

User manual

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	12.12.2008	Final version.
0.5	11.12.2008	Added information.
0.4	9.12.2008	Parameters part fixed.
0.3	4.12.2008	Parameters part written.
0.2	3.12.2008	Introduction part written.
0.1	3.12.2008	Document created.

Contents

1	Introduction	1
2	Vocabulary	2
3	Running instructions	3
3.1	Screen Layout and Using the Menu	4
4	Parameters	7
4.1	User defined parameters	7
4.2	Administrative parameters	8
5	Using self-made PGM images	9
5.1	Sample ini File:	9

1 Introduction

This is a users manual for a program called Chopper. It is a path planning system, used to guide airplanes safely to their destination. The program displays an arrival tree, with airplanes following it to airport. The main purpose being, to find out, how many planes can be routed to the airport in some period of time.

2 Vocabulary

Airport Airport is where arrival tree begins, in the middle of the map.

Arc Arcs are circles at a determined radius distance of the airport. The merge points are located into these arcs.

Arrival tree A binary tree consisting of paths. Has a root at the airport.

Checkstyle Java code review for Eclipse.

dbZ dBZ stands for decibels of Z. It is a meteorological measure of equivalent reflectivity (Z) of a radar signal reflected off a remote object.

EclEmma Java Code Coverage for Eclipse.

Flight plan Every plane has a flight plan which describes its path.

FMI Finnish Meteorological Institute.

Integration Testing Integration testing purpose is to assure that integrated classes do all those services they are planned to do in requirement document.

Java2D Display and print 2D graphics in Java programs.

JAR Runnable Java archive, which based on the ZIP file format.

JUnit JUnit testing framework.

Map A map from somewhere in the world used in this product.

Merge point A point on the map where two paths merge into one path.

nmi nautical mile (=1,8520km)

Path A route to the airport that should avoid storms.

PGM Portable Gray Map, a graphics file format.

Plane An airplane that tries to land at an airport along a path avoiding storms.

PMD Java code review for Eclipse.

Storm A set of pixels with a dBZ-value over a certain threshold (that is a parameter) close each other on the map. Indicated with red color on the map.

System testing System testing purpose is to assure that software corresponds it's requirements.

User A person using the product to watch animations on aircrafts landing at an airport in presence of hazardous weather systems.

Unit testing Unit testing purpose is to assure that certain class or unit do all those services it is planned to do in requirement document.

3 Running instructions

The program can be run using a `.jar` file. This file is found from the link "The program" of the index (also `index/install/Chopper.zip`), in a `zip` -package, which should be unzipped first. In an unzipped folder the running is done in unix environment with a command `java -jar Chopper.jar`. A window, with a radar like screen will be opened. Some parameters can be used by using an initialization file. Initialization file must be specified as a command-line argument when running the program eg. `java -jar Chopper.jar samplePGM.ini`.

If no command-line arguments are specified, the default settings will be used. The defaults setting will display weather storm pattern over the greater Helsinki area on the 14th August 2008. The default data is located in `data/images`-directory.

