Project Plan
for
Triangulation Games

Version 1.1 approved

Prepared by:

Ville Parviainen,
Chienting Weng,
Jon Sahlberg,
Kyösti Karila,
Salvador Jesús Romero Castellano,
Umair Azfar Khan

<18.1.2006>
## Version History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason for Change</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ville Parviainen</td>
<td>10-10-2005</td>
<td>First inside Review</td>
<td>0.9</td>
</tr>
<tr>
<td>Umair Azfar Khan</td>
<td>19-12-2005</td>
<td>Final Review</td>
<td>1.0</td>
</tr>
<tr>
<td>Ville Parviainen</td>
<td>18-1-2006</td>
<td>WBS-changes</td>
<td>1.1</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Introduction ................................................................. 5  
   1.1. Purpose and Scope .................................................. 5  
   1.2. Product and Environment........................................ 5  
   1.3. Definitions, Acronyms and Abbreviations .................. 5  
   1.4. References ............................................................ 5  
   1.5 Overview ............................................................... 5  
2. Current System ............................................................. 6  
   2.1. Old System ............................................................ 6  
   2.2. Other Similar/Corresponding Systems ....................... 6  
3. Advantages and Disadvantages ...................................... 6  
   3.1 Advantages ............................................................ 6  
   3.2 Disadvantages ....................................................... 6  
4. Project Organization ..................................................... 7  
   4.1 People (project group) ............................................. 7  
   4.2 Related Organizations ............................................ 7  
   4.2.1 Inside Company .................................................. 8  
   4.2.2 Client ............................................................... 8  
   4.3 Changing Group Size ............................................. 8  
5. Project Goals and Ending/Termination ............................ 8  
   5.1. Goals of Project Group .......................................... 8  
   5.2. Goals of Client ..................................................... 8  
   5.3. Goals of the Project ............................................. 8  
   5.4. Termination Criteria ............................................ 8  
   5.5. Ending Criteria .................................................... 9  
6. Project Phases and Timing ............................................. 9  
   6.1. Methods and Tools ............................................... 9  
   6.2. Phases ............................................................... 9  
   6.3. Work and Task Estimates ...................................... 10  
   6.4. Deliverables ....................................................... 11  
7. Monitoring and Guidance ............................................. 11  
   7.1. Inside Group ....................................................... 11  
   7.2. Outside Group ..................................................... 11  
   7.2.1. Client ............................................................. 11  
   7.2.2. Course ........................................................... 12  
   7.3 Others ............................................................... 12  
8. Standards, Directives, Guidelines .................................. 12  
   8.1 Standards and Directives ....................................... 12  
   8.2 Confidentiality ..................................................... 13  
   8.3 IPR and Copyrights .............................................. 13  
9. Risk Management ....................................................... 13  
10. Learning and Studying Plan ........................................ 15  
   10.1. (To) Project Group .............................................. 15  
   10.1.1 Theoretical Knowledge ..................................... 15  
   10.1.2 Practical Knowledge ........................................ 15
10.2. (To) Client ................................................................................................................. 16
11. Installation Plan .............................................................................................................. 16
12. Starting (Initialization, Deployment) Plan ........................................................................ 17
13. Costs (Budget) ................................................................................................................ 17
14. Rejected Alternatives and New Ideas ........................................................................... 18
  14.1 Rejected Alternatives ............................................................................................... 18
  14.2 Ideas for Further Development ................................................................................. 18
15. Efficiency of Project ....................................................................................................... 19
  15.1 To satisfy the desires of the Clients .......................................................................... 19
  15.2 Utility of the software .............................................................................................. 19
  15.3 Computer efficiency ............................................................................................... 19
  15.4 Computability efficiency .......................................................................................... 19
  15.5 Code efficiency ....................................................................................................... 19
References .......................................................................................................................... 19
1 Introduction

The project Triangulation Games has been put together to create a program for playing and editing games which are explained in the document “Games on Triangulation” [1].

A triangulation of a finite planar point set $S$ is a decomposition of its convex hull whose vertices are precisely the points in $S$. This convex hull is divided by joining the vertices together with straight line segments to form faces commonly known as triangles. All the games are based on the various rules defined for making such triangulations. This project aims to develop these games on the predefined rules and also aims at bringing an easy to use interface for developing and implementing new triangulation games.

