

# Projet 2019

## ***Risk***

# Auteurs

Nom	Adresse
Boursier Louis	<a href="mailto:louis.boursier@etu.univ-nantes.fr">louis.boursier@etu.univ-nantes.fr</a>
Filaudeau Éloi	<a href="mailto:filaudeau.eloi@etu.univ-nantes.fr">filaudeau.eloi@etu.univ-nantes.fr</a>
Lasherme Loïc	<a href="mailto:loic.lasherme@etu.univ-nantes.fr">loic.lasherme@etu.univ-nantes.fr</a>

# Introduction

Au cours de ce projet nous avons pour but de développer un logiciel réparti permettant de jouer au jeu du Risk. Pour se faire nous devons mettre en oeuvre les connaissances vues en cours pour nous permettre de réaliser au mieux ce projet.

Le jeu du Risk est à la base un jeu de plateau qui se joue de 2 à 6 joueurs. À tour de rôle, chaque joueur va définir un territoire sur lequel il placera une unité jusqu'à remplir chaque zone. Une fois cela fait, les joueurs pourront compléter les forces de leur armée sur les zones occupées par celle-ci afin de se libérer des unités en main. Une fois cela fait, le jeu peut enfin commencer et le joueur qui possédera tous les territoires se verra remporter la victoire.

Nous avons donc pour objectif de penser, modéliser, et implémenter ce jeu décrit brièvement ci-dessus.

# Organisation du document

# Analyse du domaine

## Introduction

### Objectif

Ce chapitre constitue le premier livrable d'une série de quatre chapitres destinés à fournir une analyse et une conception par objets complètes répondant au cahier des charges qui nous a été fourni. Ce document présente l'ensemble de la démarche suivie ainsi que les résultats obtenus lors de la phase de l'analyse du domaine de notre système. Il se décompose en plusieurs parties.

Dans la première partie, nous présentons de manière détaillée l'ensemble des cas d'utilisation que nous avons dégagés lors de l'analyse. Nous utiliserons pour cela le canevas proposé par Cockburn que nous compléterons par des instantanés ainsi que par des post-conditions exprimées en OCL (Object Constraint Language) et quelques scenarii. Cette partie constitue une étape clé de la phase de l'analyse du domaine.

Dans la deuxième partie, nous présentons le diagramme de classes métiers (i.e. diagramme de classes au niveau analyse) que nous avons construit à partir de l'analyse réalisée. Ce diagramme fournit une vue statique et synthétique du domaine de notre projet. Cette partie constitue également une étape clé de la phase de spécification des besoins.

Dans la troisième et dernière nous fournissons le dictionnaire des données que nous avons construit suite à l'analyse du domaine. Il s'agit d'un listing de l'ensemble des termes relatifs au domaine étudié ainsi que leur définition précise.

Dans une quatrième et dernière partie,

### Organisation du chapitre

Ce chapitre est organisé en \$n\$ sections. Dans la première section, nous décrirons....

## Cas d'utilisation

### Mise en place d'un jeu

#### Use Case Jouer

#### Revision History

Date	Author	Description of change

Use Case Template. Copyright (c) 2004-2005 TechnoSolutions Corporation

(Learn more about “TopTeam for Use Cases” at [www.technosolutions.com](http://www.technosolutions.com))

Permission is hereby granted, free of charge, to any person obtaining a copy of this document and its associated documentation, to use the document on their projects for any commercial or non-commercial purpose. However you may not publish, distribute, sublicense, and/or sell copies of this document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT. TECHNOSOLUTIONS CORPORATION MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS DOCUMENT FOR ANY PURPOSE.

**Use Case:** Jouer

**Id:** UC- 1

**Description** Résumé d’une partie de RISK dans sa globalité.

**Level:** High Level Summary

**Primary Actor** Jeu

**Supporting Actors** Joueur

**Stakeholders and Interests**

**Pre-Conditions**

Au moins 3 joueurs sont présents.

**Post Conditions**

Success end condition

Le jeu s’est déroulé correctement jusqu’à l’obtention d’un joueur gagnant selon les règles du jeu RISK.

Failure end condition:

Un des joueurs ne réalise pas d’action et bloque le jeu.

Minimal Guarantee

Pas de garantie.

**Trigger**

Un groupe de joueurs veut faire une partie.