3.1 Screen Layout and Using the Menu

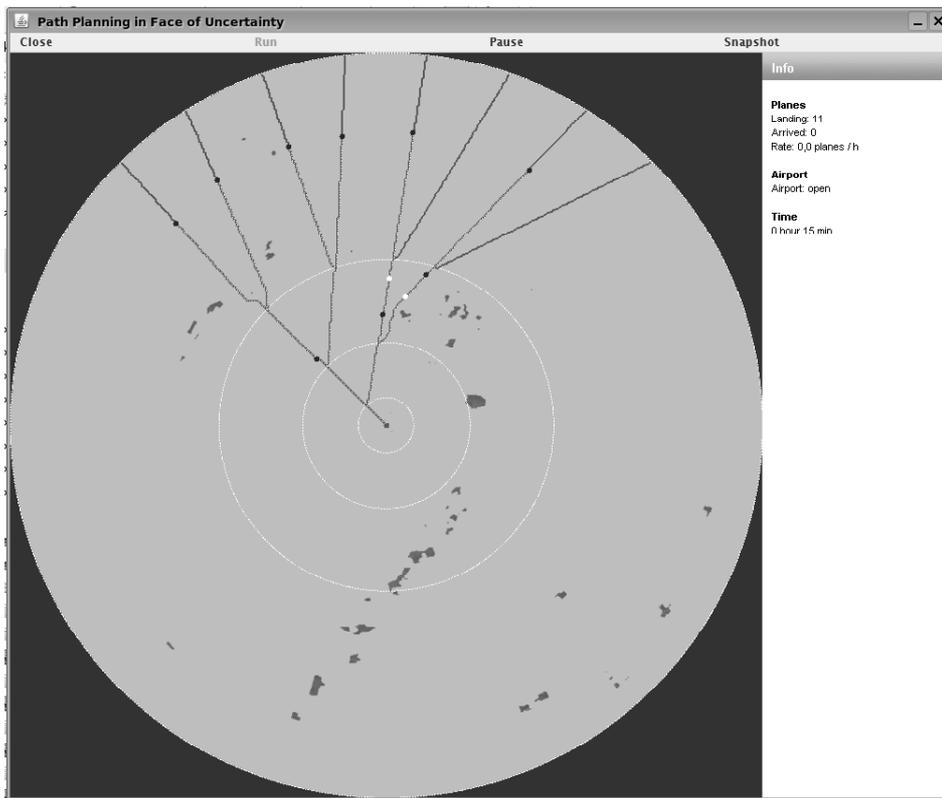


Table 1: Description of the menubar items

Item	Explanation
Close	Ends the execution of the program and closes the screen.
Run	Execution of the program continues after clicking the 'Run'.
Pause	Execution of the program can be paused by clicking the 'Pause'.
Snapshot	Takes the jpg file format snapshot of the screen and saves the file in DataDir-directory naming it with the timestamp.

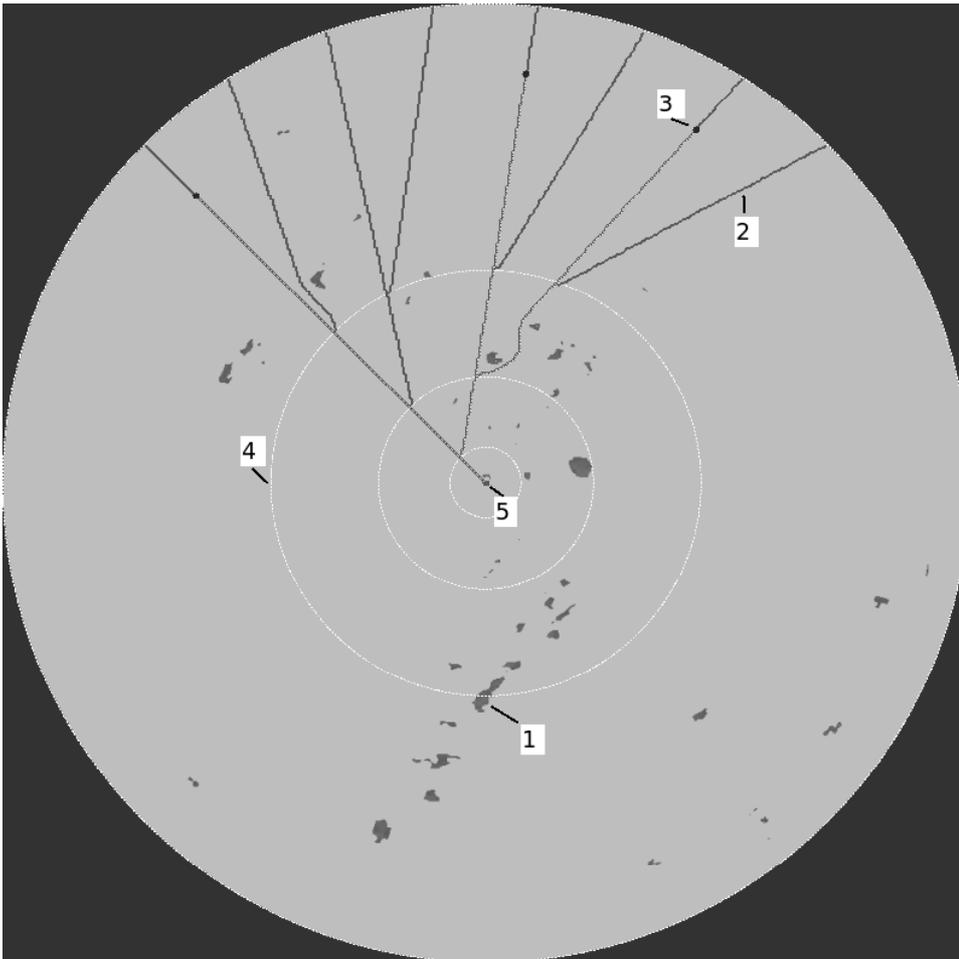


Table 2: Description of the Picture Elements

Nr	Explanation
1	Storm.
2	Arrival-tree branch.
3	Plane.
4	Arcs.
5	Airport.

Table 3: Description of the Info items on the right-hand side

Planes landing	Number of planes in the air approaching the airport.
Planes arrived	Number of planes landed since the start of the program.
Planes rate	How many planes have landed per hour.
Airport	Can the airport be reached or it closed).
Time	The time the program has been running in simulated time.

4 Parameters

This section describes all the parameters that can be used to modify the programs operation.

