User Requirement Specification

2016



Mikaeil Shaghelani Lor – jiefan lin-Wen Li | ei8s2

Group 3

Date: 28/Apr/2016

version: 0.0.1

Contents

[Introduction 2](#_Toc449614719)

[Functional Requirements 2](#_Toc449614720)

[-Use Case diagram 2](#_Toc449614721)

[-UseCases 3](#_Toc449614722)

[1001-Register 3](#_Toc449614723)

[1002-Login 3](#_Toc449614724)

[Enter specifications 4](#_Toc449614725)

[Run simulation 4](#_Toc449614726)

[Post: Pop-up results visible on screen 4](#_Toc449614727)

[Save as 5](#_Toc449614728)

[Save 5](#_Toc449614729)

[Save simulation results 6](#_Toc449614730)

[Open 6](#_Toc449614731)

[Reset 6](#_Toc449614732)

[Stop simulation 7](#_Toc449614733)

[Exit Application 7](#_Toc449614734)

[About 7](#_Toc449614735)

[User Manual 8](#_Toc449614736)

[New 8](#_Toc449614737)

[Draw grid 8](#_Toc449614738)

[Enter time and location information 9](#_Toc449614739)

[User Interface 10](#_Toc449614740)

[Non-functional Requirements 11](#_Toc449614741)

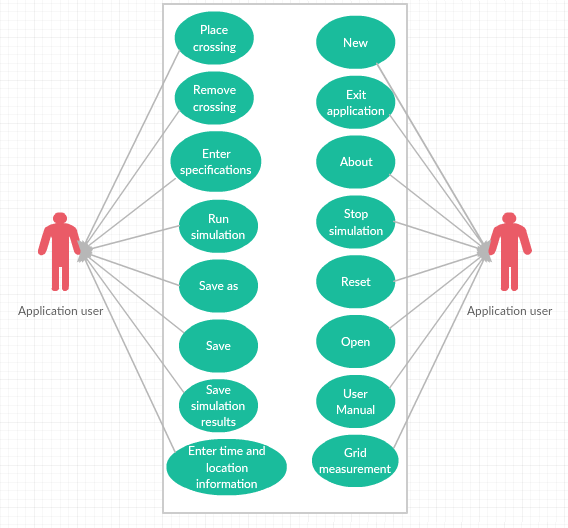
# Introduction

Assignment: Building a Contributed Game

We are going to develop a contributed application (game) that make use of Microsoft WCF feature to make server and client part. Our project is to make a game called “2D Heroes of the storm”, that is a combat form game. Players choose a hero that has health and mana (requirement for casting an attack) they fight each other till one of them die. Users can register and login trough the client app and can make their personal page that holds user information and game’s history such as win, loss, game replay and some statistics. In home page users are provided with a list of online players and a list of current trending games. Users have single and multiplayer games, they can communicate through a group chat and send game request to each other or start to watch a trending game. On single player mode user need to choose difficulty level also.

# Functional Requirements

## -Use Case diagram



## -UseCases

## 1001-Register

Actor: The User

Pre: The software is running

MSS:

1. User click on register option.
2. System shows the registration form.
3. User fill in the form and press on submit.
4. System saves user’s data.
5. System shows successful register message.

Ex:

* 1. Actor entered incorrect data.
  2. System informs actor of the missing data and go to step 2.

Post: user is registered.

## 1002-Login

Actor: The User

Pre: The software is running

MSS:

1. Actor click on login option.
2. System shows the login form.
3. Actor fill in the username and password and click on login.
4. System log in the user and redirect to the main page.
5. System shows successful login message.

Ex:

3.1 Actor entered incorrect data

3.2 System inform shows login error message and go to step 2.

Post: User is logged in and redirected to main page.

=====\*\*\* from here need to be changed\*\*\*====

## Edit Profile

Actor: The person who is using the software

Pre: There is at least one crossing on grid

MSS:

1. Actor chooses crossing layer from Crossing Panel
2. Actor specifies required crossing information for every available feeder (see figure 2.b)
3. Actor clicks on apply button
4. System saves values in the model
5. System shows success message

Ex:Actor didn’t specify all required information

* 1. System informs actor of the error in status section

Post: Success message is visible in status section

## Watch replay

Actor: The person who is using the software

Pre: The software is running and there is an open model

MSS:

1. Actor clicks on Start from the simulation box
2. System checks if all input is complete
3. System starts the simulation
4. System shows results at the end in a pop up form

Ex:

* 1. Actor didn’t input all required information

2.2 System informs actor of error in the status section and system goes stops the use case

## Post: Pop-up results visible on screen

## Single player

Actor: Person who is using the software

Pre: System is running

MSS:

1. Actor clicks on save as
2. System shows save as dialog
3. Actor choose the directory and inputs the file name
4. Actor clicks on save
5. System saves the project into a file.

