Norwegian University of Science and Technology

IT2901 - Project 2

Don't Panic!

A Serious Electronic Board Game

Authors:
Jens Even Blomsøy, Stian Aurheim, Jørgen Foss Eri
Sindre Svendsrud, Adrian Arne Skogvold, Jim Frode Hoff

Customer:
Ines Di Loreto

Supervisor:
Mohsen Anvaari

April 8, 2013

Contents

1 Introduction

1.1 The course

The main goals in this course are to experience and learn how to work on a development project as a team. In addition, the team has to answer to a customer, as software development companies often do, which stands out from other projects in the past. This is an advanced course and it is expected that knowledge obtained from previous courses is used, especially the development courses such as Informatics Project I and the collaborative System Development project.

The group has an appointed guidance counselor as well as a customer. The counselor will be available for answering questions regarding the project management in general, and push the group to reflect on its decisions and review the work done. Status reports will be delivered regularly, so the counselor can stay up-to-date with the work in the group.

During the course, several project reports are scheduled for delivery; the preliminary project report, the mid-term project report and the final project report. Working on and delivering these reports will help in the planning and development of the project, and feedback will be given from the counselor. The grading of the project will take the final project report into consideration, as well as the final product.

1.2 The team

The team consists of six students of Informatics at NTNU:

Stian Aurheim

- Third year bachelor in Informatics
- Main experience in Java. Some experience in Python, PHP, HTML, JavaScript

Jens Even Berg Blomsøy

- Third year bachelor in Informatics
- Programming languages worked with: Java, Python.
- Main knowledge in System Development, system architecture and system documentation.

Jørgen Foss Eri

- Third year bachelor in Informatics
- Experience with Java, Python, JavaScript/HTML5/CSS3 and general web development

Jim Frode Hoff

- Third year bachelor in Informatics
- Programming languages worked with: Java, Python, PHP, JavaScript and general web development

Adrian Arne Skogvold

- Third year bachelor in Informatics
- Programming languages worked with: Java, C#, Oz, Actionscript

Sindre Svendsrud

- Third year bachelor in Informatics
- Experience with Java, Python and C++

1.3 Problem description

The customer has developed a paper prototype [Figure 1] of a board game called Don't Panic. The game is designed to help crisis workers make the right decisions during a crisis in a city. It is a turn based, collaborative multiplayer game where the players take actions to stop the inhabitants from panicking. After the game is finished, an expert will review the actions with the players to evaluate whether their choices were sound.

Our customer wants an electronic version of the board game. The electronic board game should maintain the social aspect (both physical and verbal) of a regular board game. In addition, a replay function will be added, to make it easy for the expert to review the game with the different players. The physical version of the board game takes a lot of work setting up and maintaining, as it is time consuming to move the pieces and update the panic levels. The electronic version will automate all of this.