## Main Success Scenario

1. Les joueurs mettent en place la partie (Voir UC2: Initialisation)
2. Les joueurs jouent tour à tour (Voir UC3: Tour)
3. On répète 2. tant qu'il y a plus d'un joueur sur les territoires.
4. Le dernier joueur est le gagnant. La partie est terminée.

**Frequency:** Une seule fois

## Use Case Initialisation

### Revision History

Date	Author	Description of change

Use Case Template. Copyright (c) 2004-2005 TechnoSolutions Corporation

(Learn more about "TopTeam for Use Cases" at [www.technosolutions.com](http://www.technosolutions.com))

Permission is hereby granted, free of charge, to any person obtaining a copy of this document and its associated documentation, to use the document on their projects for any commercial or non-commercial purpose. However you may not publish, distribute, sublicense, and/or sell copies of this document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT. TECHNOSOLUTIONS CORPORATION MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS DOCUMENT FOR ANY PURPOSE.

**Use Case:** Initialisation

**Id:** UC- 2

**Description** Initialisation et mise en place d'une partie de RISK

**Level:** User Goal

**Primary Actor** Jeu

**Supporting Actors** Joueurs

**Stakeholders and Interests** Pioche Défausse Carte Territoire Continent

**Pre-Conditions**

Au moins 3 joueurs sont présents.

## Post Conditions

### Success end condition

Le jeu a été correctement mis en place selon les règles de RISK. Chaque territoire est en possession d'un joueur. Chaque territoire supporte une ou plusieurs armées du joueur possédant ce territoire. La pioche est remplie avec toutes les cartes du jeu dans un ordre aléatoire.

### Failure end condition:

Un des joueurs ne place pas une de ses armées lorsque c'est à son tour de le faire. Il bloque l'initialisation.

### Minimal Guarantee

La partie ne peut pas continuer tant que le jeu n'a pas été correctement initialisé.

## Trigger

Les joueurs sont prêts à jouer.

## Main Success Scenario

1. On distribue à chaque joueur des unités d'armé selon le nombre de joueurs total. 35 unités pour 3 joueurs, 30 pour 4, 5 pour 25 et 6 pour 20.
2. L'ordre de jeu des joueurs à l'initialisation et lors des tours qui suivront est déterminé aléatoirement avec un lancé de dé 6. Chaque joueur lance le dé et le joueur ayant le plus grand chiffre commence. Les joueurs jouent ensuite dans le sens des aiguille d'une montre après le premier joueur **Extension 1**.
3. Les joueurs posent tour à tour une de leur unité d'armé sur les territoires non occupés par un autre joueur, avec obligation de placer sur un territoire innocupé si possible. **Extension 2**.
4. Répéter 3. tant qu'il reste des armées aux joueurs.
5. On mélange toutes les cartes du jeu et les met dans la pioche.

## Extensions

1. Si il y a une égalité entre les premiers, ces joueurs rejouent jusqu'à ce qu'il y ai un gagnant.
2. Si le territoire était innocupé par une armée, le joueur qui y pose l'armée se l'approprie.

**Frequency:** Une seule fois

## Use Case Tour

## Revision History

Date	Author	Description of change



Date	Author	Description of change

Use Case Template. Copyright (c) 2004-2005 TechnoSolutions Corporation

(Learn more about “TopTeam for Use Cases” at [www.technosolutions.com](http://www.technosolutions.com))

Permission is hereby granted, free of charge, to any person obtaining a copy of this document and its associated documentation, to use the document on their projects for any commercial or non-commercial purpose. However you may not publish, distribute, sublicense, and/or sell copies of this document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT. TECHNOSOLUTIONS CORPORATION MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS DOCUMENT FOR ANY PURPOSE.

**Use Case:** Tour

**Id:** UC- 3

**Description** Le tour d’un joueur lors d’une partie de RISK

**Level:** User Goal

**Primary Actor** Joueur

**Supporting Actors** Jeu

**Stakeholders and Interests** Pioche Défausse Carte Territoire Continent Main

**Pre-Conditions**

C’est au joueur de jouer. La partie n’est pas terminée.

**Post Conditions**

Success end condition

Le joueur a déployé ses armées gagnées dans ce tour lors de la phase de renfort sur son territoire. Il a éventuellement attaqué d’autres joueurs et renforcé ses territoires.

