
DAT220 Høsten 2013

Forprosjektrapport

X1337

Av Gunnar Abdusal, Per Arne Andersen
og Shahab Afsharipour

Innhold

Innledning	4
Prosjektstrategi	5
Prosjektroller	5
Scrummaster	5
Gitmaster	5
Programmerere	5
Prioriteringsliste	6
A- mål	6
B- mål	6
D- mål	6
Spesifikasjoner	7
A – Mål	7
Spiller	7
Fiender	7
Bosser	7
Kuler	7
Stages	7
B - Mål	8
Vanskelighetsgrad	8
Meny	8
Lyd	8
Bakgrunnsbilde	8
Highscore	8
C - Mål	9
Power-ups	9
Stage select	9
Special Effects	9
D - Mål	9
Online Multiplayer	9
Online Highscore	9
Forskjellige typer skip	9
Achievements	9

Klasse Diagram	10
Standarder	11
Verktøy	14
JIRA	14
GIT	14
Visual Studio 2012	14
Boost.....	14
LucidChart.....	15
Google Docs & Microsoft Office.....	15
Dropbox.....	15
SFML.....	15
Simple and Fast Multimedia Library er et bibliotek for C++. Vi har valgt å ta dette i bruk i vår utvikling av X1337. SFML er et uhyrekjapt og lettvekt bibliotek som gir oss muligheten til å tegne opp 2d grafikk. Det er hardware akselerert med OpenGL som også er cross compatible. SFML er delt opp i 6 forskjellige moduler. I disse modulene finner du:.....	
• System Module: Håndterer bla multithreading om vi ønsker å benytte dette.....	15
• Window Module: Håndterer vindu og User Input.....	15
• Graphics Module: Inneholder klasser for shapes, sprites og mye mer.	15
• Audio Module: Lydhåndtering	15
• Network Module: Nettverksprogrammering I form av sockets.	15
Alle disse modulene kan bli brukt uavhengig av hverandre (med unntak av graphics og window module), Dette er noe av det som gjør SFML et fleksibelt og godt bibliotek for programmering av 2d grafikk.	
Gantt skjema	16

Innledning

Ideen vår er å lage et "bullethell - top - down shooter"-spill som består av at spilleren spiller mot datamaskinen (CPU). Spilleren vil kunne bevege seg fritt inne på banen i alle retninger. Et bullethell spill har top-down left-right oversikt (view). Spillet vil ha to forskjellige vanskelighetsgrader: easy og Hard. På easy har spilleren liv (HP), som synker etterhvert som spilleren blir truffet av motstanderens prosjektiler (bullet). På hard vil spilleren ikke ha flere liv, og dø med en gang spilleren blir truffet av en kule.

Begrepet "Bullet Hell" refererer til en sub-genre av skytespill der det viktigste er å unngå tilsynelatende endeløse bølger av kuler fra fiender som har form av skip, bakkekanoner, eller andre type former.

I et bullethell spill er det som regel flere motstandere som samtidig sender ut ulike bullets som ofte har forskjellige farger og som har ulike typer negative effekter. Noen prosjektiler skader spillerens skip (mister HP), mens andre kan for eksempel hemme kontrollen av skipet til spilleren. Spillet har flere forskjellige stadier og på enden av hvert stadie møter spilleren en veldig vanskelig fiende (boss) som vil gjøre det utfordrende for spilleren å vinne over brettet (level) eller spillet.

Bullethell spill er kjent for å bli svært intensive og setter spillernes ferdigheter på prøve. Spilleren må ha god presisjon, hukommelse og reaksjonsevne for å beseire spillet. Det er heller ikke uvanlig at bullethell spill har mange visuelle effekter og farger for å bryte spillerens konsentrasjon

Prosjektstrategi

Det er vanskelig på forhånd å vite hvor krevende mange av oppgavene i prosjektet er. Vi har derfor valgt å sette opp en prioriteringsliste, hvor vi rangerer hver av de funksjonene vi ønsker å implementere fra A-D.

Vår prioriteringsdefinisjon er som følger:

A - Funksjoner som **må** implementeres

B - Funksjoner som **bør** implementeres

C - Funksjoner som **kan** implementeres

D - Forslag til ekstrafunksjoner (implementasjon er ikke forventet)

Gruppen bruker JIRA for oversikt over de viktigste delene som må utføres før de andre.