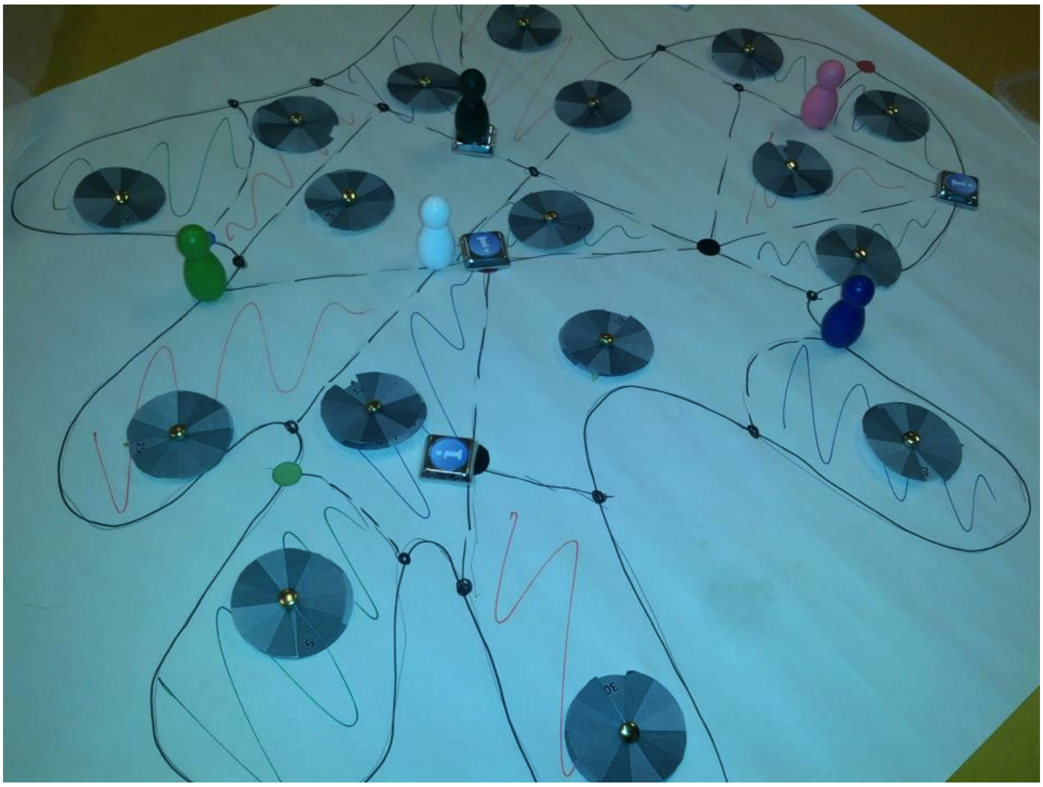

Figure 1: Picture of the paper prototype

1.4 Constraints

Only a few of us have some past experience with HTML5 and JavaScript
The game is to be completed within one semester (21 January - 27 May)

1.5 Customer and supervisor

The customer for this project is Ines Di Loreto, a researcher in the Department of Computer & Information Science at NTNU. The supervisor assigned for this project is Mohsen Anvaari, a PhD candidate in the same department.

2 Requirements

2.1 Functional requirements

- The game interface should provide experts with the ability to efficiently manage and control the players.
- The players must be able to interact with the client using an I/O device.
- The game interface should have elements for managing game sessions.
- All game objects that exist longer than a single task, must be created and kept in storage. An object is anything from the players' data to an invisible trigger object.
- Game rules should be accepted as parameters in input. These parameters should be set by the experts through the management client.
- The expert should be able to follow a game session and comment on the tracking of the game, but should not be able to participate.
- The expert should have the ability to effectively monitor the server.

2.1.1 Use cases

The use cases are mainly based on the functional requirements of the game and are a graphical representation of the users' interactions with the board game. They document all the different ways in which the user can interact with the game.
A detailed set of use case diagrams and textual use cases are provided below.

ID	01
Name	Login Player
Goal	To be connected to the server
Actors	Player, server
Start requirements	None
End requirements	- The player gets logged in. - The game is displayed.
Case	- The player gets prompted with the login window. - The players gives login-info. - The player is now logged in.
Alternative Case	Wrong password
Previous Use Case	None
Spawned Use Case	05

Table 1: Use Case: Login player

ID	02
Name	Login Expert
Goal	To be connected to the server
Actors	Expert, server
Start requirements	None
End requirements	- The expert gets logged in. - The expert view is displayed.
Case	- The expert gets prompted with the login window. - The expert gives login-info. - The expert is now logged in.
Alternative Case	Wrong password
Previous Use Case	None
Spawned Use Case	03

Table 2: Use Case: Login expert

ID	03
Name	Game Setup
Goal	To create a successful game session
Actors	Expert, server
Start requirements	The expert is logged in
End requirements	The expert is able to create a game setup The expert is able to save the game setup
Case	The expert creates the appropriate map for the game. The expert adds the wanted board pieces. The expert manages the zones, the number of people and panic levels. The expert manages the cards, adds the wanted cards to the game. The expert adds the wanted number of players to the game. The expert assigns roles to each player. The expert sets the starting point for each player.
Alternative Case	None
Previous Use Case	02
Spawned Use Case	04, 05

Table 3: Use Case: Game Setup

ID	04
Name	Watcher
Goal	To get a non player version of the game
Actors	Expert, server
Start requirements	The expert is logged in, a game is running
End requirements	The expert is able to watch the wanted game
Case	The expert selects the game to watch from a list of games The server provides a game window in which the expert is not participating as a player
Alternative Case	None
Previous Use Case	02
Spawned Use Case	None

Table 4: Use Case: Watcher

ID	05
Name	Join Game
Goal	To successfully join a starting game
Actors	User, game session, server
Start requirements	A game has been created by the exper
End requirements	A user is able to join the appropriate game
Case	The user clicks on join options for the game The game asks for user info The user joins the game
Alternative Case	The user gives incorrect info and is not added to the game
Previous Use Case	03
Spawned Use Case	06, 07

Table 5: Use Case: Join Game

List of Figures

List of Tables