Failure end condition:

Le joueur ne réalise aucune action.

## Minimal Guarantee

Le joueur ne peut pas perdre lors de son tour.

## **Trigger**

L'initialisation est terminée. Le joueur précédent le joueur courant a fini son tour.

## **Main Success Scenario**

1. Le joueur entre en phase de renfort.
  - a. Le joueur reçoit des armées selon les territoires qu'il possède. Nombre d'armées =  $\text{arrondiInférieur}(\text{nombreDeTerritoirePossédés}/3)$ . **extension 1**
  - b. Le joueur peut échanger des jeux de 3 cartes de sa main contre des armées. Voir UC4 Echange de cartes.
  - c. Le joueur pose ses nouvelles armées sur ses territoires occupés de la façon qu'il souhaite.
2. Le joueur entre en phase d'attaque. **extension 2**
  - a. L'attaquant choisit un de ses territoires depuis lequel il attaque, possédant au moins 2 armées, et un territoire d'un autre joueur à attaquer avec lequel il possède une voie maritime directe ou une frontière partagée.
  - b. Le joueur attaquant et le joueur défenseur mettent en jeu un nombre de dés correspondant au nombre d'armées qu'ils veulent faire combattre. Ce nombre ne doit pas dépasser:
    - i. Le nombre d'armées présentes sur le territoire
    - ii. 3 pour l'attaquant
    - iii. 2 pour le défenseur
  - c. On lance les dés. Le dé le plus faible de chaque paire de dé attaquant/défenseur fait perdre une armée à son possesseur. Dans le cas d'une égalité, c'est le défenseur qui gagne. Le territoire attaquant ne peut descendre en dessous d'une armée. **extension 3**
3. Le joueur rentre en phase de fortification. **extension 4**
  - a. Le joueur peut déplacer une fois entre 1 et n-1 armées d'un de ses territoires à un de ses autres si le chemin entre les deux est composé uniquement de ses propres territoires.
4. Le tour du joueur prend fin et laisse place au suivant.

## **Extensions**

1. Si le joueur possède tous les territoires d'un continent, il reçoit 5 armées supplémentaires.
2. Le joueur peut passer cette phase si il le veut.
3. Si le défenseur perd toutes ses armées sur le territoire attaqué, l'attaquant déplace entre 1 et n-1 armées sur ce territoire. Avec n le nombre d'armées présentes sur le territoire attaquant. L'attaquant devient le possesseur de ce territoire. **extension 3.1** p 3.1. Si c'était le dernier territoire du défenseur, celui-ci est éliminé de la partie et l'attaquant récupère les cartes de sa main. Si l'attaquant vient à posséder 5 cartes ou plus grâce à cela, il retourne à l'étape 1.b. avec obligation d'échange de carte.

4. Si le joueur a capturé au moins un territoire lors de la phase d'attaque, il pioche la première carte de la pioche et la met dans sa main. **extension 4.1**

4.1. Si la pioche est vide, on mélange les cartes de la défausse et on les placent dans la pioche.

**Frequency:** Quelques dizaines de fois par partie pour chaque Joueur.

### Use Case Echange de carte

#### Revision History

Date	Author	Description of change

Use Case Template. Copyright (c) 2004-2005 TechnoSolutions Corporation

(Learn more about "TopTeam for Use Cases" at [www.technosolutions.com](http://www.technosolutions.com))

Permission is hereby granted, free of charge, to any person obtaining a copy of this document and its associated documentation, to use the document on their projects for any commercial or non-commercial purpose. However you may not publish, distribute, sublicense, and/or sell copies of this document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT. TECHNOSOLUTIONS CORPORATION MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS DOCUMENT FOR ANY PURPOSE.

**Use Case:** Echange de carte

**Id:** UC- 4

**Description** Le procédé par lequel un joueur échange des cartes contre des armées

**Level:** User Goal

**Primary Actor** Joueur

**Supporting Actors** Jeu

**Stakeholders and Interests** Pioche Défausse Carte Territoire Continent Main

**Pre-Conditions**

C'est au joueur de jouer. La partie n'est pas terminée.

## Post Conditions

### Success end condition

Le joueur a pu échanger certaines de ses cartes contre des armées et les a placées sur ses territoires.

