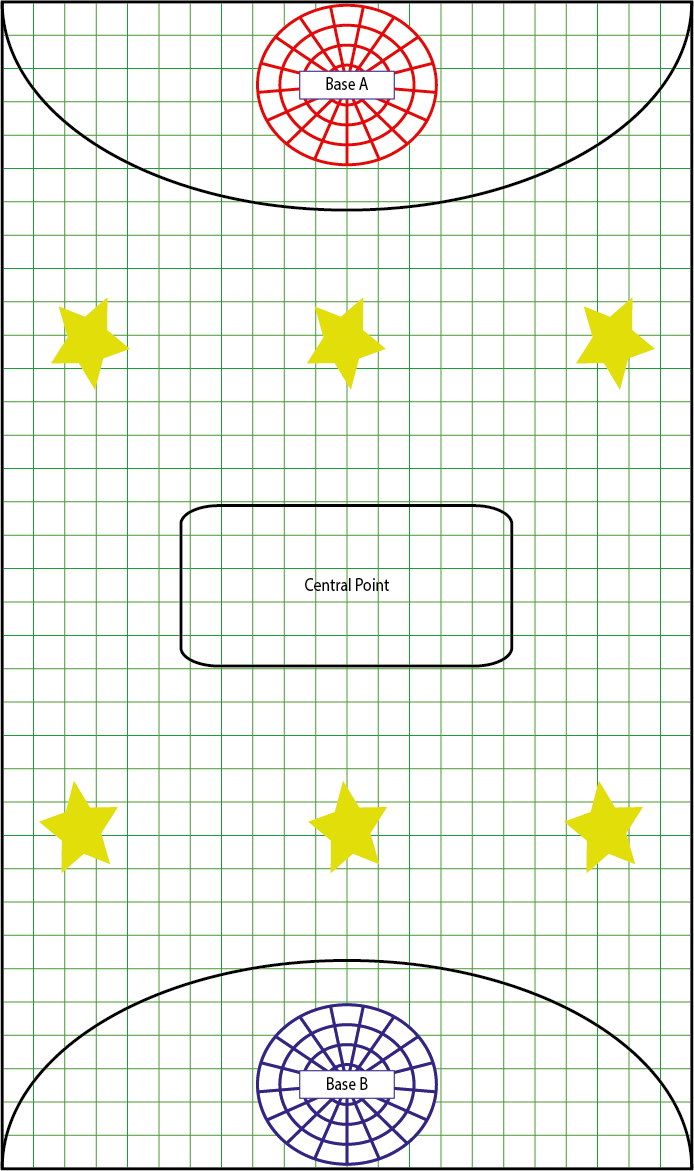
Big Little Wargame

Volledig adviesrapport



Team Fatal Error C1003

Joost Wagensveld 1664713

Remco Nijkamp 16……

Zehna van den Berg 16……

Waila Woe 16……

Docent: Gerald Ovink, , Joost Schalken-Pinkster, Wouter van Ooijen

**Versie: 0.1**

**Datum: maandag 2 november 2015**

# Samenvatting

In dit verslag is het project van het 2e jaar blok 1 vastgelegd. De opdracht van het project, een game geprogrammeerd in C++ en met gebruik van de SFML bibliotheek. De opdrachtgever is de Hogeschool Utrecht.

De opdracht is begonnen met het zelf bedenken van een computergame en het ontwerpen van een game design document. Hierna is een klassendiagram gemaakt en daaruit is de basis van de gamecode ontstaan.

Inhoudsopgave

[Samenvatting 2](#_Toc434245708)

[1. Inleiding 4](#_Toc434245709)

[2. Functioneel ontwerp 5](#_Toc434245710)

[3. Technisch ontwerp 6](#_Toc434245711)

[3.1. Architectuur 6](#_Toc434245712)

[3.2. Klassendiagram 6](#_Toc434245713)

[4. Realisatie 8](#_Toc434245714)

[4.1. Software 8](#_Toc434245715)

[4.2. Planning 8](#_Toc434245716)

[5. Evaluatie 10](#_Toc434245717)

[6. Conclusies en aanbevelingen 11](#_Toc434245718)

[6.1. Functionele conclusies en aanbevelingen 11](#_Toc434245719)

[6.2. Technische conclusies en aanbevelingen 11](#_Toc434245720)

[Appendices 12](#_Toc434245721)

[1. Klassendiagram 12](#_Toc434245722)

[2. Bronvermelding 13](#_Toc434245723)

[3. Game Design Document 14](#_Toc434245724)

[4. Technisch ontwerp 15](#_Toc434245725)

# Inleiding

In dit verslag word verteld hoe de opdracht, het maken van een game in de programmeertaal C++, tot stand is gekomen.

