## horizontal line



SCORE Project: DRisk  
A Dynamically Configurable Risk Game  
- Requirements Document -

10.25.2015

**M**eng-Jung Lin  
**A**nant Srivastava  
**S**hujian Ke  
**C**haoyi Fu  
**S**atyam Jaiswal

**Table of Content**

[Executive Summary](#h.au51mny0sx6)

[Purpose](#h.jne3kfijaa3q)

[Project Requirements](#h.rs1fxx4zy2zz)

[Functionality Requirements](#h.stkxeg8x1sbv)

[Usability Requirements](#h.gi64wgrytlet)

[Reliability Requirements](#h.azba5bz74g9q)

[Performance Requirements](#h.yl8qaa578f20)

[Supportability](#h.1dmpsv8wjasg)

[Constraints](#h.j5x2nqw0dht1)

[Use Cases](#h.tqkehqbm92tc)

[Project Schedule](#h.cj69ac1er7v1)

[Gantt Chart](#h.edovvi96pjpk)

[Tasks & Milestones](#h.1nikza7o7q2)

# Executive Summary

*SCORE Project: DRisk - A Dynamically Configurable Risk Game -* (DRisk) is a project built by Team MASCS, a group of students from Arizona State University. DRisk is one of the project topics of the 2016 SCORE Contest, which is a student contest on software engineering held every year. The DRisk project is sponsored by Nazareno Aguirre.

Risk is a popular strategy board game designed and created in the late 1950s. Risk is a turn-based board game that allows two to six players to fight over conquering forty two territories. Because of its popularity, and with the technology growing at a rapid pace in the recents years, there have been several PC version of Risk game release to the public, including web versions.

Risk is a strategy game. There are many rules set in stone in the manual. With the current official set up, the playing time of the Risk game can be very lengthy, and can be challenging for some first-time players. There are a couple of simple versions of the risk game, but the rules and game play are still immutable.

The ultimate goal of this project is to create a dynamic Risk game (D stands for Dynamic in DRisk). Team MASCS has created a web-based dynamic Risk game that allows the users to configure some variables within the game. With this setting, we are giving the user more control on how the games is setup and on the playing time of the game, we also think that this can help the first-time players learn the game in a simpler version before going on to a complex full version of the game.

# Purpose

The purpose of this software requirements document is to describe the client-view and the developer-view for the DRisk project. The client-oriented requirements describe the project from the client’s perspective, which are mostly listed under functional requirements. These requirements will explain how the user use the product and what they can see from the outside. The developer-oriented requirements describe the project from the developer’s perspective, which includes the detailed description of functional, data, performance, and supportability. The requirements documentation gives a broad view of how the final product will look like.

# Project Requirements

Since DRisk is a contest project topic, there are some set requirements that the team has to follow when building the project.

Original project description and requirements can be found on the official SCORE project website (<http://score-contest.org/2016/projects/drisk.php>)

Note that ID is subject to change as the lists grow

## Functionality Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **FEATURE SET** | F01 | 1 | The project implements all core functionalities of the standard version of Risk game. |
| F02 | 1 | The project is a Risk game to be played through a web site. |
| F03 | 1 | The game allows 2-6 player to play. |
| F04 | 1 | The game has a preloaded, 42-territory map. |
| F05 | 1 | The game allows user to define and upload map configuration: image (aesthetic aspects) |
| F06 | 1 | The game allows user to define and upload map configuration: number of territories |
| F07 | 1 | The game allows user to define and upload map configuration: grouping into continents |
| F08 | 1 | The game allows user to define and upload map configuration: neighbor relations |
| F09 | 3 | The game allows user to define and upload map configuration: number of players |
| F10 | 1 | The game allows for maps to automatically adjust granularity by collapsing neighboring territories according to number of players |
| F11 | 1 | The game allows the user to select complexity of the game (easy, medium, hard) |
| **Security** | F12 | 1 | User will not be able to access the server |

## Usability Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **HUMAN FACTORS** | U01 | 1 | The game runs on Internet Explorer. |
| U02 | 1 | The game runs on Chrome. |
| U03 | 1 | The game runs on Firefox. |
| U04 | 5 | User can access the mobile version of the webpage through mobile devices. |
| **AESTHETIC** | U05 | 3 | The game contains an elegant interface to make it easy for user to navigate around. |
| **CONSIS- TENCY** | U06 | 3 | Information of each new game will be stored in the database in the same format |
| U07 | 3 | All code follows the same coding standards for easy understanding and maintenance. |

## Reliability Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **PREDICTA-BILITY** | R01 | 2 | Server will restart itself if it crashes. |
| R02 | 1 | Entering a URl will take the user to the correct page and load the correct game instance. |
| R03 | 2 | The web site work the same way in IE8, Chrome 46, and Firefox 41 |
| **ACCU- RACY** | R04 | 4 | HTTPS will be supported for users who wish to use it. |
| R05 | 1 | Information for each game is accurate. |
| **RECOVER-ABILITY** | R06 | 2 | When user enters a wrong URL or click on a broken link, a 404 page is displayed. |

## Performance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **RESOURCE CONSUMPTION** | P01 | 2 | Database storage does not exceed 30 GB. |
| P02 | 2 | Website handles 1,000 games at the same time. |
| P03 | 4 | Server downtime does not exceed an hour per week. |
| **RESPONSE TIME** | P04 | 4 | Game page loaded within 5 seconds after starting new game or join existing game. |
| P05 | 1 | Web pages loaded under 10 seconds even under slow connection |