### Failure end condition:

Le joueur ne réalise aucune action.

### Minimal Guarantee

Le joueur ne peut que faire des échanges valides.

## Trigger

Le joueur est en phase de renfort lors de son tour.

## Main Success Scenario

1. Le joueur peut échanger des jeux de 3 cartes de la forme: **extension 1**
  - a. 1 carte de chaque type: Fantassin, Cavalerie, Char
  - b. 3 cartes du même type
  - c. 2 cartes quelconques et une carte de type "Wild"
2. Pour chaque jeu échangé, il gagne un nombre d'armées correspondant au nombre total de jeu de carte échangés depuis le début de la partie par tous les joueurs:
  - a. Premier échange: 4 armées
  - b. 2: 6
  - c. 3: 8
  - d. 4: 10
  - e. 5: 12
  - f. 6: 15
  - g. 7: 20
  - h. n: de 5 en 5...
3. Une fois les cartes jouées, elles partent dans la défausse.

## Extensions

1. Si le joueur a 5 cartes ou plus, il est obligé de faire un échange.

**Frequency:** Une fois par tour pour chaque Joueur. Quelques dizaines de fois par partie pour chaque Joueur.

# Modèle de classes du domaine

## Diagramme de classe du domaine

[DomainClassDiagram] | *DomainClassDiagram.png*

## Diagramme d'objet du domaine

[ObjectDomainClassDiagram] | *ObjectDomainClassDiagram.png*

## Action attaquer - Snapshot

[ActionAttaquerSnapshot] | *ActionAttaquerSnapshot.png*

## Action échanger carte - Snapshot

[ActionEchangerSnapshot] | *ActionEchangerSnapshot.png*

## Invariants

```
context Etudiant::age() : Integer
post correct: result = (today - naissance).years()

context Typename::operationName(param1: type1, ...): Type
post: result = ...

context Typename::operationName(param1: type1, ...): Type
post resultOk: result = ...
```

# Dictionnaire de données

Term	Short Description	Meaning
Port	a	ax
Publication	Activity	Registers a Service with a DNS Responder.
Registration		See <a href="#">Publication</a>
Package		Minimum size = 9,000 bytes
One-Shot Multicast DNS Queries	bla	bla
Continuous Multicast DNS Querying	bla	bla
Service Instance Name	a	a
Service Type	bla	bla

## Conclusion

# Requirements Specification

## Introduction

### Purpose

The purpose of this document is to describe the requirement specifications for the project «Risk» for software engineering students.

The intended audience of this specification includes the prospective developers of the tool, as well as the technical assessment personnel.

### Document Conventions

None so far.

### Intended Audience and Reading Suggestions

### Project Scope

The software system to be produced is a simplified version of the Hearthstone online game, which will be referred to as «Risk» thorough this document.

The Risk system will allow players from different locations to confront each-other in short and intensive games.

### References

1. IEEE Standard 830-1993: IEEE Recommended Practice for Software Requirements Specifications

### Overview

The rest of this document contains an overall description of the Risk software system (section [Overall Description](#)), the specific functional requirements (section [System Features](#)), and the non-functional requirements for the system (see [Other Nonfunctional Requirements](#)).

## Overall Description

### Product Perspective

Hearthstone is a card game where two players confront each-other. The Risk software should allow players that are connected to the Internet to use their connected devices to play. Thus, «Risk» is an online, electronic version of the card game.

While the system is distributed and organized in different components, players should perceive it as a single piece of software. Figure [UML Deployment Diagram](#) presents the overall architecture of the software. Players interact with a Web Client, which uses the HTTP protocol to communicate

with (at most) one Game Server. Servers use TCP/IP to communicate with a Database Management Server, which stores all software data.

```
Dot Executable: /opt/local/bin/dot
File does not exist
Cannot find Graphviz. You should try

@startuml
testdot
@enduml

or

java -jar plantuml.jar -testdot
```

*UML Deployment Diagram*

## Product Functions

The Risk software must provide two main functions:

1. Game creation: allowing two players to discover each other and start a game.
2. Game play: allowing players to actually play Risk until the victory of one of them.

## User Classes and Characteristics

The Risk software has only one class of user: players. Players may have different levels: beginners, intermediate, or expert. However, independently from their level, players should use the same user interface to play against each other.