Gruppen har også avtalt faste arbeidsdager som vi møter opp på grupperom for å kunne diskutere utviklingsstrategi og uavklarte momenter som dukker opp i utviklingen.

Prosjektroller

Scrummaster

Shahab Afsharipour & Per-Arne Andersen

Gitmaster

Shahab Afsharipour

Programmerere

Gunnar Abusdal

Per-Arne Andersen

Shahab Afsharipour

Prioriteringsliste

A- mål

Spiller
Fiender
Boss
Kuler
Stages

B- mål

Vanskelighetsgrad
Meny
Lyd
Bakgrunnsbilde
Andre typer fiendemodeller
Highscore

C- mål

Stage select
Power-ups
Special Effects

D- mål

Online Multiplayer
Online Highscore
Achievements

Spesifikasjoner

Under denne seksjonen forklares de forskjellige funksjonene vi ønsker at spillet skal ha. Funksjonene rangeres slik at noen av funksjonene kun implementeres dersom det er tilstrekkelig nok tid i prosjektet, og for å unngå å tvinne tommer om vi gruppen innfrir alle mål.

A – Mål

Spiller

Vi skal implementere en karakter som kan styres av spillere. Spilleren skal unngå å bli truffet av kuler som blir skutt av diverse fiender og samtidig å bekjempe bosser på slutten av brettene. Det er ikke nødvendig for spilleren å bekjempe de vanlige fiendene for å komme videre, men dette vil gi spilleren høyere poengsum (og gjøre at spilleren får mindre kuler skutt etter seg).

Fiender

Spillet vil inneholde fiender på brettene. Disse fiendene har predefinerte spawnpunkter og flyr til et sluttpunkt (hvor den forlater vinduet), mellom disse to punktene vil fiendene skyte kuler på spilleren eller i en bestemt bane.

Bosser

Det vil være en boss i slutten av hvert brett som spilleren må overvinne for å komme seg videre til neste brett. At spilleren klarer å bekjempe en boss er også en god indikator på om spilleren behersker de evnene som skal være nødvendig for å komme seg videre til neste brett i spillet.

Kuler

Det vil implementeres kuler i spillet som spiller og fiender skyter mot hverandre. Vi ønsker også å implementere flere typer kuler som skytes. Sannsynligvis vil vi bruke en factory for å generere kuler for å øke effektiviteten av spill eksekveringen (Slipper og instansere kulene).

Stages

Vi vil implementere "stages". Hver stage vil inneholde forskjellig antall fiender og vanskelighetsgraden vil også stige ettersom du klatrer opp i stages. På slutten av en stage vil du møte på en boss som har skytemønstre avhengig av vanskelighetsgrad. Vi vil i vårt spill prøve å lage minst to nivåer, men vil om tid lage flere.

B - Mål

Vanskelighetsgrad

Vi ønsker å innføre to vanskelighetsgrader. Lett (easy) der spilleren har liv (HP) og får mer i form av drops fra fiender som dør (dropchance), Vi vil også implementere vanskelig (hard) der spilleren ikke får HP fra fiendtlige drops og vil dø på ett skudd.

Meny

Spilleren skal få tilgang til meny der personen kan redigere innstillinger som oppløsning, fullskjerm-modus og lyd.

Lyd

Vil innføre lyder til spillet som er laget til de forskjellige hendelsene i spillet som skyting av kuler, eksplosjoner og liknende effekter.

Bakgrunnsbilde

Spillet skal inneholde bakgrunnsbilde som passer spillet.

Metoden vi har tenkt å bruke for å gi effekten av evig lang bane er "swimming" lane prinsippet der du deler opp et passende bilde i swimming lanes. Du setter da disse sammen til et langt bilde om dette skulle være nødvendig. Vi kan også benytte oss av self repeating bilder. Vi har kommentert dette ytterligere på JIRA (DX-7)

Highscore

Vi vil innføre highscoreliste (toppliste) over sluttpoengene som spillerne samler opp ettersom de dreper fiender, bosser, plukker opp drops og ut ifra hvor langt de har kommet på brettene

C - Mål

Power-ups

Fiender dropper drops som spilleren kan plukke opp. Drops kan for eksempel inneholde regenerering av HP, kjappere skyting, bedre våpenskade og andre type effekter.

Stage select

Spilleren kan velge hvilken stage han vil starte på. Spilleren kan derimot ikke spille på en stage som ikke er oppnådd.