Ex:

1. File name already exists

System informs actor

* 1. If coming from use case open system goes to use case open step 2
  2. If coming from use case exit application system goes to use case exit application step 2
  3. If coming from use case new system goes to use case new step 2

Post: Current project is visible on screen

## Multiplayer

Actor: The person who is using the program

Pre: Software is running

MSS:

1. The actor clicks on save from the file tab
2. System checks if project has been saved before.
3. System saves the project

Ex:

2.1 If the project hasn’t been saved before

2.2 System goes to use case save as step 2

3.1 If coming from use case open system goes to use case open step 2

3.2 If coming from use case exit application system goes to use case exit application step 2

3.3 If coming from use case new system goes to use case new step 2

Post: Current project is visible on screen

## Exit Application

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on Exit from File tab
2. System closes the application.

Ex:

* 1. If project is modified system goes to save use case step 2

Post: Application is not visible on screen anymore

## About

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on About in Help tab
2. System opens a pop-up page.

Post: Pop-up page is visible on screen

## User Manual

Actor: The person who is using the software

Pre: The Software should be running

MSS:

1. Actor clicks on the User’s Manual in Help tab
2. System opens the Digital User’s Manual.

Post: User’s manual is visible on screen

# User Interface

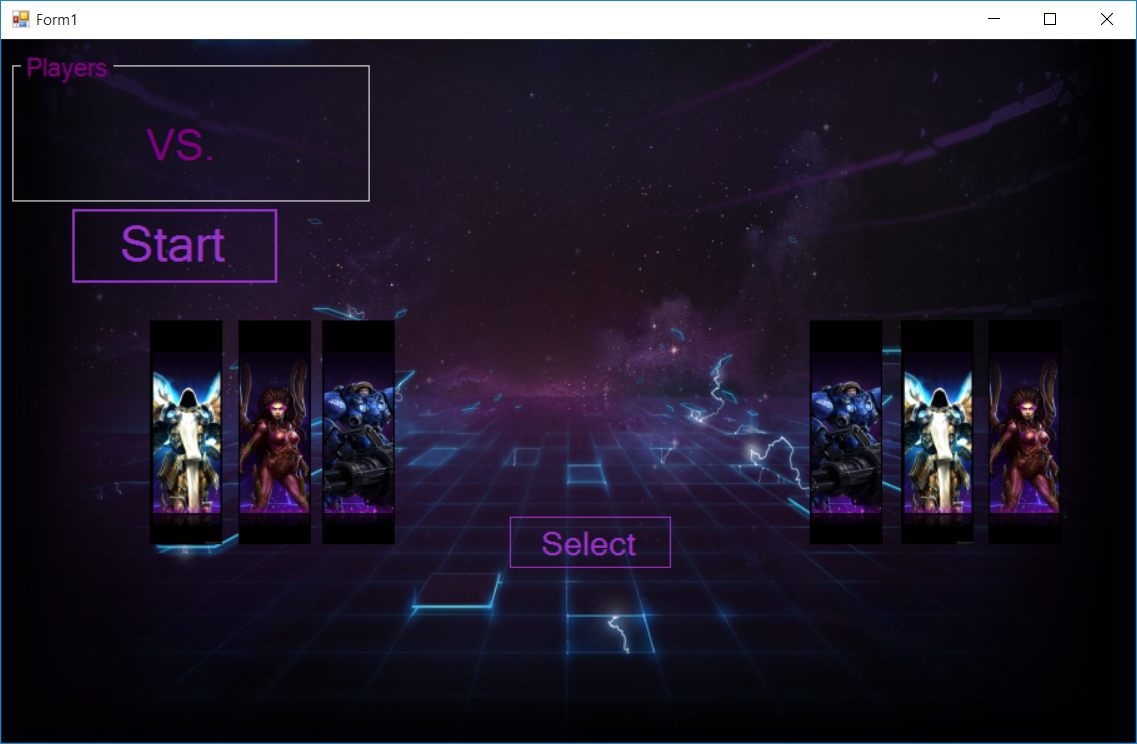


Figure 2.a

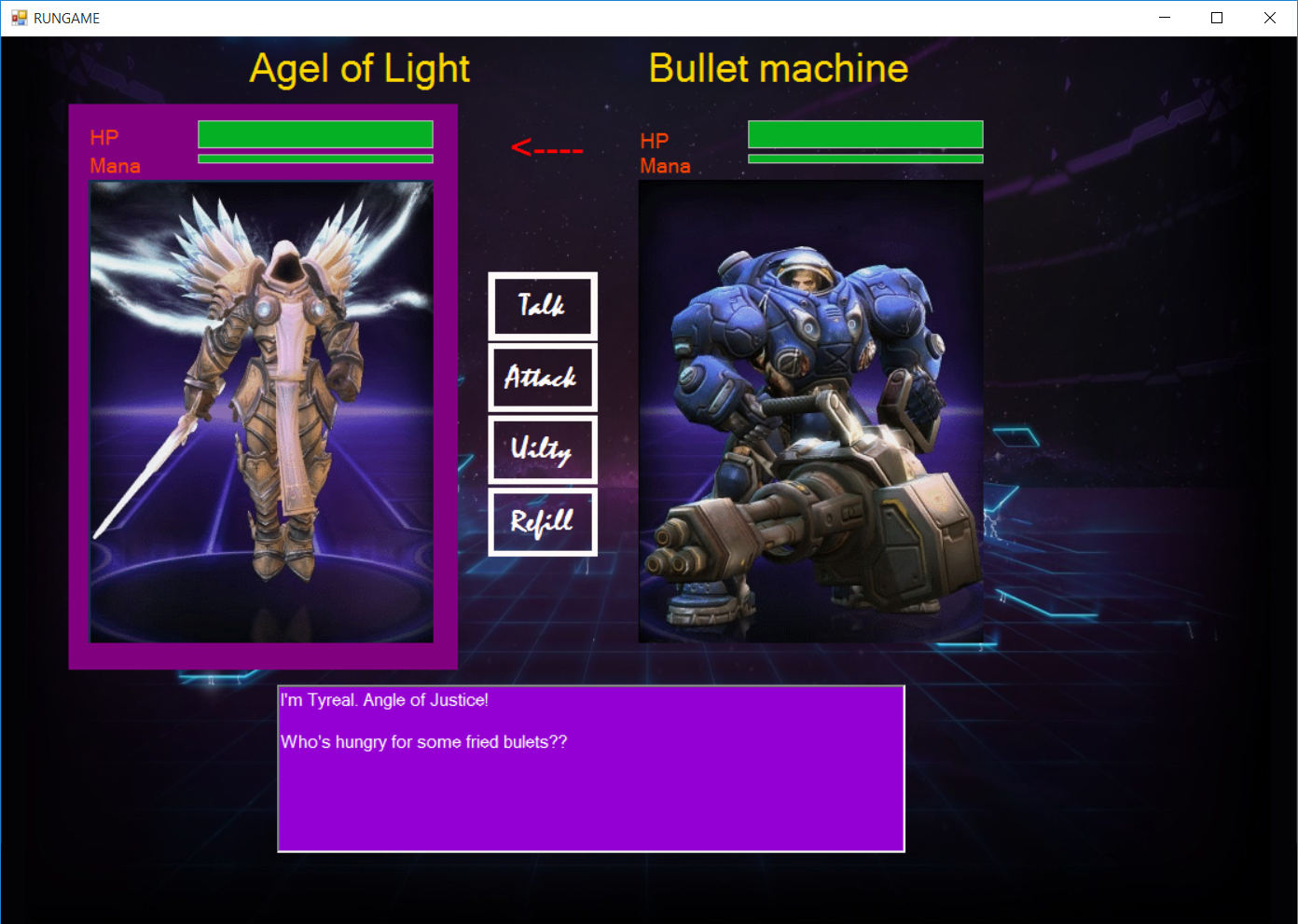


Figure 2.b



# Non-functional Requirements

* This application will run on windows machines.
* Crossing’s positions: users can create crossing by dragging a crossing on the grid work space inside a grid cell.
* Completed Crossing simulation can be saved.
* Project can be saved.
* The user manual use case is a could requirement. All other use cases are must requirements.
* Every grid cell must have a crossing before running simulation.
* Traffic light time input must be in seconds.
* Car flow percentage must be whole numbers.
* A new crossing cannot be placed on a full grid.
* A maximum of 12 crossings.