

NAAM: **Giovanni Koolhoven**

KLAS: **PST1T**

VAK: **Software**

JAAR: **2017**



Inhoud

[leerdoelen 2](#_Toc503129807)

[Gebruikers 6](#_Toc503129808)

[Doelgroep 6](#_Toc503129809)

[Interactie gebruiker 7](#_Toc503129810)

[Functional requirements 8](#_Toc503129811)

[Schets 9](#_Toc503129812)

[Visual studio installatie onderhouden 10](#_Toc503129813)

# leerdoelen

1. Je kunt je eigen **Visual Studio-installatie onderhouden**.

* Door middel van de laatste updates te houden kan je visual studio gemakkelijk onderhouden. In het kopje: [Visual studio installatie onderhouden](#_Visual_studio_installatie) hier meer over.

1. Je kunt een **analysedocument schrijven** waarbij je voorafgaand aan een project een gepriotiriseerde opsomming geeft van **functionele eisen** aan een applicatie.

* Zie kopje: [Functional requierements](#_Functional_requirements)

1. Je moet vanuit een specificatie een C#-programma kunnen schrijven waarmee **tekstbestanden** kunnen worden ingelezen en worden geschreven.

* Ik heb een speciaal profiel applicatie gemaakt waardoor je een profiel kan maken voor het mario spel. Dit word opgeslagen en opgehaald in tekst bestanden

1x in uwp programma

1x in mario spel

1. Je moet op een correcte manier **excepties** in je C#-programma kunnen afhandelen.

* Ik heb try en catches gebruikt in mijn code. Bij mijn database haal ik zelfs de exceptie foutcode op

1x databasecon

1x titlescreen

1. Je bent in staat om vooraf te documenteren uit welke **klassen** je applicatie bestaat, en wat de eigenschappen, methoden en verantwoordelijkheden van die klassen zijn en deze grafisch in UML weer te geven.

Zie ontwerprapport, klassendiagram

1. Je kunt associaties (relaties) met **multipliciteiten tussen klassen ontwerpen** en deze weergeven in een UML-klassendiagram.

Zie ontwerprapport, klassendiagram

1. Je kunt in C# vanuit een specificatie constructoren programmeren en deze gebruiken.

Ik heb eigen classen gemaakt met eigen geprogrammeerde constructors.

1x character

1x enemys

1. Je kunt in C# binnen een klasse methoden programmeren met dezelfde naam en je kunt uitleggen wat *methodoverloading* betekent.

In mijn Enemy en Character class heb ik overload constructors gebruikt

1x characters

1x enemys

1. Je kunt programmeren met **private** en **public**.

Ik weet het verschil en ik heb geprogrammeerd met private en public

1x characters

2x enemys

1. Je kunt **property's programmeren** vanuit een gegeven specificatie.

In mijn classes heb ik gebruik gemaakt van proppertys om mijn private attributen op te halen.

1x characters

2x enemys

1. Je kunt **enum's** programmeren.

Ik heb gebruik gemaakt van enums en weet hoe ze werken en he je ze moet programmeren.

1x menubuttons

2x character

1. Je kunt vanuit een specificatie een programma schrijven dat werkt met de diverse functies van de **Windows GDI** (graphics).

Ik heb de kubus en iedereen kan schilderen gemaakt waar de graphics een grote rol speelt.

1x iedereen kan schilderen

1x kubus

1. Je kunt gebruik maken van bestaande methoden die **static** zijn en je kunt uitleggen wat het **static** keyword betekent.

Ik heb gebruik gemaakt van static. Meerdere malen in mijn mario project

1x inloggen

1x dbclassen

1. Je kunt programmeren met casting in de context van het **casten van een object** naar het gewenste subtype.

Ik heb gebruik gemaakt van casting met nadruk om casten van een object.

1x world.cs casten picturebox

1x enemycontroller form

1. Je kunt **programmeren met foreach** en lijsten van objecten.

Ik heb erg veel gebruikt gemaakt van foreachen door objecten en gebruik gemaakt van arrays

1x cntcharacters

1x chtenemys

1. Je stelt je professioneel op. Dit blijkt ondermeer uit je aanwezigheid, actieve deelname tijdens de lessen, een kritische houding en samenwerken met anderen.

* Momenteel ben ik alleen 12-12-2017 en 13-12-2017 afwezig geweest door zware omstandigheden thuis. De andere dagen aanwezig geweest en goed met de les meegedaan, ook veel klasgenoten geholpen en hierbij ook mijn docente geholpen met de les en verschillende dingen voor de klas uitgelegd.