Special Effects

Vi ønsker å innføre animasjoner på prosjektiler, fiender, skip og bosser for å distrahere konsentrasjonen til spilleren.

D - Mål

Online Multiplayer

Noe vi kunne vi tenke oss å innføre er en samarbeidsmodus, slik at opp til to spillere kan spille med hverandre i spillet.

Online Highscore

Vi kunne også tenke oss å implementere online highscore der brukere som spiller spillet kan konkurrere om å få den beste poengsummen. Dette vil vi prøve å gjøre via et REST api.

Forskjellige typer skip

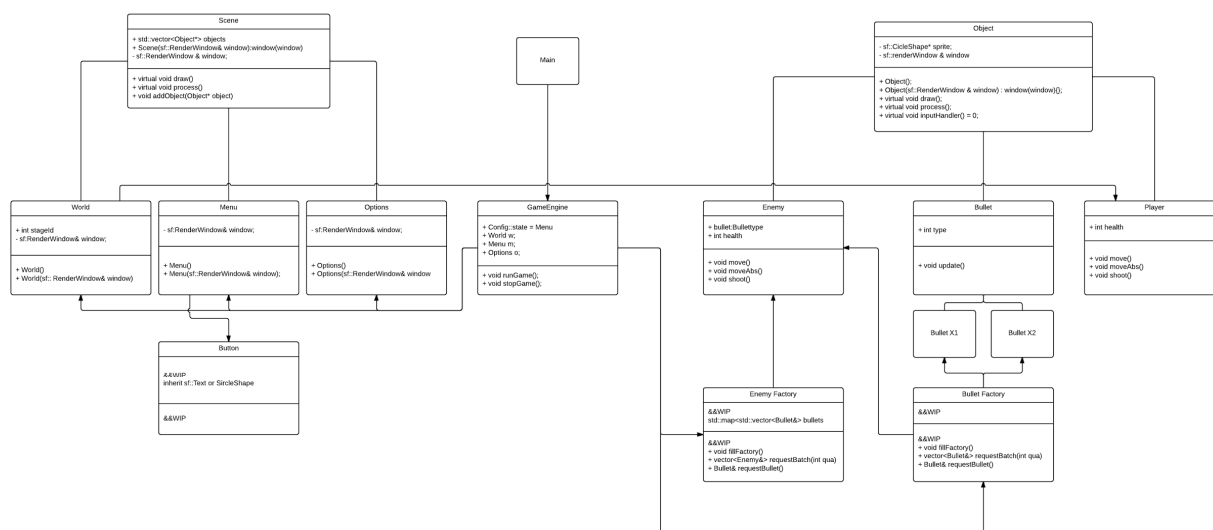
Vi vil gjerne legge til mulighet til å velge mellom to typer skip som spilleren kan spille med. Skipene vil ha forskjellige farger, våpen, osv.

Achievements

Det siste D-målet vårt er å legge til achievements for spilleren som de kan oppnå. Dette vil også bli en type bonus for å holde spilleren interessert.

Technical design / teknisk design

Klasse Diagram



(<http://bildr.no/image/UjJqdmr.jpeg>)

Bilder ovenfor beskriver X1337s elementære struktur, og vi har her bare inkludert det vi ser på som viktig i vårt prosjekt. Vi har planlagt alle klasser i detalj men har en god idé om hvordan vi skal utvikle disse, Vi vil selvfølgelig komme tilbake til de klassene som fortsatt har usikkerhet rundt seg på et gruppemøte, men vi har fokusert på de viktigste tingene for å få kommet i gang med utviklingen av produktet.

Spillet vårt vil bli basert på Superklassen Scene, Scene er en abstrakt klasse som alle Stages, Menyer og lignende arver fra. Dette gir oss et godt rammeverk for å håndtere like klasser på samme måte, i tillegg til at det minimerer redundans i klassene.

På samme måte vil alle tegnbare objekter arve fra Object som på samme måte gir et abstrakt grensesnitt for utviklingen av objektene. Det kan merkes at alle funksjoner i disse klassene ikke er pure virtual. Grunnen til dette er at store deler av de arvende klassene har en felles måte å bli håndtert på. Derfor har vi valgt og bare holde disse virtual.