Het spel is een turn based strategy game. Maar wat het spel zo speciaal maakt is dat het meerdere mogelijkheden heeft tot het winnen van het spel.

Allereest is er natuurlijk de mogelijkheid tot het vernietigen van je tegenstanders basis. Als tweede mogelijkheid is er het voor langere tijd bezetten van een centraal punt.

De naam van ons team komt van de vele errors die tijdens het hele project langs zijn gekomen. De uitleg van Microsoft van deze error is:

Fatal Error C1003

Visual Studio 2015

error count exceeds number; stopping compilation

Errors in the program are too numerous to allow recovery. The compiler must terminate.

# Functioneel ontwerp

WAILA WAAR IS DIT STUK?

# Technisch ontwerp

## Architectuur

## Klassendiagram

### Game

### Player

### Building

### Unit

### Drawable

# Realisatie

Samenwerken in een project met andere leden is niet altijd gemakkelijk. Om dit in soepele banen te laten lopen kan men van verschillende hulpmiddelen gebruik maken. Een goede planning is noodzaak bij een project, zo weet iedereen waar die aan toe is. Ook zijn er programma’s beschikbaar voor het uitwisselen van werk. Door zulke programma’s te gebruiken, kan ieder project lid een goed overzicht houden van wat er al is gebeurd. En wat er dus nog moet gebeuren.

## Software

In het komende stuk wordt een korte uitleg gegeven van de software die tijdens het project is gebruikt.

### Visual Studio 2015

Visual Studio is gebruikt voor het schrijven van de code van van de game. Visual Studio is een programmeerontwikkelomgeving van Microsoft. Het biedt een complete set ontwikkelingstools om computerprogramma's in diverse programmeertalen. De taal die bij de realisatie van Big Little Wargame is gebruikt is C++. In combinatie met de bibliotheek van SFML, hierover verderop meer.

### SFML

Simple and Fast Multimedia Library (SFML) is een cross-platform software ontwikkel bibliotheek ontworpen als een eenvoudig interface voor verschillende multimedia componenten in computers([Gomila, 2015](#_ENREF_1)). Binnen dit project hier dan ook uitvoerig gebruik van gemaakt

### Softwares ideas modeler

Het klassen diagram is tot stand gekomen met behulp van Softwares ideas modeler. In dit programma kan men verschillende diagrammen maken. Verder is het ondervonden als een eenvoudig te gebruiken programma, wat redelijk voor zichzelf spreekt.

.

### Git

Om de code tussen de projectleden uit te wisselen is gebruik gemaakt van GitHub met als software Git Shell. Met deze software kan elk project lid zijn code uploaden naar een online Git repository, waar de rest van de groep deze weer vanaf kan halen. Ook kan de software van Git gebruikt worden om de codes van de verschillende teamleden samen te voegen. Zo word het tegelijkertijd werken op verschillende computers een stuk eenvoudiger. Ook zijn de verschillende documenten die voor dit project nodig zijn in de Git repository opgeslagen. Zo kan iedereen er bij en zijn ze uiteindelijk ook terug te vinden voor de klant.

### Microsoft Office

Voor dit project zijn meerdere programma’s gebruikt van het Office pakket. Word voor de verslaglegging en PowerPoint voor de presentaties en Outlook voor het onderlinge emailcontact.

### Google Drive

Om de documenten van dit project voor ieder project lid beschikbaar te maken is gebruik gemaakt van de online dienst Google Drive([Google, 2015](#_ENREF_2)). Hier kan elk project lid zijn documenten in kwijt. Hier staan alle documenten die betrekking hebben tot dit project. Documenten die ook voor de klant beschikbaar moeten zijn staan zoals eerder vernoemd ook in de Git repository.

### Doxygen

De documentatie van de code is tot stand gekomen door middel van het programma Doxygen. Dit programma kan na een aantal aanpassingen in de code, automatisch een verslag maken van de code.

## Planning

Hier moet ook nog iets…

### Scrum

Ohh ja, en de scrum.

# Evaluatie

In dit hoofdstuk wordt uiteengezet hoe dit project is verlopen en wat er allemaal goed en fout is gegaan. Hieruit zullen ook conclusies getrokken worden zodat deze fouten niet meer gemaakt zullen worden in de toekomst.

# Conclusies en aanbevelingen

In het volgende hoofdstuk worden een paar methoden van het besturingsprogramma toegelicht. Bepaalde fouten die zijn gemaakt zullen naar voren komen.

## Functionele conclusies en aanbevelingen

Het zou onzes inziens([OnzeTaal, 19-12-2013](#_ENREF_3)) beter zijn als er in plaats….