1. Je kunt vanuit een gegeven **UML-klassendiagram relaties in C# programmeren** waarbij je gebruik maakt van List's (lijsten van objecten).

Ik heb arrays gemaakt van objecten vanuit mijn eigen gemaakte uml klassendiagram relaties

Goed voorbeeld: level

1. Je kunt het verchil tussen **value en reference types uitleggen** en er mee programmeren.

Ik heb gebruik gemaakt van values en reference types in mijn projecten

1. Je bent in staat om kleine applicaties te programmeren waarbij er een goede scheiding is tussen code in klassen en userinterface-code.

De overige opdrachten waren klein en hierbij kon ik gemakkelijk scheiding maken tussen code van klassen en userinterface-code

Groot pluspunt, mario code goede scheiding

1. Je kunt user interfaces voor applicaties maken met zowel de **Windows Forms designer als met de XAML designer.**

* Het project wat ik ga maken maak ik in windows forms. Hierbij wil ik een klein projectje bij maken waar ik ook laat zien dat ik xaml beheers. Dit word waarsch een klein projectje waar je je eigen “Profiel” in kunt maken.
* Kiezen van een avatar, naam etc. en dit word dan doormiddel van tekstbestandje ook naar de windows forms project verwerkt.

1x mario uwp programma

# Gebruikers

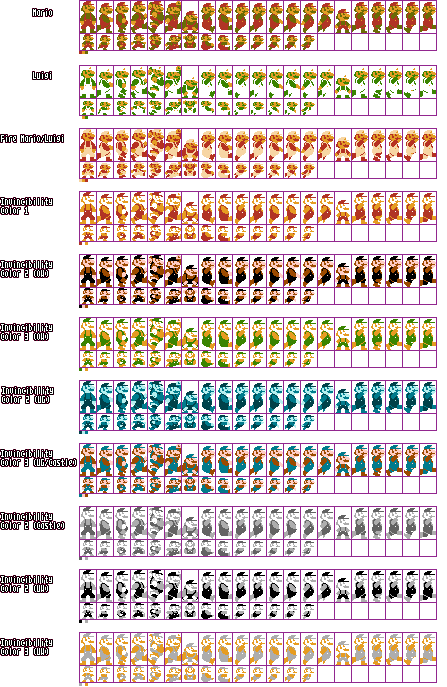
De applicatie word gemaakt voor gebruikers die in zijn voor een leuke game. De applicatie is een remake van het spelletje Super Mario Bros. De applicatie is daarom ook bedoeld voor vermaak. Door de applicatie kunnen gebruikers levels uitspelen en verder komen in het spel. Dit kan een gebruikers als een prettig gevoel aanvaarden. In interactie met gebruikers word meer verteld wat de gebruiker kan doen.

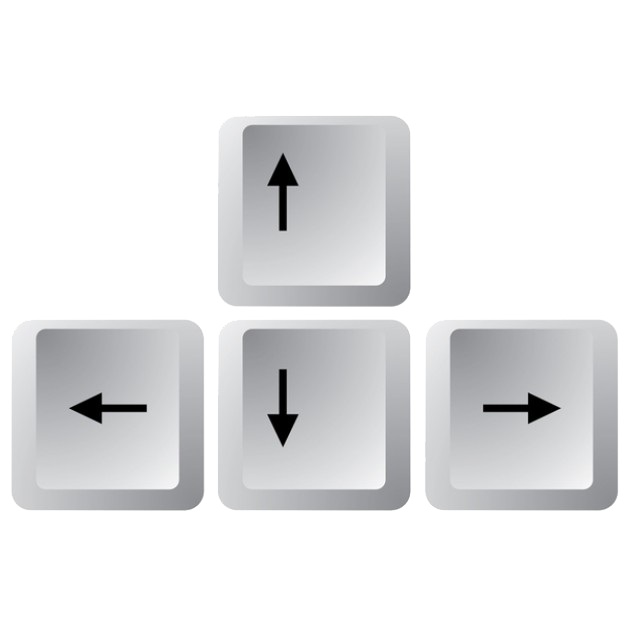
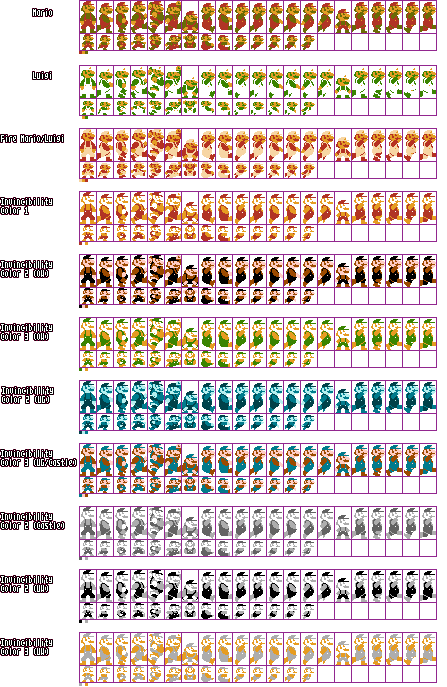
# Doelgroep