Det vi kan se på som utfordring i vårt design er håndtering prosesseringen av Kuler på en optimalisert måte, Vi har så langt diskutert bruken av factory for å gjøre dette bedre. og vil da brukere BulletFactory klassen for dette.

Filstruktur

Den overordnede filstrukturen vi være på følgende måte

 objects	19.10.2013 23:42	File folder
 bullet.txt	19.10.2013 23:42	Text Document
 enemy.txt	19.10.2013 23:42	Text Document
 player.txt	19.10.2013 23:42	Text Document
 scenes	19.10.2013 23:42	File folder
 menu.txt	19.10.2013 23:42	Text Document
 options.txt	19.10.2013 23:42	Text Document
 world.txt	19.10.2013 23:42	Text Document
 gameengine.txt	19.10.2013 23:43	Text Document
 main.txt	19.10.2013 23:43	Text Document

Merk at dette bare demonstrerer filstruktur. Denne kan også differere fra det ferdige produktet.

Standarder

Noen av de standardene vi kommer til å følge er følgende:

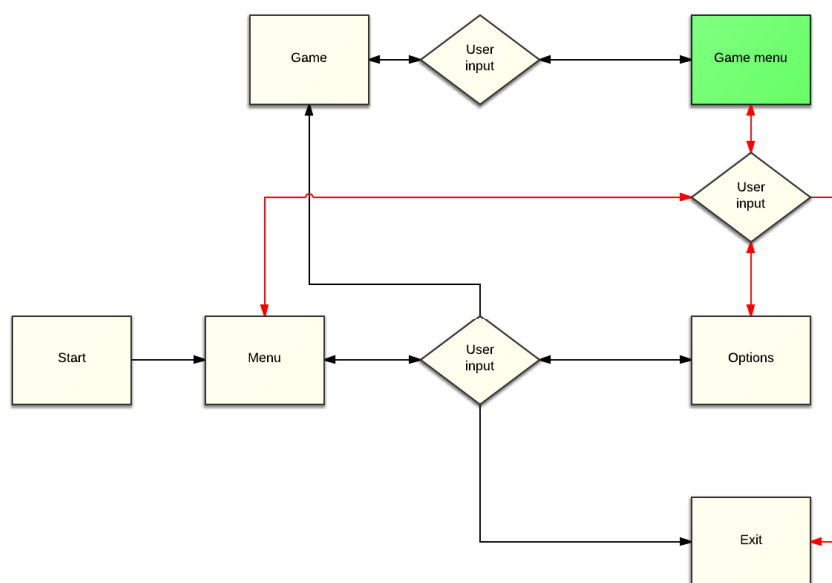
- camelCase variabelNavn
- const variabler skal være CAPS
- Referanse eller Pointer skal alltid være rett etter type. Ex: Type* varName
- Krøllparenteser skal alltid komme på andre linje etter metode.
 - void example()


```
{
}
```
- Foretrekk referanse over pekere der mulig
- Klassenavn skal alltid ha stor Forbokstav.
- GIT - Alle commits bør linkes opp til en issue sak, merk at dette ikke er nødvendig commiten ikke tilhører en bestemt issue. Eksempel på commit message er:
 - "DK-x - Implemented new functionality"

Produkt-design

Screen-flow

Vi tenker oss at når spillet starter opp, så kommer spilleren inn på en hovedmeny med tre valg: “Start”, “Options” og “Exit game”. “Options” presenterer spilleren med en ny meny hvor han kan velge forskjellige innstillinger for spillet (f.eks. vanskelighetsgrad), “Exit game” vil avslutte spillet, og “Start” vil ta spilleren til en ny meny hvor han kan velge bane (“stage”) å spille på.