Wat zijn de conclusies en aanbevelingen op functioneel vlak

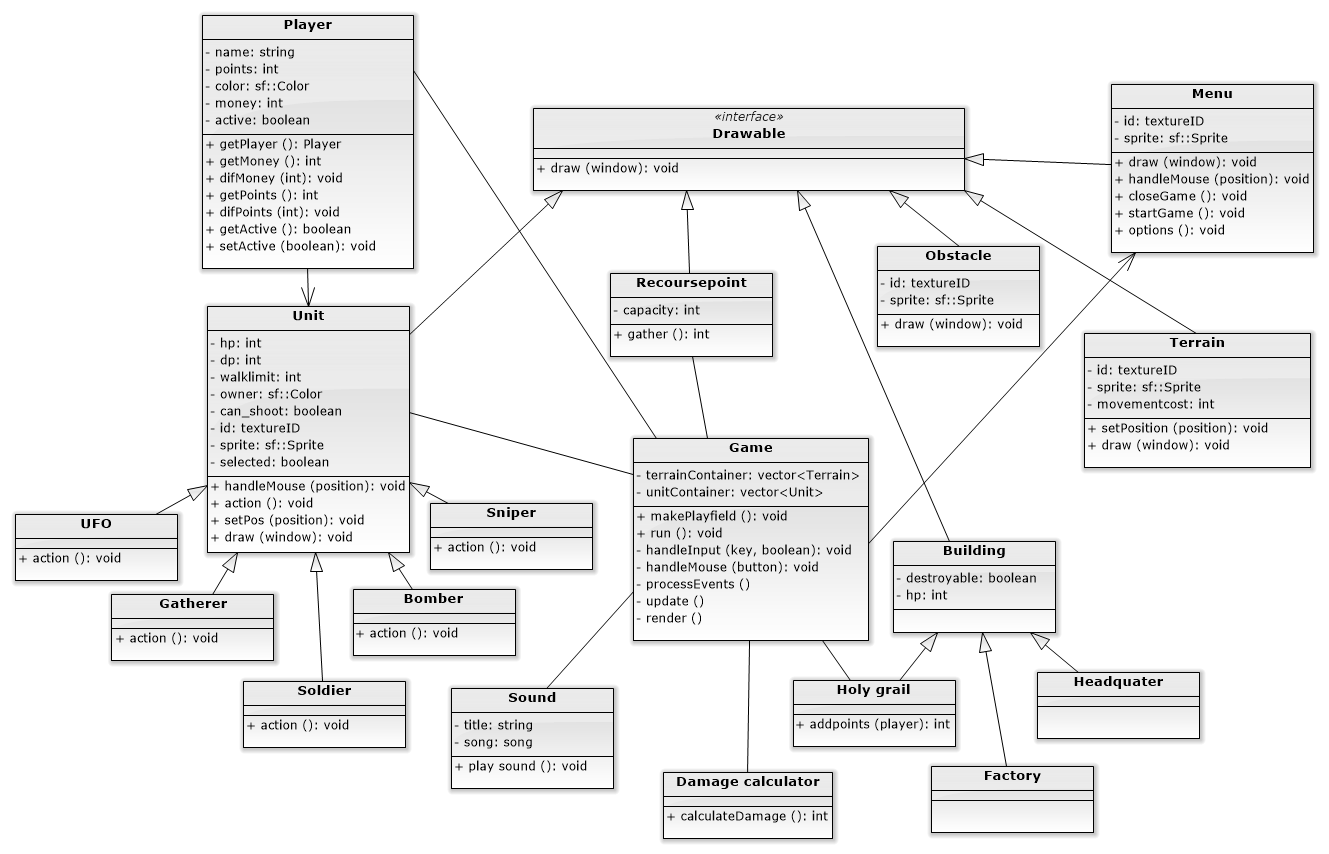
## Technische conclusies en aanbevelingen

Het zou onzes inziens([OnzeTaal, 19-12-2013](#_ENREF_3)) beter zijn als er in plaats….

Wat zijn de conclusies en aanbevelingen op technisch vlak?

# Appendices

## Klassendiagram



## Bronvermelding

Gomila, L. (2015). SFML (Simple and Fast Multimedia Library).