4.1 User defined parameters

Table 4: Users parameter definition table

Parameter	Default	Range	Description
SectorStartDeg	45	[0 - 360]	Arrival tree sector start angle.
SectorEndDeg	135	[0 - 360]	Arrival tree sector end angle.
NumberOfStartPoints	8	[0 - 8] ¹	Number of possible arrival points for planes.
MinSafeDistance	4	[1 - 5] ²	Minimum safety distance between planes.
StormSafetyDistance	5	[0 - 10] ²	Safety zone distance for storms (how many graph nodes).
WeatherDataType	radar	radar/PGM	Source image type.
WeatherDataFile	weatherset.dat	>1 images	Data file that contains a list of images.
WeatherMapWidth	1000	unlimited	Width of image file size. Only in case of radar files.
WeatherMapHeight	1000	unlimited	Height of image file size. Only in case of radar files.
DBZStormSensitivity	30	[15 - 50]	Limit for stormy weather.
DataDir	data/		Program data files path. Relative to main class.
PlaneCreateInterval	8	[1 - *]	Number of planes per time period.
PlaneMaximumSpeed	400	>minspeed	Nmi / h
PlaneMinimumSpeed	100	[1 - *]	Nmi / h
PlaneCreateMultiplier	2	[1 - 8] ¹	Number of planes to create per time period.
PlaneMinSafeDistance	20	unlimited	Safety distance for planes to each other.

¹This can only be at most the number of leaves in the arrival tree.

²The program will slow down too much if this is set higher.

4.2 Administrative parameters

Table 5: Administrative parameter definition table

Parameter	Default	Range	Description
ProgramVersion	1.0	unlimited	Defines program version.
ProgramDate	2008-10-28	unlimited	Defines last release date.
Demo	0	[0 - 1]	Sets demo mode on/off.
Debug	0	[0 - 1]	Sets debug mode on/off.
StartDelay	300ms	unlimited	Start delay for the program. At least default recommended to let the program to initialize the tree.
MediaDir	data/		Programs image source directory.
JarFile	0	[0 - 1]	Flag to set if run as jar file.
LocationCornerLatitude	0	[0 - 360] ³	Latitude of the left-down corner.
LocationCornerLongitude	0	[-90 - +90] ³	Longitude of the left-down corner.
TimerInterval	1000	>0	Program speed setting.
WeatherMapRenderHeight	800	unlimited	Height of rendered weath- ermap.
WeatherMapRenderWidth	800	unlimited	Width of rendered weath- ermap.
WeatherMapCornerBackgroundColor	50	[0 - 255]	Map corner colour.
WeatherMapRadarBackgroundColor	190	[0 - 255]	Map background colour.
NoiseFilterSensitivity	6	[0 - 6]	Filter for radar disturbance.
GraphWidth	500 ³		
GraphHeight	500 ³		
WindowWidth	1100	>1100	Program window width.
WindowHeight	800	>800	Program window height.
RenderAirspace	true	boolean	Draw planes.
RenderArrivalTree	true	boolean	Draw arrival tree.
RenderArcPoints	true	boolean	Draw arc points.
RenderPlanePath	true	boolean	Draw single plane paths.
RenderPlaneAsTriangle	true	boolean	Draw planes as triangles, or discs.
GraphicsEngineArcpointsColor	250	[0 - 255]	Color to use for arc points.
GraphicsEngineArrivalTreeColor	180	[0 - 255]	Color to use for arrival tree.
ColorNormal	0000FF ⁴	All colors	Airplane color in no traffic
ColorAlert	FFFFFF ⁵	All colors	Airplane color in traffic.

5 Using self-made PGM images

First you must make your own PGM(portable graymap image)-format files. You can use any image editor to make your own files. Make sure that the image is saved in raw-format, not in ascii-format, if given a choice.

Enter all the filenames you wish to use in one file and save it as eg. samplePGM.dat .One filename per line and do not include the path for directory, since the image directory path is specified by `DataDir`-parameter. All the lines having '#'-character at the start of the line are considered as comment-lines and ignored by the program. Enter the filename as a `WeatherDataFile`-parameter in the ini-file.

`WeatherDataType` must be set 'PGM'.

5.1 Sample ini File:

There is a sample ini-file and data included with the program. To use sample data, run the program with `SamplePGM.ini` as a command-line argument.

Contents of the samplePGM.ini file

```
WeatherDataType = PGM
WeatherDataFile = samplePGM.dat
DataDir = data/pgmimages/
SectorStartDeg = 0
SectorEndDeg = 360
NumberOfStartPoints = 45
```

From the above ini-file contents, we can read the following:

- Weather data images are in PGM format
- samplePGM.dat will include all the names of the user made image files.
- Data directory will be data/pgmimages and it is relative to the directory from where the program is run since there is no slash in the front of data/pgmimages/.

Contents of the samplePGM.dat file

```
JollyRoger-Maze
#pic08
pic07
pic02
pic03
pic04
pic05
pic06
```

³This value should not be changed.

⁴Blue color.

⁵White color.

pic01

All the other *.pgm files are used, except the pic08. The file names and extension is case-sensitive in unix-environment.