1.1 Purpose and Scope

This document describes the plan of the Triangulation Game project. An external reader can use this document as an overview of the whole project. The main purpose of this project is to highlight the idea behind triangulation games and to explore their research potential over different areas of computer science.

1.2 Product and Environment

Triangulation Games software will be standalone program developed using Java 2 Platform, Standard Edition (J2SE) [2]. During the development of the project we will be using versions 1.4 and 1.5 interchangeably. This will enable the Triangulation Games software to work in any computer which has Java 2 (version 1.4 or later) Runtime Environment (JRE) [3] installed. Java 2 Runtime is available free of charge for Microsoft Windows, Linux, Solaris SPARC and Solaris x86. There is also 64 bit version of Java 2 environment available, but for this project we will use the 32 bit version. The software will require a mouse and keyboard interface to work in a graphical environment (X window system, Microsoft Windows, Mac OS, etc.). For development purposes, we will be using Eclipse SDK Version: 3.1.1 [6].

1.3 Definitions, Acronyms and Abbreviations

Triangulation Games: The name of this project which also represents all the games that are going to model in this project.

Java: A programming language by Sun Microsystems, Inc.

1.4 References

This document and the whole project are based on the article “Games on Triangulations” [1] by Oswin Aichholzer, et al. which specifies the theoretical basis for the triangulation games.

1.5 Overview

The goal of the Triangulation Games project is to create standalone Java 2 software for playing and editing triangulation games explained in the article "Games on Triangulations" [1]. The program will be released under GPL-license which means that the program and its
source code will be available for anyone. The program is mainly made for scientists who are interested in studying game algorithms and artificial intelligence, but also for common people who like to play triangulation games

2 Current System

We are going to start doing this program from the beginning so currently there is no current system available. The first demo version of the program should be finished in a next few days.

2.1 Old System

Previously the triangulation games were only played using a pen and a paper. Such a system is limited by the working area and the time required for finishing a game. As a computer based system, no such project has been in existence before. This is going to be its first development cycle.

2.2 Other Similar/Corresponding Systems

This is the first project known for us to make Triangulation games playable with a computer. Some parts of the project may be similar to some other game, but as a whole we haven’t found any program modeling these games.

3 Advantages and Disadvantages

Many of the advantages and disadvantages come from the choice of the programming language and the skill of the development team. Such a combination can result in the following advantages and disadvantages:

3.1 Advantages

The program will be able work in different environments as it is going to be made in Java. The user will be able to change the rules of the game and will also have the capability of creating a new game by following simple development techniques.

3.2 Disadvantages

Because the program will run on the Java Virtual Machine, it will only have a limited amount of system resources thus limiting the functionality of the program. There are going to be many processor intensive functions (for example, different algorithms for calculating artificial intelligence) required by the program which may reduce the speed of the program.
4 Project Organization

All the members of the project group are studying Information Sciences in the University of Tampere.

4.1 People (project group)

Project Manager:

Ville Parviainen
e-mail: Ville.Parviainen@uta.fi
Currently a student of Masters Degree in Computer Science (Software Development)
Experience: 2 years as a Software developer at FinanssiData Oy, Osuuspankkikeskus, Finland. Familiar with programming in Java (J2ME, J2SE and J2EE), DotNet enviroment with C#, web technologies such as JSP, PHP, PERL, ASP.

Team:

Kyösti Karila
e-mail: kyosti.karila@uta.fi
Currently studying Interactive Technology at the University of Tampere.

Umair Azfar Khan
e-mail: umair.khan@uta.fi
Experience: 1 year experience as a TA at GIK Institute, Pakistan. 1 year experience as a Lecturer at Hamdard University, Pakistan. 1 year experience as a Technical Writer at Jin Technologies Pvt. Ltd. Pakistan,. Familiar with programming in C/C++, Visual C++ MFC/API and 4th Dimension.

Salvador Jesús Romero Castellano
web: http://www.alu.us.es/s/sjromcas/
e-mail: salvador_lsi@yahoo.es
Currently a student of Masters Degree in Computer Science Engineering. Familiar with different web technologies such as JSP and ASP and quite proficient in C, C++, C# (.Net) and Java Languages. Have a good experience related to graphics programming in Java.

