

Efterstudie

Grupp 5
Version 1.0

Status

Granskad		
Godkänd		

AntiTD

Grupp 5, HT 18-2
Umeå Universitet, Institutionen för datavetenskap

Namn	Ansvar	Telefon	E-post
Jonas Sjödin	Projektleddare (KUN)	070-274 71 95	id16jsn@cs.umu.se
Petter Skogh	dokumentansvarig (DOK)	073-971 80 19	id16psh@cs.umu.se
Petter Skogh	Designansvarig (DES)	073-971 80 19	id16psh@cs.umu.se
Jonas Sjödin	testansvarig (TST)	070-274 71 95	id16jsn@cs.umu.se
Marcus Jonsson	kvalitetssamordnare (QS)	070-380 23 85	c17mjn@cs.umu.se
Martin Hedberg	implementationsansvarig (IMP)	070-339 90 12	c17mhg@cs.umu.se

E-postlista för hela gruppen: id16jsn@cs.umu.se

Hemsida: Nä

Kund: Johan Eliasson, Umeå Universitet,
johane@cs.umu.se

Kontaktperson hos kund: Johan Eliasson, johane@cs.umu.se

Kursansvarig: Johan Eliasson, johane@cs.umu.se

Handledare: Ander Broberg, bopspe@cs.umu.se

Elias Åström, eam@cs.umu.se

Jakob Lindqvist, jakobl@cs.umu.se

Klas af Geijerstam Unger, klasa@cs.umu.se

William Viktorsson, willamv@cs.umu.se

Innehåll

<u>1</u>	<u>TIDSÅTGÅNG</u>	4
<u>1.1</u>	<u>ARBETSFÖRDELNING</u>	4
<u>1.2</u>	<u>TIDSÅTGÅNG JÄMFÖRT MED PLANERAD TID</u>	4
<u>2</u>	<u>ANALYS AV ARBETE OCH PROBLEM</u>	4
<u>2.1</u>	<u>VAD HÄNDE UNDER DE OLIKA FASERNA (BRA/DÅLIGT/ORSAK)?</u>	4
<u>2.2</u>	<u>HUR VI ARBETADE TILLSAMMANS (ANSVAR, BESLUT, KOMMUNIKATION ETC.)?</u>	5
<u>2.4</u>	<u>HUR FUNGERADE RELATIONEN MED BESTÄLLAREN?</u>	5
<u>2.5</u>	<u>HUR FUNGERADE RELATIONEN MED HANDEDAREN?</u>	5
<u>2.6</u>	<u>TEKNISKA FRAMGÅNGAR/PROBLEM</u>	5
<u>3</u>	<u>MÅLUPPFYLLELSE</u>	5
<u>3.1</u>	<u>VAD HAR UPPNÅTTS?</u>	5
<u>3.2</u>	<u>HUR FUNGERADE LEVERANSEN?</u>	5
<u>3.3</u>	<u>HUR HAR STUDIESITUATIONEN PÅVERKAT PROJEKTET?</u>	5
<u>4</u>	<u>SAMMANFATTNING</u>	6
<u>4.1</u>	<u>DE TRE VIKTIGASTE ERFARENHETERNA</u>	6
<u>4.2</u>	<u>GODA RÅD TILL DE SOM SKA UTFÖRA ETT LIKNANDE PROJEKT</u>	6

1 Tidsåtgång

1.1 Arbetsfördelning

Huvudansvar:

Marcus Jonsson - GUI och event handlers

Martin Hedberg - Troops, Towers och programmet

Jonas Sjödin - XML, hantering av XML, databas och programmet

Petter Skogh - Game state klassen World, grafik och grafik generering i programmet

1.2 Tidsåtgång jämfört med planerad tid

Fas	Planerad tid i timmar	Använd tid i timmar
Före	12	8
Under	100	90
Efter	8	8

Aktivitet	Planerad tid i timmar	Använd tid i timmar
Testning	12	8
Karta	12	10
World	8	7
GUI	12	12
Grafik	10	11
Grafik generering	5	4
Kontroller	12	12
Enemy	8	7
Player	2	2
Trupper och torn	8	8
Rapport och presentation	8	8