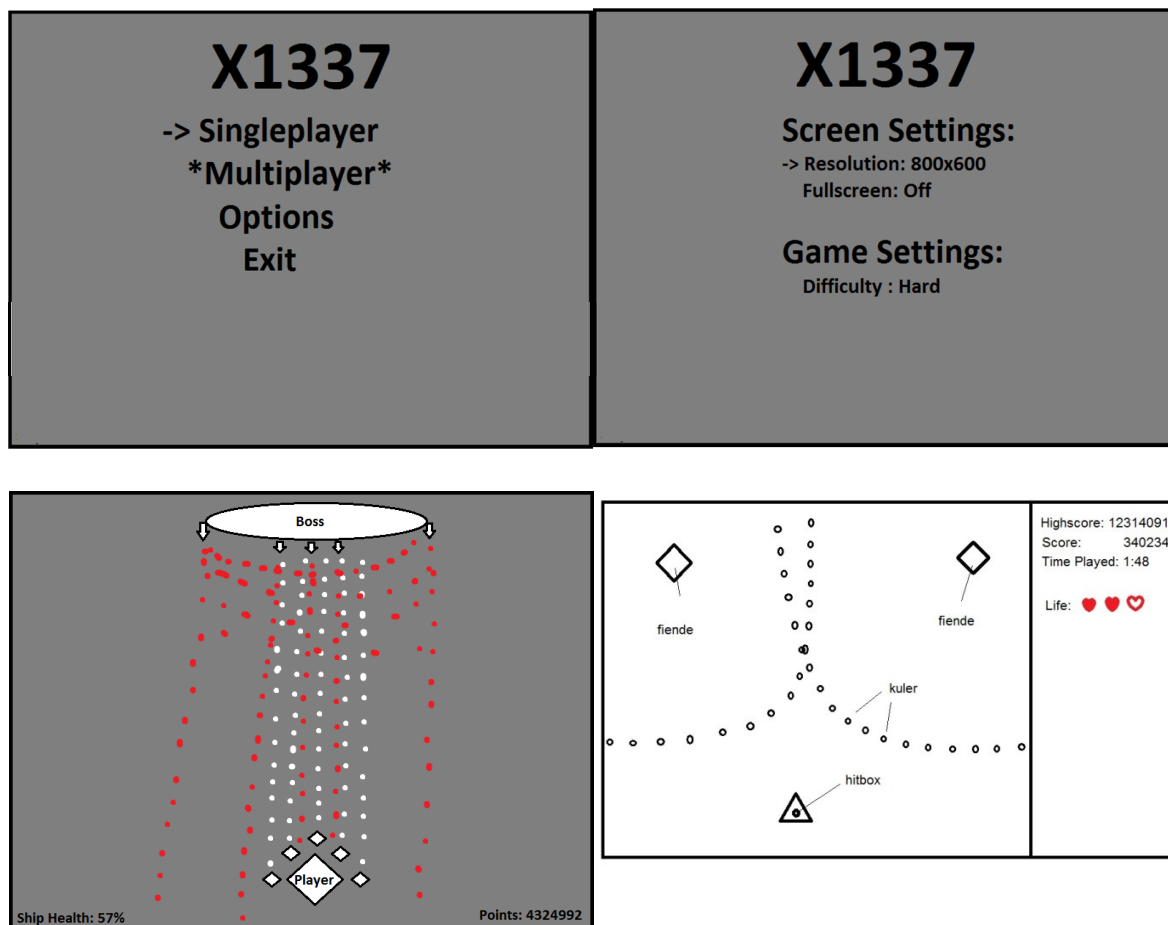


Til å starte med vil bare den første banen være mulig å velge, men etter hvert som man kommer videre i spillet vil flere baner bli låst opp. Fra Options-menyen vil man kunne navigere tilbake til hovedmenyen. Når man har valgt bane, vil spillet starte på den valgte banen. Herfra vil man kunne trykke på en knapp for å bringe opp en ny meny med valgene “Return to Main menu”, “Options” og “Exit game”. “Return to Main menu” tar spilleren tilbake til hovedmenyen, “Options” knappen bringer spilleren til den samme options-menyen som fra main menu (bortsett fra at spilleren kan navigere tilbake til menyen i spillet isteden for til hovedmenyen), og “Exit game” avslutter hele spillet.

GUI

Vi har enda ikke bestemt oss for et konkret GUI, så det er noe som vi kommer til å jobbe med å utvikle mens vi arbeider. Vi kommer sannsynligvis til å ta inspirasjon fra andre bullethell spill. Vi må i hvert fall ha med et område for spilleren å spille på, og et avgrensa område som holder informasjon som score, liv, osv.

In game gameplay



Det grå illustrerer bakgrunnen på spillet, dette kommer til å være et bilde in game og på menyen skal vi forsøke å legge inn en game demo som spiller seg selv/ eller en «godmode» som spilleren selv kan styre.

Vi er enda ikke helt sikker på game UI og vi har ved den anledning lagt til 2 design vi kan se aktuelle i vårt spill.