Jon Sahlberg
e-mail: jon.sahlberg@uta.fi
Experience: Administrator in ToasNet network. Currently doing his Bachelor's thesis about gametrees and gametree pruning in the game of chess.
Knowledge: Familiar with programming in C/C++, Java, Pascal, Perl and Basic.
4.2 Related Organizations
For the group Triangulation Games project is a part of our studies in the University of Tampere.

4.2.1 Inside Company
Department of Computer Sciences, University of Tampere, Tampere, Finland.

4.2.2 Usability Team
The Usability Team is going to write the Project Usability Plan and devise an easy to use graphical user interface.

4.2.3 Client
Ph.D. Timo Poranen
Lecturer at University of Tampere, Department of Computer Sciences

Ph.D. Erkki Mäkinen
Professor at University of Tampere, Department of Computer Sciences

4.3 Changing Group Size
We started the project with five members excluding the project manager. One member of our group was out of country for a couple of week and was unapproachable. Just a few weeks after we had started the project one member of the group left but she asked to let her work remotely from her home country. We decided to keep her with us owing to her willingness for work and her past experience in the project meeting of 3rd November, 2005.

5 Project Goals and Ending/Termination
The main emphasis of this project is to understand the mechanics behind all the triangulation games. During its development, there are different goals associated with each member that has taken part in this project. These goals associated with different teams, the ending criteria and the terminating criteria are as following:

5.1 Goals of Project Group
The goal of the project group is to reach all the milestones in time and to complete the project within the allotted time. The team aims to acquire in-depth knowledge about triangulation games using the waterfall development model and wants to implement the same using Java as the high level development language.
5.2 Goals of Client
The client aims to acquire software that is easy to use and provides practical examples of the different triangulation games. This software is then going to be used for doing research and the implementation of new triangulation games, with the user-defined rules.

5.3 Goals of the Project
The project's goal is to develop fast, easy to use software that is workable on different environments and provides space for research and user modifiability.

5.4 Termination Criteria
The project will be terminated in case the project group fails to complete the tasks on time and does not provide regular progress report. The completion of the first prototype is also a condition to note the progress of the group and the failure in its delivery can also result in the project's termination. If one of the major risks comes true, that might also result in the termination of the project.

5.5 Ending Criteria
The project will be considered to be complete if it satisfies the client's needs. The most basic requirements must be met in order to term the project as a success. The ending criteria include the completion of the software and all the relevant documentation as expressed in Chapter 6.1.

6 Project Phases and Timing
The project uses the waterfall development approach for on time completion of different phases. The different methods and tools needed for on-time completion of this project are as following:

6.1 Methods and Tools
To strictly follow the waterfall model, it is highly imperative that all the deadlines are followed and all the requirements are completed to the letter. Weekly meetings, assigning of tasks and continuous consultation among the project members, manager and the client will ensure that everyone remains up-to-date on the requirements and progress. Microsoft's Project Manager will be used to calculate the relevant progress and assign time-lines for the completion of the software.
6.2 Phases

We are using the waterfall software development model. Next table has the work breakdown structure (WBS) with scheduled starting and ending date for each task.