2 Analys av arbete och problem

2.1 Vad hände under de olika faserna (bra/dåligt/orsak)?

Planering

- Bra planering underlättade överlag under dom andra faserna. Små oklarheter då planering inte gick på djupaste detalj.
- Satte upp designen och modellen för projektet
- Delade ut ansvarsområden
- Diskuterade med handledare våran lösning

Utveckling

- Dåligt val av frame (canvas) för spelfältet då flicker kunde uppstå. Därför byttes det till BufferedImages.
- Arbetade med våra ansvarsområden tillsammans men separat och satte ihop dem. Eftersom att vi satt tillsammans och utvecklade kunde vi ha en pågående diskussion och hjälpa varandra i rätt riktning.

Finputsning

- Efter att spelet satts ihop letade vi efter buggar och fixade dessa samt finputsade spelet. Vi kollade även då igenom projektspecifikationen och såg till så att vi nådde upp till de krav som projektet skulle innehålla.
- Vi utvärderade klassernas ansvarsområde och flyttade metoder och ansvar utifrån en objekt-orienterad standard.
- Vissa klasser som endast gjorde statiska operationer såsom Database, GraphicsGenerator och ParseMap gjordes statiska med privat konstruktor så att de inte kunde instansieras och därigenom endast nås statiskt. Endast konstanter lagrades i dessa klasser, alla annan data kom från parametrar. Detta underlättade strukturen och förtydligade systemet.

Rapport och presentation

- Sist i processen skapades rapporten tillsammans i latex där vi använde oss av overleaf för att synka redigering i realtid.

2.2 Hur vi arbetade tillsammans (ansvar, beslut, kommunikation etc.)?

Ansvar

Som i avsnitt 1.1 fördelades ansvaren och den person som ansvarade för ett område såg också till att det blev färdigt men hjälp fick tas av gruppmedlemmar. Projektledare har varit Jonas Sjödin men ansvaret för projektet har funnits på hela gruppen.

Löpande möten

Gruppen möttes i löpande möten då arbetet i de individuella ansvarsområdena kunde slås ihop och lösningar diskuteras.

Messenger

All kommunikation skedde över Facebook messenger. Där bestämdes tid och plats för möten och utvecklingstillfällen.

2.3 Hur fungerade relationen med beställaren?

Vi har inte haft någon kommunikation med beställaren under projektet.

2.4 Hur fungerade relationen med handledaren?

Lite luddigt vad som krävdes men de har varit hjälpsamma då vi haft frågor om uppgiften.

1.1 Tekniska framgångar/problem

Databasen var lite problematisk då vi kontaktade ITS istället för CS supporten och därav inte fick någon databas. Men efter det så gick allt felfritt. Då vi arbetat i olika utvecklingsmiljöer så har vi även använt olika tecken encodings på våra filer vilket ledde till mycket inläsnings och kompileringsproblem. Tillslut så blev det windows-encodingen windows-1252 som användes. Detta för att så många filer var i denna encoding då det är standard i eclipse och därav var övergången till att använda dessa enklare än att använda UTF-8 vilket hade varit rimligare.

3 Måluppfyllelse

1.1 Vad har uppnåtts?

Ett spel som följer de givna kriterierna från beställaren.

1.2 Hur fungerade leveransen?

Bra

1.3 Hur har studiesituationen påverkat projektet?

Eftersom gruppmedlemmar går olika kurser har föreläsningar ibland kommit i vägen för möten.

2 Sammanfattning

2.1 De tre viktigaste erfarenheterna

Planering

Under projektets start fick vi möjlighet att göra en genomförlig planering som vi kunde få feedback på av handledare. Planeringen var bra för att få erfarenhet i att planera upp ett program som man sedan ska arbeta flera på.

Samarbeta med kod

Projektet har varit väldigt nyttigt för att få erfarenhet i att arbeta tillsammans på ett programmeringsprojekt och vad som krävs.

Tänka igenom alla features till implementationen man planerat

Vi har lärt oss att det är viktigt att tänka igenom alla delar även om man tycker att de verkar väldigt enkla att implementera sen eftersom dessa ofta skapar problem senare.

2.2 Goda råd till de som ska utföra ett liknande projekt

Planera.