Test Plan

Potkuri-group

Helsinki December 12, 2008

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (6 cr)

Project Group

Veera Hoppula
Mikko Kuusinen
Jesse Paakkari
Tobias Rask
Timo Tonteri
Eero Vehmanen

Client

Valentin Polishchuk

Project Masters

Sampo Lehtinen

Homepage

`http://www.cs.helsinki.fi/group/potkuri`

Change Log

Version	Date	Modifications
1.0	13.11.2008	1st draft

Contents

1 Introduction

1.1 Document Purpose

The purpose of this document is to describe the different tests the group Potkuri will have to perform to ensure that our software will work with the least bugs possible.

The only important section in this document is the System testing. Unit tests and integration tests are rather well commented, and the descriptions can be found from program's javadoc.

2 System testing

2.1 Test Plan

System testing purpose is to assure that software corresponds it's requirements. Every member of group must attend to System test- event. Then group members test all use cases and go through non functional requirements, quality requirements and confines. Group will then write test-report, which reveal tested requirements and results.

Functional requirements are tested through user interface.

2.2 Test Design Specification

Requirement document's non-functional requirements, quality requirements and confines are validated or justify, if requirement can't be validate.

2.3 Test Case Specification

2.3.1 Buttons

Program should terminate when pressing Close-button or X-button - Ok.

Program should pause when pressing Pause-button - Ok.

Program should continue running when pressing Run-button - Ok.

2.3.2 Parameters

Program should take settings from ini-file, if it is defined in parameters - Ok.

Program should override only settings which are defined in ini-file - Ok.

Program should use default settings, if ini file doesn't exists - Ok.

2.3.3 User Requirements

Plane Planes don't slow down speed when approaching airport. This was optional and ok. Planes slow down their speed when they are too close each others, but this is not sufficient condition to prevent plane crashes. This problem may be solved in future by control planes speed on the grounds of arriving time to merge point.

Weather This section is Ok.

Map This section is Ok.

Airport This section is Ok.

Algorithm Algorithm section is ok, but group Potkuri put forward that the tree should be re-calculated regulary when the time passed by.

Arrival tree This section is Ok.

Merge points This section is Ok.

General This section is ok. Group accentuate that the program doesn't calculate the safest path but rather the optimal path with defined safety distance.

2.4 Functional Requirements

F1 - Planes have a location at certain time on map Status: Ok

F2 - Planes never fly over intense storm In some situations plane may fly over storm. Basically there are two reasons: user has set too small value for StormSafetyDistance - parameter, or the strom suddenly appear from nowhere. The main point is that the current arrival tree never perch over the storm.

F3 - Arrival tree's branches never cross Status: Ok

F4 - Weather data is acquired from file Status: Ok.

F5 - The program shows visually planes, arrival tree and storms Status: Ok

F6 - Parameters for program are in text file Status: Ok

F7 - User can create weather conditions Status: Ok

F8 - Map can be zoomed Status: This feature is not implemented.

F9 - User is able to give amount and arriving place of the planes Status: OK

F10 - Wind has direction and speed Status: This feature is not implemented.

F11 - Weather data is generated randomly Status: This feature is not implemented.

F12 - User is able to give storm centers, their intensity and wind speed Status: This feature is not implemented.

F13 - Weather data is acquired from Testbed Status: Ok

F14 - Program stores every weather data picture from Testbed to hard drive Status: This feature is not implemented.

2.4.1 Non-Functional Requirements

N1 - Program works Status: Ok. Refresh rate can be adjust so that calculating frame is smooth.

N2 - Program is highly visual Status: Ok.

2.5 Environment

E1 - Program runs on relatively modern laptop computer Status: Ok. This feature is tested on Windows, Linux and Mac Os platforms.

2.6 Test Summary Report

System testing date: 28.11.2008 10.15 Place: The Department of Computer Science at the University of Helsinki.