<table>
<thead>
<tr>
<th>WBS</th>
<th>Name</th>
<th>Start date</th>
<th>End date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Learning phase</td>
<td>Fri. 14.10.2005</td>
<td>Fri 2.12.2005</td>
</tr>
<tr>
<td>1.1</td>
<td>Get to know with triangulation games</td>
<td>Fri 14.10.2005</td>
<td>Fri 18.11.2005</td>
</tr>
<tr>
<td>1.2</td>
<td>Study java</td>
<td>Fri 14.10.2005</td>
<td>Fri 2.12.2005</td>
</tr>
<tr>
<td>1.3</td>
<td>Prototype</td>
<td>Mon 21.11.2005</td>
<td>Fri 16.12.2005</td>
</tr>
<tr>
<td>1.4</td>
<td>Lectures</td>
<td>Fri 14.10.2005</td>
<td>Fri 24.2.2005</td>
</tr>
<tr>
<td>2</td>
<td>Requirements</td>
<td>Mon 10.10.2005</td>
<td>Fri 16.12.2005</td>
</tr>
<tr>
<td>2.1</td>
<td>Requirements phase RE Group</td>
<td>Mon 10.10.2005</td>
<td>Thu 8.12.2005</td>
</tr>
<tr>
<td>3</td>
<td>Design</td>
<td>Mon 19.12.2005</td>
<td>Fri 3.3.2006</td>
</tr>
<tr>
<td>3.1</td>
<td>Implementation plan</td>
<td>Mon 19.12.2005</td>
<td>Fri 17.2.2006</td>
</tr>
<tr>
<td>3.2</td>
<td>Test plan</td>
<td>Mon 19.12.2005</td>
<td>Fri 3.2.2006</td>
</tr>
<tr>
<td>4</td>
<td>Implementation</td>
<td>Mon 17.2.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>4.1</td>
<td>User interface</td>
<td>Mon 17.2.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>4.2</td>
<td>Artificial intelligence</td>
<td>Mon 17.2.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>4.3</td>
<td>Game handling</td>
<td>Mon 17.2.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>4.4</td>
<td>Software logic</td>
<td>Mon 17.2.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>5</td>
<td>Testing</td>
<td>Mon 3.4.2006</td>
<td>Fri 5.5.2006</td>
</tr>
<tr>
<td>5.1</td>
<td>Module testing</td>
<td>Mon 3.4.2006</td>
<td>Fri 21.4.2006</td>
</tr>
<tr>
<td>5.2</td>
<td>Integration testing</td>
<td>Mon 24.4.2006</td>
<td>Fri 5.5.2006</td>
</tr>
<tr>
<td>5.3</td>
<td>Approve testing</td>
<td>Thu 4.5.2006</td>
<td>Fri 5.5.2006</td>
</tr>
<tr>
<td>6</td>
<td>Ending the project</td>
<td>Mon 8.5.2006</td>
<td>Fri 12.5.2006</td>
</tr>
<tr>
<td>6.1</td>
<td>Final reports</td>
<td>Mon 8.5.2006</td>
<td>Fri 12.5.2006</td>
</tr>
<tr>
<td>6.2</td>
<td>Operating instructions</td>
<td>Mon 8.5.2006</td>
<td>Fri 12.5.2006</td>
</tr>
<tr>
<td>6.3</td>
<td>Project presentations</td>
<td>May</td>
<td>May</td>
</tr>
</tbody>
</table>
In this picture the lifespan of this project is displayed as it has been scheduled by the project manager.

<table>
<thead>
<tr>
<th>No.</th>
<th>Task</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Learning phase</td>
<td>3.05</td>
<td>4.06</td>
</tr>
<tr>
<td>2</td>
<td>Get to know with simulation</td>
<td>5.05</td>
<td>6.06</td>
</tr>
<tr>
<td>3</td>
<td>Study Java</td>
<td>11.05</td>
<td>12.06</td>
</tr>
<tr>
<td>4</td>
<td>Prototype</td>
<td>0.01</td>
<td>1.02</td>
</tr>
<tr>
<td>5</td>
<td>Examinations</td>
<td>2.06</td>
<td>3.07</td>
</tr>
<tr>
<td>6</td>
<td>Requirements phase RE Group</td>
<td>4.06</td>
<td>5.07</td>
</tr>
<tr>
<td>7</td>
<td>Requirements document</td>
<td>6.07</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Implementation plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Test plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Implementation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>User interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Application intelligence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Games handling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Software logs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Miscellaneous testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Integration testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Acceptance testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Ending the project</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Final inspections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Operating instructions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Project presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6.3 Work and Task Estimates

Development Team = 5 developers, 240 hours each  
Usability Team = 50 hours  
Project Manager = 200 hours  
Total = 1450 hours

### 6.4 Deliverables

The deliverables consist of different software phases and the documents as associated with the water fall model. The different phases of software include the prototype and the completed software. The documentation that is required to be delivered is:

- Project Plan
- Requirements Specifications Document
- Design Document
- Maintenance Document
- Final Project Report.
- Source Code
- User Guide and,
- A web page for downloading the software
7 Monitoring and Guidance

In order to ensure the project goes well and that the specifications are followed, the project team will set some monitoring and guidance policy.

7.1 Inside Group

The project team will have regular meetings, contact through Instant Messengers, E-mail communications and working-hour tables to do the monitoring. The requirement specifications and feedback from the clients will help with proper guidance in completing the project.