Google. (2015). Google Drive. from https://drive.google.com/

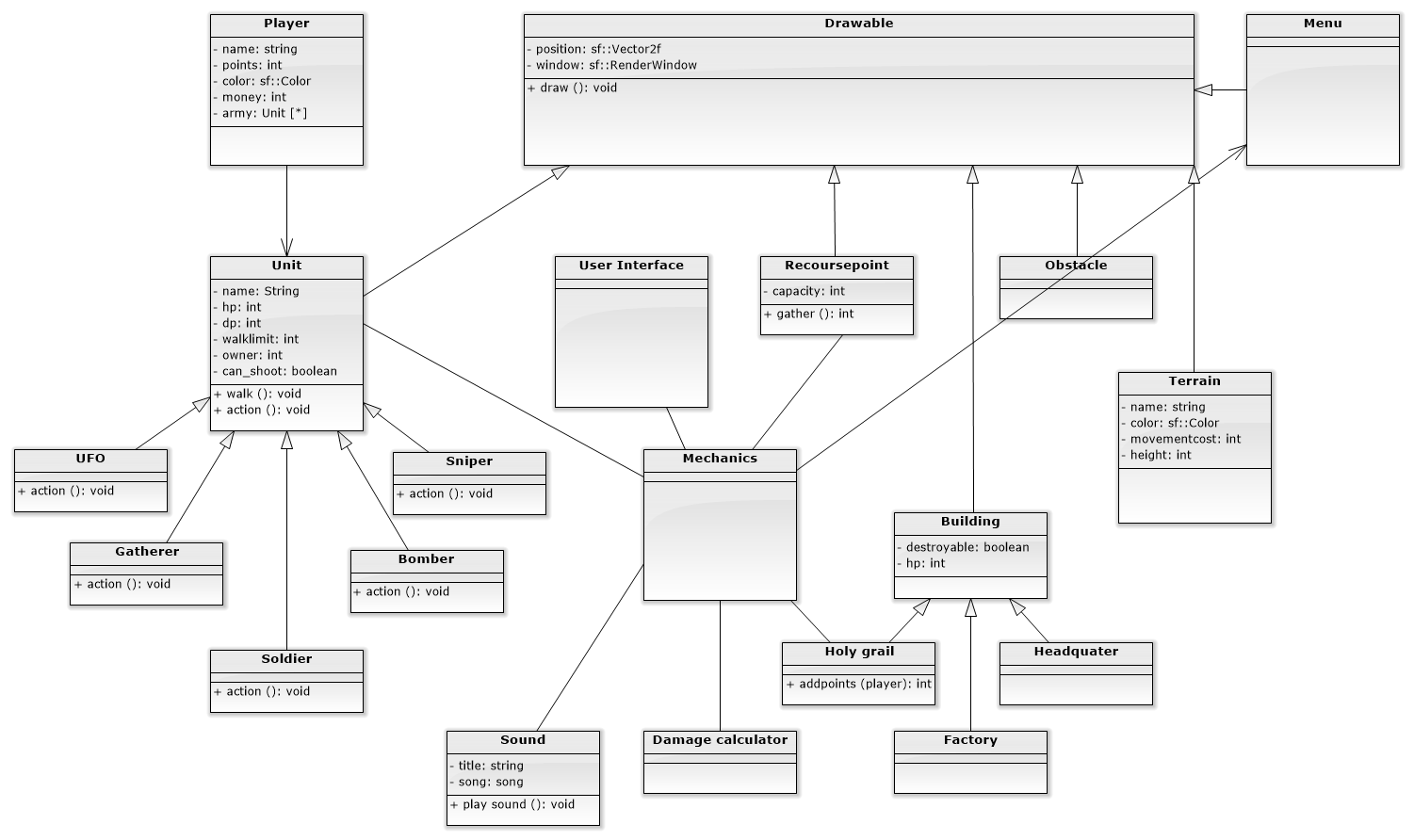
OnzeTaal. (19-12-2013). Genootschap Onze Taal. from https://onzetaal.nl/taaladvies/advies/ons-inziens-onzes-inziens

## Game Design Document

Of moet dit niet worden bijgevoegd???

## Technisch ontwerp

**Klassendiagram**



**Objecten:**

De klasse mechanics vormt als het ware het hart van het spel. Hier ligt een groot deel van verantwoordelijkheden die bij het spel komen kijken.

Er zijn een aantal klassen die objecten in het spel voorstellen. De categorieën hiervan zijn:

* unit
* building
* resource point
* obstacle
* terrain

Deze klassen zijn allemaal subklasses van een drawable superklasse.

Ook kunnen deze objecten zelf weer superklasse zijn, zo heeft een unit onder zich nog de verschillende soorten units (ufo, soldier, gatherer, sniper, bomber).

Omdat het spel met 2 spelers gespeeld wordt, is ervoor gekozen een player klasse te maken die zorgt dat er meerdere player objecten aangemaakt kunnen worden die de informatie van iedere speler onthoudt.

Erg belangrijk verder is nog de user interface. Deze zorgt voor de afhandeling van de interactie van de speler.

**Fileformaten**

* Menu file (bevat menu)
* Sound file (bevat alle geluiden van het spel)
* HUD file (evt onder menu / player)
* Player file (bevat player info)
* Terrain file (bevat speelveld en de verschillende tiles)
* Units file (bevat alle units)
* Buildings file (bevat alle buildings)
* Mechanics(controller) file
* Exceptions file (bevalt alle exceptions)

**Libraries**

* SFML/Graphics
* fstream
* iostream

## Scrumverslagen