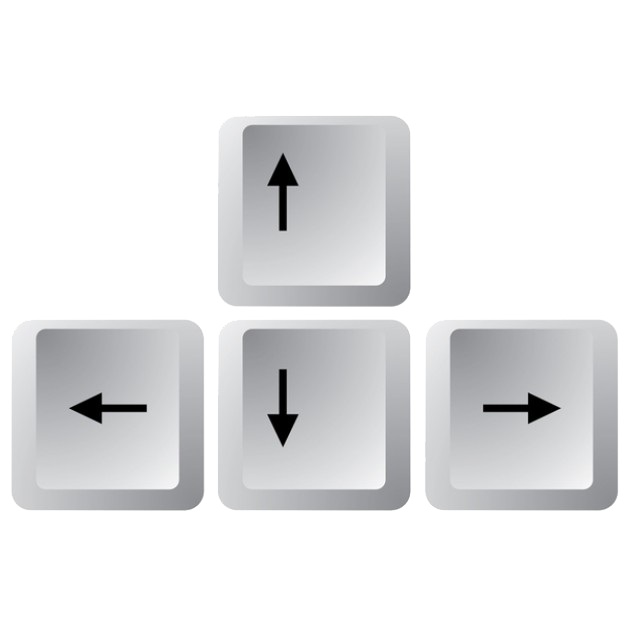
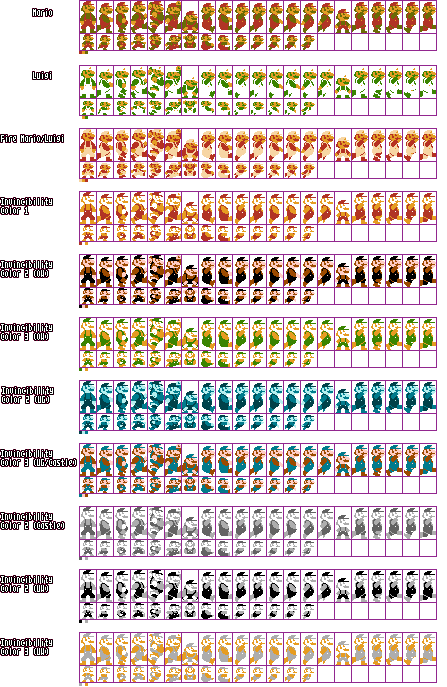
De meeste jongeren tussen de 7 en 16 spelen mario games. Maar mario is in principe voor jong en oud. Daarom is er geen specifieke doelgroep. Toch zouden de mensen die meerdere games hebben gespeeld meer geneigd zijn om de game te spelen. Ook kunnen bepaalde mensen al eerdere Mario games hebben gespeeld en die zullen ook meer in aanmerking komen voor de applicatie. Al met al is deze applicatie voor iedreen bedoeld die het leuk vind om bezig te zijn met een computerspel.

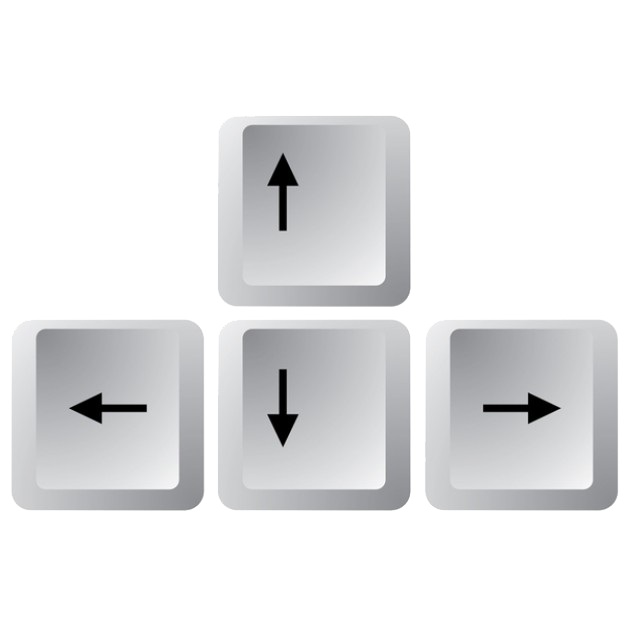


# Interactie gebruiker

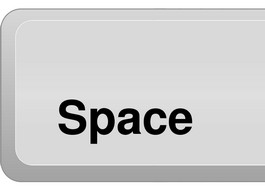
De gebruiker is in staat om zijn eigen profiel te maken. Denk hierbij aan een avatar naam etc kiezen. Ook kan de gebruiker het spel spelen. Dit kan gespeeld worden via de pijltjes toetsen. Hieronder is dit verder uitgebreid.