7.1.1 Meetings

In order to monitor the progress of the project, the project group will be subjected to weekly meetings along with some informal meetings depending on need. During the weekly meetings, Team members will discuss issues and progress through face-to-face meetings and all conclusions and decisions will be documented in the meeting reports.

The informal meetings will be called throughout the entire project on an as-needed basis which will be determined by either team members or the clients.

7.1.2 E-mails

E-mail will be the daily communication channel for solving instant problems and confusions. E-mails will also be used for scheduling the weekly meetings and transferring documents.

7.1.3 Status Report

Each week, every member has to fill a working-hour table for that week. The working-hour table separates the whole project into six stages and lists eight different working items for each stage. With the help of the working-hour table, both project manager and team members will be able to keep track of the time spent on the project.

7.2 Outside Group

The Project Manager will discuss the progress of the triangulation project for each scheduled meeting between the team members, the client representative, and the course lecturer. Requirements Specification document will be used as the conformance standard for making the project.

7.2.1 Client

The communication with the clients will be divided into review meetings and status reports and different checkpoints to let them comprehend the current situation.

Review Meetings

The Project Manager will discuss the progress of the triangulation project for each scheduled meeting between the team members, the client representative, and the course lecture. Different milestones such as the completion of project plan, requirements specifications report, design document, etc., will be discussed during these review meetings.
Weekly and Monthly Status Reports
The Project Manager will update the status report on a weekly/monthly basis to provide the approvers information about recent accomplishments, and efforts expended. All of the information will be summarized in the Technical Status Report (TSR). The TSR is the official report for the client. It will be prepared monthly, and will be distributed according to the standard distribution list and technical monitor.

Checkpoints
Each stage will conclude with a formal checkpoint called “stage exit”. When a stage has been successfully exited, this will indicate that all the deliverables due on that date have been completed and all outstanding issues have been resolved.

The checkpoint process will consist of a meeting of client and project manager to present and review the completed objectives. The project’s designated approvers and any affected functional areas involved in the project must provide a written position of concur/non-concur at each stage checkpoint.

7.2.2 Course
Weekly working-hour table will indicate the time and effort each member spending on the project. The project manager will discuss with the lecturer on the situation and performance of each member.

7.3 Others
Besides the project team, lecture, and client representatives, other people with relation to this project will attend the meetings when needed. They will also be provided with the related information. The project manager will decide who and when to attend the meetings.

8 Standards, Directives, Guidelines
The project team has to follow the rules given below during the process of developing Triangulation Games.

8.1 Standards and Directives
The implementation, functions and user interfaces standards will be from the requirements specification document. We will be using the Code Conventions for the Java Programming Language document for coding and documentation.

8.2 Confidentiality
This project is going to have no confidentiality. After the completion of this project, all the source code will be available according to the terms of the GPL license.
8.3 IPR and Copyrights
The accomplished product and documents will be applied to the GNU GENERAL PUBLIC LICENSE 2.0 and this will be added to the code. For the terms and conditions please refer to: http://www.gnu.org/copyleft/gpl.html#TOC1

9 Risk Management
The following table has all the risk the project team has come up with. Analysis number tells if the risk is high or low (higher number higher risk). Probability Serious and predictability are on a scale of 1-5.

