleur de la zone

Elle permet de calculer le centre de chaque zone pour placer un pion de chaque armée (soldat, tank, cavalier). Les pions auront donc un emplacement spécifique sur chaque zone calculée avant le lancement du jeu, d'où le temps de latence.

```
Partie à 6 joueurs //introduction
Game lancée ! //maitre du jeu
Tour de Annie //maitre du jeu
Asie zone 1= 1027 128
Asie zone 2= 1121 123
Asie zone 3= 1278 123
Asie zone 4= 1247 277
Asie zone 5= 1125 282
Asie zone 6= 1029 255
Asie zone 7= 934 294
Afrique zone 1= 680 337
Afrique zone 2= 760 326
Afrique zone 3= 848 340
Afrique zone 4= 746 427
Afrique zone 5= 879 437
Afrique zone 6= 843 530
Afrique zone 7= 830 605
Amerique_du_nord zone 1= 520 44
Amerique_du_nord zone 2= 204 97
Amerique_du_nord zone 3= 290 150
Amerique_du_nord zone 4= 185 208
Amerique_du_nord zone 5= 306 219
Amerique_du_nord zone 6= 180 260
Amerique_du_nord zone 7= 194 328
Amerique_du_sud zone 1= 329 460
Amerique_du_sud zone 2= 426 478
Amerique_du_sud zone 3= 359 537
Amerique_du_sud zone 4= 456 538
Amerique_du_sud zone 5= 416 581
Amerique_du_sud zone 6= 398 629
Amerique_du_sud zone 7= 382 694
Europe zone 1= 675 241
Europe zone 2= 725 190
Europe zone 3= 775 179
Europe zone 4= 821 138
Europe zone 5= 900 144
Europe zone 6= 956 120
Europe zone 7= 658 111
Oceanie zone 1= 1575 470
Oceanie zone 2= 1407 605
Oceanie zone 3= 1482 582
Oceanie zone 4= 1566 611
```

## Classe Joueur

Cette classe permet de définir l'objet "Joueur" possédant 5 paramètres :

- Icône du pion du soldat
- Icône du pion du cavalier
- Icône du pion du tank
- Nom
- Couleur du joueur

Elle possède 1 listes par types de pions. Soit 3 liste :

- liste\_de\_pion\_soldat
- liste\_de\_pion\_cavalier
- liste\_de\_pion\_tank



Ces listes permettent de savoir combien chaque joueur possède de soldat, cavalier ou tank.

### **Classe Pion**

C'est la classe mère de *Pion\_Cavalier*, *Classe Pion\_soldat*, *Classe Pion\_Tank*.

Elle permet de définir la position du pion sur une zone et le chemin d'accès à son image.

### **Classe Pion\_Cavalier**

Classe fille de la classe pion

### **Classe Pion\_soldat**

Classe fille de la classe pion

### **Classe Pion\_Tank**

Classe fille de la classe pion

## 4. Problèmes rencontrés

La taille de l'Europe nous a posé problème. En effet, étant trop petite, il est impossible de poser plusieurs pions dessus dans toutes les régions.

Solution : *Nous avons donc rétrécis les pions grâce à Gimp avant de les importer dans le code Java.*

Le nombre de pion sur le plateau est aussi un problème car si nous voulons poser 5 soldat et un cavalier sur une zone, cela sera illisible.

Solution : *Nous avons décidé d'agrandir l'Angleterre et l'Espagne pour une meilleure visibilité des régions. Nous avons mis en place un tableau récapitulatif d'armée par joueurs pour ne pas afficher beaucoup de pions sur la carte. Uniquement 3 pions maximum sur chaque territoire sont possibles pour une meilleure visibilité. Chaque pion a des coordonnées précises dans chaque zone.*

Chaque pion ayant la même forme, il est impossible de discerner ses propres pions et ceux des adversaires.

Solution : *Une couleur est désignée pour chaque joueur.*

Selon les comptes de chaque codeur, impossibilité de lancer le jeu car les directions ne sont pas les mêmes et doivent être changées à la main après chaque push

Solution : *Correction du chemin qui n'est donc plus personnalisé*

Taille du plateau de jeu change selon la taille de l'écran. Il est donc difficile voire impossible de jouer sur certains écrans.

Solution : *Aucun test sur les écrans de l'ISEP car le plateau de jeu est trop grand. Possibilité de changer la taille de la fenêtre en fin de projet s'il nous reste du temps pour chercher.*

La gestion d'ajout des pions sur la carte : lorsque nous ne pouvions plus ajouter un type de pion, les deux autres types devenaient indisponibles aussi.

Solution : *Créer une variable pour chaque type de pion qui indique la possibilité de le poser ou non.*

Les deux derniers pions posés durant les différents tours n'apparaissent pas.

Solution : *Bouton Refresh créé permettant de les afficher.*

Problème avec le bouton Refresh : Ce bouton dessine les pions manquants mais il y a un problème lors de la phase de déplacement. En effet, ce bouton affiche 2 fois le même pion : l'ancienne et la nouvelle position.

Solution : *Création de deux boutons : un pour effacer et un pour redessiner*

Problème de l'affichage des pions lors des déplacements. Initialement, nous voulions faire glisser les pions interactivement d'une zone à l'autre.

Solution : Nous allons utiliser les deux boutons précédent : “Effacer” et “Redessiner”.

Problème rencontré avec la répartition et la réception des pions et territoires lors de l'initialisation du jeu. En effet, nous pensions que les joueurs recevront X pions plus des pions données durant la partie initialisation du jeu.

Conclusion : Les X soldats donnés comprennent les armées reçues et placées en phase d'initialisation.

Solution : Changer dans le code le nombre d'armée reçue lors de la phase d'initialisation.

Problème lors de la phase de combat entre deux unités : Nous voulions faire combattre une à une chaque unité l'une après l'autre, puis leur générer un nombre correspondant à leur puissance. Mais cela ne respectait pas la règle de l'ordre de puissance de combat et rendait impossible le fait que l'attaquant puisse attaquer avec trois unités, tandis que le défenseur puisse se défendre avec deux unités, ni la règle de la priorité défensive.

Solution : Les armées de l'attaquant et du défenseur vont chacune commencer par générer un nombre aléatoire entre leur puissance minimale et leur puissance maximale. Ce nombre décidera de leur ordre de passage en les triant dans la liste. En cas d'égalité c'est la priorité d'attaque ou de défense qui décidera l'ordre d'attaque

Problème lors de la phase de combat résultant d'une égalité lorsque plusieurs armées attaquent : Les priorités d'attaque ou de défense ne sont pas respectées lors d'un tirage avec égalité.

Solution : la solution n'a toujours pas été trouvée à ce jour.