## Operating Environment

The Risk software should operate on any popular and recent operating system: Linux, Windows, or MacOS. The Web Client should operate on any recent web browser: Firefox, Chrome, Safari, or Edge.

## Design and Implementation Constraints

1. The Game Server must be developed in Java (version 1.8), using the [Spring Framework](#).
2. The Client must be developed in TypeScript (version 3.1), using the [Angular Framework](#).
3. All software artifacts must use a building tool: Maven or Groovy for Java, npm for TypeScript.
4. Dynamic tests must use JUnit (version  $\geq 5.0$ ) and Jasmine (version  $\geq 3.5.0$ ).
5. The code must log its main operations using [SLF4J](#).

## Verification Constraints

1. Test Doubles must be used to test each component independently.
2. Each unit test must describe its intention.



## User Documentation

No user documentation is required for the first version of the software.

## Assumptions and Dependencies

None until now.

# External Interface Requirements

## User Interfaces

## Hardware Interfaces

The software does not interact directly with any hardware device.

## Software Interfaces}

The client part of the software must operate on web browsers, whereas the server part must interact with a database through the Java Persistence API (JPA).

## Communications Interfaces

Communications between the client and the game server must use Websockets.

# System Features

## Game initialization

The Risk software must allow the setup of a game with two players and automatically prepare and distribute cards.

## Description and Priority

See Chapter [\[domain\]](#) (domain analysis) for further details.

## Stimulus/Response Sequences

## Functional Requirements

## Game play

The Risk software must allow two players to play against each other until a winner is settled. See Chapter [\[domain\]](#) (domain analysis) for further details.

# Other Nonfunctional Requirements

## Performance Requirements}

1. The game must be *playable*, meaning that users must have fast feedback for their actions and delays due to communications/connection problems must be correctly held.
2. The Web Client must be able to execute on a personal computer with 4GB of RAM.

## Safety Requirements

## Security Requirements

## Software Quality Attributes

### Extensibility

The software must be extensible, it must be easy for developers to add new cards and heroes to the game.

### Maintainability

1. The software must be readable and easy to maintain.
2. The Java source must respect Google's guidelines: <https://google-styleguide.googlecode.com/svn/trunk/javaguide.html>

## Business Rules

# Other Requirements

## Appendix A: Glossary

## Appendix B: Analysis Models

See Chapter [\[domain\]](#) (domain analysis) for further details.

## Appendix C: To Be Determined List

# Spécification des composants

Ce livrable correspond à la "conception préliminaire" du projet. Il comprend la division de la solution en différents composants, l'explication des fonctionnalités attendues de chaque composant, la spécification des interfaces fournies par chaque composant, ainsi que des diagramme de séquence qui valident ces interfaces.

## Objectif

Préciser les objectifs de ce chapitre.

## Organisation du chapitre

Cette section décrit le contenu du reste du chapitre et explique comment le document est organisé.

## Description des composants

Établir les frontières du système.

Division du système en composants.

Décrire le comportement souhaité des composants.

## Le composant **Game Server**

Décrire succinctement le comportement du composant.

### Spécification des interfaces

#### Spécification de l'interface A

Présentation de l'interface en UML (ou HUTN). Description du comportement de chaque opération.  
Spécification éventuelle des pré-conditions en OCL.

#### Spécification de l'interface B

## Le composant **Web Client**

Décrire succinctement le comportement du composant.

### Spécification des interfaces

#### Spécification de l'interface A

Présentation de l'interface en UML (ou HUTN). Description du comportement de chaque opération.  
Spécification éventuelle des pré-conditions en OCL.

# Interactions

Objectif: décrire, à haut-niveau, la collaboration entre les composants majeurs, en faisant référence aux besoins.

Utiliser des interactions, c'est à dire, des diagrammes de séquence et des diagrammes de communication.

- Ne vous limitez pas à une seule interaction par cas d'utilisation

## Mise en place d'un jeu

**Interaction: cas nominal**

**Interaction: cas A**

**Interaction: cas B**

## Tour d'un joueur

**Interaction: cas nominal**

**Interaction: cas A**

**Interaction: cas B**

# **Conception détaillée**

**Le composant WebServer**

**Le composant WebClient**

# Conclusion

Points forts de votre projet

Limites (points faibles) de votre projet