<table>
<thead>
<tr>
<th>Risks</th>
<th>Analyses</th>
<th>Probability</th>
<th>Serious</th>
<th>Predictability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team member leaves</td>
<td>Team member may leave because of personal reasons.</td>
<td>45</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Team member may leave because of personal reasons. How to avoid: Make sure every member understand the importance of the project and has the same determination to achieve it. Minimize the risk: Make a backup plan, if the team member really leaves, the rest of the team can still complete the project. Before the team member is leaving, he/she should document all the working items he/she has done, so the other people can take it over.</td>
<td>32</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Static requirements</td>
<td>The static requirements may be beyond the ability of the team member.</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Develop a program that is not usable</td>
<td>The resulted program may not fit the client’s requirements or cannot run on the assigned platform. How to avoid: Perform system analysis and system design carefully before writing the program. Do the checkpoints carefully. Minimize the risk: Discuss with the lecturer, team members and client representatives and see if this can be resolved by changing the design or implementation.</td>
<td>50</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Incompetent project management</td>
<td>Incompetent project management may lead to failure of the project. How to avoid: The project manager will remain up to date with the latest technology and keep track of the project development. Minimize the risk: Discuss with the lecturer and team members and try to reorganize the team.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Lack of knowledge | Lack of technical and domain knowledge of the project.  
How to avoid: All the team members have to study hard beforehand, the client should provide learning resources or contact channels for the development team.  
Minimize the risk: Ask for help from experts. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>75 5 5 3</td>
</tr>
</tbody>
</table>
| Time related issues | There may not be enough time for completing this project.  
How to avoid: Conduct good time control and monitoring all the time.  
Minimize the risk: Discuss with the lecturer and the client representatives, see if it's possible to either reduce the scale of the project or extend the developing time. |
|                   | 48 4 4 3 |
| Too ambitious requirements | The client may ask for too many functions for this project.  
How to avoid: Perform system analysis and system design carefully before writing the program. If there are too ambitious requirements, discuss them with the lecturer and the client representatives.  
Minimize the risk: Discuss them with the lecturer and client representatives. Delay the implementing these requirements till next version or reduce the requirements. |
|                   | 60 4 3 5 |
| Bad design | It can include: bad system design or user interface design or working flow design.  
How to avoid: Analyze the design the system properly at the beginning; consult the expert, lecturer and client representatives for their opinions. Review and integrate the system all the time.  
Minimize the risk: Consult the experts and lecturer. If there is still enough time, redesign and re-implement the code. |
|                   | 16 2 4 2 |
| Lack of client interaction | The project team is possible to lose the real time feedback from the client due to the lack of communication and interaction. This may lead to unusable or unqualified program.  
How to avoid: Conduct review meetings and status reports regularly, make sure there is a contact channel between the project team and the client.  
Minimize the risk: Visit the client regularly |
|                   | 2 1 2 1 |
| Lack of hardware resource | The project team may not have enough hardware for developing this project, such as testing machines.  
How to avoid: Prepare the hardware according to the system analysis  
Minimize the risk: Discuss this problem with the client and lecturer. |
|                   | 6 3 2 1 |
**10 Learning and Studying Plan**

In this section we consider the necessary knowledge for the project; it is divided in two sections, one for the Project Group and the other for the Client.

**10.1 (To) Project Group**

We can classify the necessary knowledge of the Project Group into theoretical knowledge and practical knowledge. The group members are supposed to be up to date on all the theory related with the project and must update their skills about the tools that are going to be used in this project.

**10.1.1 Theoretical Knowledge**

First of all, the team has to read and understand the article "Games on Triangulations" [1]. The team must have a good grasp about the games described there and about their mathematical logic as it will help to make a good design of the application.

**10.1.2 Practical Knowledge**

Everyone should have a good knowledge about the basic Java programming (control structures, objects, inheritance, interfaces, cloning and data structures). Also it will be necessary to know the Java 2D libraries for drawing in the interface using a mouse or a keyboard, as well as the swing libraries.

Some basic of the Java introspection API will be probably necessary to load new games or algorithms, as well as the file management and the XML library.

The group plans to gain this formation mainly from the manuals at the Java Web Site [4].

Knowledge about software engineering process, specially regarding the documents and methods necessary for this project will be also needed.
10.2 (To) Client
As the Client is also the lecturer of the subject it is assumed that he already knows all the engineering process that we are going to follow. Occasionally, the group will give some information about the technology used, so he will have to remain updated on the technology.

11 Installation Plan
The software will be delivered in a compressed format (like Zip or similar). The user will extract the files from the compressed package and will install the software in the local machine. Since there won’t be net usage in this software, no more net or path adjustments will be needed. Depending on the final design of the system, new games or algorithms will be installed in it by just creating new files. In this case, there will be a folder in the directory structure of the files of the software where to copy the files representing games.

This installation procedure is very common in important Java Projects and Software, like Eclipse [6]. There will be a dedicated website for downloading the documents and software online from the internet for easy installation.

The installed files set will contain a script file to make it easier to run in Windows and Linux. All the files will be available on the project’s web site for downloading.

12 Starting (Initialization, Deployment) Plan
The program starting will be made on a Jar file which will load one concrete class, or with a script file in the case of Windows or Linux. For other systems, and also for the previous ones, there will be always the possibility of run the program from the command line. The parameters of the command have not been specified yet (it will be done at design or implementation phase).