Verktøy

Vi vil benytte oss av en rekke forskjellige verktøy gjennom prosjektet. Følgende verktøy vil bli benyttet i prosjektfasen:

- JIRA som prosjekteringverktøy
- Git til versjonskontroll
- Visual Studio 2012 som utviklings IDE
- LucidChart webdiagram
- Google Docs og Microsoft Office for prosjektdokumentering.
- Dropbox

JIRA

er et prosjekteringsverktøy som brukes av utviklere for å planlegge utviklingen i et prosjekt med tanke på tidsbruk, arbeidsoppgaver og strukturell oversikt over hvilke utviklere som er på de enkelte oppgavene. Kanban stegene er delt opp i seks forskjellige trinn.

1. Visualisering av arbeidsmengde.
2. Begrensning av prosjektet.
3. Arbeidsflyt og dirigering av arbeidsoppgaver.
4. Fast arbeidsrammer.
5. Tilbakemelding av pågående deloppgaver til prosjektgruppen.
6. Felles problemløsning, forståelse og vitenskapelig stegorientering..

GIT

Gruppen bruker GIT i all utvikling på prosjektet. GIT er et versjonskontrollsystem som kan sammenlignes med SVN. Denne er hostet av UIA og kjører via atlassian stash. Med git får vi god oversikt over prosjektkoden og hvor mye arbeidsinnsats hver medlem legger inn. Vi får i tillegg muligheten til å bruke branches som gjør at feature development ikke forstyrrer main kode base før de er ferdig utviklet. Gruppen vil som mål prøve å holde kodebasen kompillerbar til enhver tid, men vil også gi hver gruppemedlem muligheten til å commit ukompilerbar kode ved å bruke identifikasjonskoden [CNA] = Compilation Not Available.

Visual Studio 2012

Visual studio blir er IDE som gruppen vil benytte seg av som støtter de språk-plattformene som vi kommer til å utvikle prosjektet i.

Boost

Boost er et sett med C++ biblioteker som gir støtte til lineær algebra, multithreading, bildeprosessering, regulære uttrykk, unit testing og pseudorandom nummer generering.

I vårt prosjekt vil vi bruke boost om vi ser det som nødvendig, det vil si at alle gruppemedlemmene vil ha boost tilgjengelig om behovet skulle vise seg.

LucidChart

LucidChart er et HTML5 online verktøy for tegning av UML diagrammer, flowcharts, mindmaps og chart maps. Dette verktøyet gjør det enkelt for gruppen å holde overordnet oversikt over utviklingen av klasser.

Google Docs & Microsoft Office

Vi bestemte oss for å bruke Google Docs til forprosjektoppgaven, og prosjektdokumentasjon, slik at alle kan delta på skrivingen samtidig, og at dokumentet også blir lagret trygt i nettskyen i tilfelle Murphys law slår inn. Microsoft Office blir også brukt sammen med Dropbox slik at vi har ekstra backup av prosjektet på alle maskinene. Google Docs brukes mest under utarbeiding av dokumentasjon, mens Office brukes til formatering når innholdet er klart.

Dropbox

Gruppen vil bruke Dropbox som backup for dokumentasjon som blir skrevet på Google Docs. Det vil også bli lagret kopi av prosjektet på Dropbox. Annet tilbehør som bilder, skjermdump og tekstdokument med ideer vil også bli lagret på Dropbox.

SFML

Simple and Fast Multimedia Library er et bibliotek for C++. Vi har valgt å ta dette i bruk i vår utvikling av X1337. SFML er et uhyrekjapt og lettvekt bibliotek som gir oss muligheten til å tegne opp 2d grafikk. Det er hardware akkselerert med OpenGL som også er cross compatible. SFML er delt opp i 6 forskjellige moduler. I disse modulene finner du:

- **System Module:** Håndterer bla multithreading om vi ønsker å benytte dette
- **Window Module:** Håndterer vindu og User Input
- **Graphics Module:** Inneholder klasser for shapes, sprites og mye mer.
- **Audio Module:** Lydhåndtering
- **Network Module:** Nettverksprogrammering I form av sockets.

Alle disse modulene kan bli brukt uavhengig av hverandre (med unntak av graphics og window module), Dette er noe av det som gjør SFML et fleksibelt og godt bibliotek for programmering av 2d grafikk.

Gantt-skjema

Gruppen har bestemt seg for å lage et Gantt-skjema i Microsoft Excel for å vise totale oversikt over prosjektet.