## Supportability

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **TESTA- BILITY** | S01 | 2 | Controllability: Possible to control the state of component under test as required for testing. |
| S02 | 2 | Observability: Testers are able to observe intermediate and final result. |
| S03 | 4 | Understandability: Component under test are documented or self-explaining. |
| **EXTENCI- BILITY** | S04 | 3 | New features are easily implemented without breaking or changing other parts of the program |
| **MAINTAIN-ABILITY** | S05 | 3 | Future maintenance is easy. |
| **EASE OF INSTAL** | S06 | 2 | User is able to access the website using any browser |

## Constraints

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **ID** | **PRIORITY** | **DESCRIPTION** |
| **DESIGN** | C01 | 2 | Object-oriented design/language is used. |
| C02 | 3 | Information displayed on the website is handled by OO database model. |
| C03 | 4 | Development period is one semester |
| **IMPLEMENTATION** | C04 | 1 | Use PHP and Java as web programming languages |
| C05 | 1 | Use Amazon AWS as the server |
| C06 | 2 | The database is MySQL |
| **INTER- FACE** | C07 | 3 | The website is fully functional using IE, Chrome, and Firefox. |
| **PHYSICAL** | C08 | 2 | The website is accessible using PCs, smartphones, and tablets. |

# Use Cases

|  |  |
| --- | --- |
| **Use Case: Creates New Game** | |
| **Identifier:** | UC01 |
| **Description:** | The case models a user creating a new game. |
| **Actors:** | User |
| **Precondition:** | User is on the home page. |
| **Flow of Event:** | 1. The user clicks on the “New Game” button on the home page. |
| **Postcondition:** | System displays the configuration page. |

|  |  |
| --- | --- |
| **Use Case: Set Game Configuration** | |
| **Identifier:** | UC02 |
| **Description:** | The case models a user setting the configuration of the game. |
| **Actors:** | User |
| **Precondition:** | User is on the configuration page. |
| **Flow of Event:** | 1. Default values are loaded onto the configuration page.  2. The user changes the value as desired.  3. The user clicks the “Start Game” button. |
| **Postcondition:** | System displays the game page with the configurations from the configuration page. |

|  |  |
| --- | --- |
| **Use Case: Join Existing Game** | |
| **Identifier:** | UC03 |
| **Description:** | The case models a user joining an existing game. |
| **Actors:** | User |
| **Precondition:** | User is on the home page. |
| **Flow of Event:** | 1. The user click on the “Join Game” button on the home page. |
| **Postcondition:** | System displays the joinGame page. |

|  |  |
| --- | --- |
| **Use Case: Join Existing Game Room I** | |
| **Identifier:** | UC04 |
| **Description:** | The case models a user joining an existing game. |
| **Actors:** | User |
| **Precondition:** | User is on the joinGame page. |
| **Flow of Event:** | 1. A list of current existing game will be shown.  2. The user selects a game room, the game entry in the list will be highlighted.  3. The user hits the “Join” button. |
| **Postcondition:** | System displays the game page of the selected room. |

|  |  |
| --- | --- |
| **Use Case: Join Existing Game Room II** | |
| **Identifier:** | UC05 |
| **Description:** | The case models a user joining an existing game. |
| **Actors:** | User |
| **Precondition:** | User is on the joinGame page. |
| **Flow of Event:** | 1. The user enters the room number into the textbox.  2. The user hits the “Join” button. |
| **Postcondition:** | System displays the game page of the room with the specific room number entered. |

|  |  |
| --- | --- |
| **Use Case: Starting The Game** | |
| **Identifier:** | UC06 |
| **Description:** | The case models a user starting the game play. |
| **Actors:** | User |
| **Precondition:** | User is on the game page. |
| **Flow of Event:** | 1. The user clicks the “Start” button. |
| **Postcondition:** | System starts the game play. |

|  |  |
| --- | --- |
| **Use Case: Deploying Army** | |
| **Identifier:** | UC07 |
| **Description:** | The case models a user deploying troops onto the map. |
| **Actors:** | User |
| **Precondition:** | User is on the game page. |
| **Flow of Event:** | 1. The user clicks the “Start” button.  2. The user clicks a territory. |
| **Postcondition:** | The page shows that the territory is marked with the player’s color. |

|  |  |
| --- | --- |
| **Use Case: Attacking Neighbor Territory** | |
| **Identifier:** | UC08 |
| **Description:** | The case models a user attacking a neighbor territory. |
| **Actors:** | User |
| **Precondition:** | All players have completed the initial deploying.  The player has a territory with more than one troop. |
| **Flow of Event:** | 1. The player clicks on his territory that has more than one troop.  2. The player clicks an opponent territory to attack. |
| **Postcondition:** | The page shows the result of the combat, whether the player wins and advance to the opponent’s territory, or the opponent wins and the territory remains the same. |

|  |  |
| --- | --- |
| **Use Case: Fortifying Troops** | |
| **Identifier:** | UC09 |
| **Description:** | The case models a user moving his troop from one territory to another territory. |
| **Actors:** | User |
| **Precondition:** | All players have completed the initial deploying.  The player has completed the attack step of his turn. |
| **Flow of Event:** | 1. The player clicks on his territory that has more than one troop.  2. The player clicks on another of his own territory. |
| **Postcondition:** | The page shows that 1 troop is deducted from the original territory and 1 troop is added to the other territory. |