For the deployment of new games or algorithm, some possibilities are to be considered:

1. To move the file that represents the game or the algorithm to a predetermined folder.
2. To load a file representing the game or the algorithm from a file in the file system with a menu, after the program has been started.

The selection of one choice or the other will depend on the final requirements document, the analysis of the application, and the usability document.

The way to define new games or algorithms will be decided in requirement and/or design phases.
13 Costs (Budget)

We are using a COCOMO (Constructive Cost Model) to estimate the costs of this project. From COCOMO model we are using the basic model and its indicators. Besides these indicators we also need an estimate of the lines of code the end program is going to need. For this estimation we broke down the software in smaller pieces and estimated the lines of code needed to implement them. Here are the pieces and the estimations.

<table>
<thead>
<tr>
<th>Component</th>
<th>Lines of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>350</td>
</tr>
<tr>
<td>Engine</td>
<td>450</td>
</tr>
<tr>
<td>AI</td>
<td>350</td>
</tr>
<tr>
<td>File handling (game opening)</td>
<td>150</td>
</tr>
<tr>
<td>Opening positions</td>
<td>100</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1400</strong></td>
</tr>
</tbody>
</table>

Now we can use the total number of codes in the COCOMO model and input it into the model. We are using the KLOC value so the total KLOC for 1400 lines of code is 1.4.

\[
E = 2.4 \times 1.4^{1.05} \approx 3.4 \\
D = 2.5 \times 3.4^{0.38} \approx 4
\]

Here \( E \) is the effort (in person-months) needed to complete the project. \( D \) is the time in chronological months to complete the work. With these numbers we can calculate the average costs of the project.

For this project as being a school project one person-month actually costs nothing. However for the learning purposes we are going to estimate that one person-month costs 5000 Euros. This includes all the salary costs as well as all other costs the project might have (resource acquisition like computers etc.). So for the total costs for the project the estimate is the following.

\[
3.4 \times 5000 = 17000 \text{ Euros}
\]
14 Rejected Alternatives and New Ideas

During the course of development, many alternatives and new ideas are discussed. Some of these are as following:

14.1 Rejected Alternatives

Since the group has had access to previous versions of the requirements documents since the beginning that specified many things, and close contact with the client, no stray ideas have been developed and consequently rejected.

14.2 Ideas for Further Development

Several ideas for the graphic environment are still under discussion. Adding new algorithms dynamically is an idea that will be left for future development. The possibility to design games with more than two players can be also added in advance.

Net gaming is also a great idea for future iterations of the software. Combining this point and the previous, the idea of having several computers competing with each other through the Internet is very appealing to be ignored without considering it deeply.

A wizard program that manages the installation on windows platform may also help in making the program reach more people.

15 Efficiency of Project

The following aspect will be relevant to evaluate the efficiency of the Project:

15.1 To satisfy the desires of the Clients

The group has to pay attention to fulfill the requirements at the requirements document.

15.2 Utility of the software

Reasons that make the software useful have to be considered in order to make better software. At this point, reasons that will make the system useful could be:

- Help researches to understand triangulation games
- Help them to think new theories about triangulation games
- Help them in trying new games
- Help the researches to try new algorithms for solve triangulation games
- Help student to improve their computer skill by developing new games or algorithms
- To help the people to have fun!

15.3 Computer efficiency

We have to keep in mind that, since the program is written in Java, it's not going to be as fast as if it were written in another language like Visual C++. This is the price we have to pay for portability.
15.4 Computability efficiency
All the algorithms implemented in the program, especially those regarding some AI, have to have P-Completeness and PSPACE-Completeness, in order to avoid uncontrolled efficiency lags.

15.5 Code efficiency
Since the software is going to be published under a GNU license, the code must be understandable by a third party. A good design documentation, manuals and tutorials will also be needed.
References

1] Games on Triangulations, Oswin Aichholzer, David Bremner, et al., Theoretical Computer Science 343 (2005) 42-17 (available online at www.sciencedirect.com)


3] Java 2 Runtime Environment (J2RE), Environment to run Java 2 programs, this is included in J2SE. (avaliable at http://java.sun.com)