 De personage loopt naar links (achteruit)

 De personage loopt naar rechts (vooruit)

De personage springt.



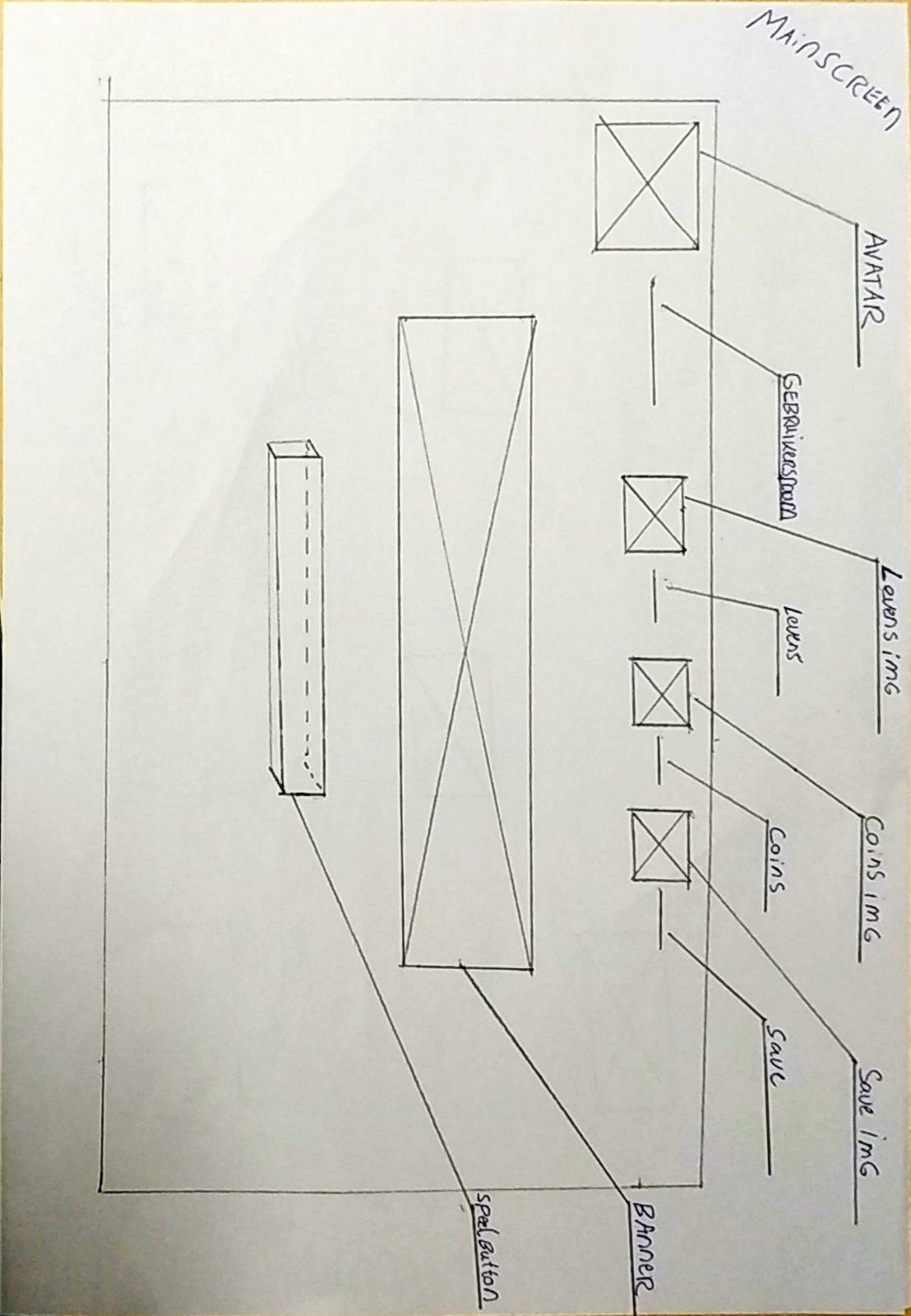
 Via de spatiebalk kan de personage vuur gooien. Hierbij moet hij dan wel de item daarbij hebben.

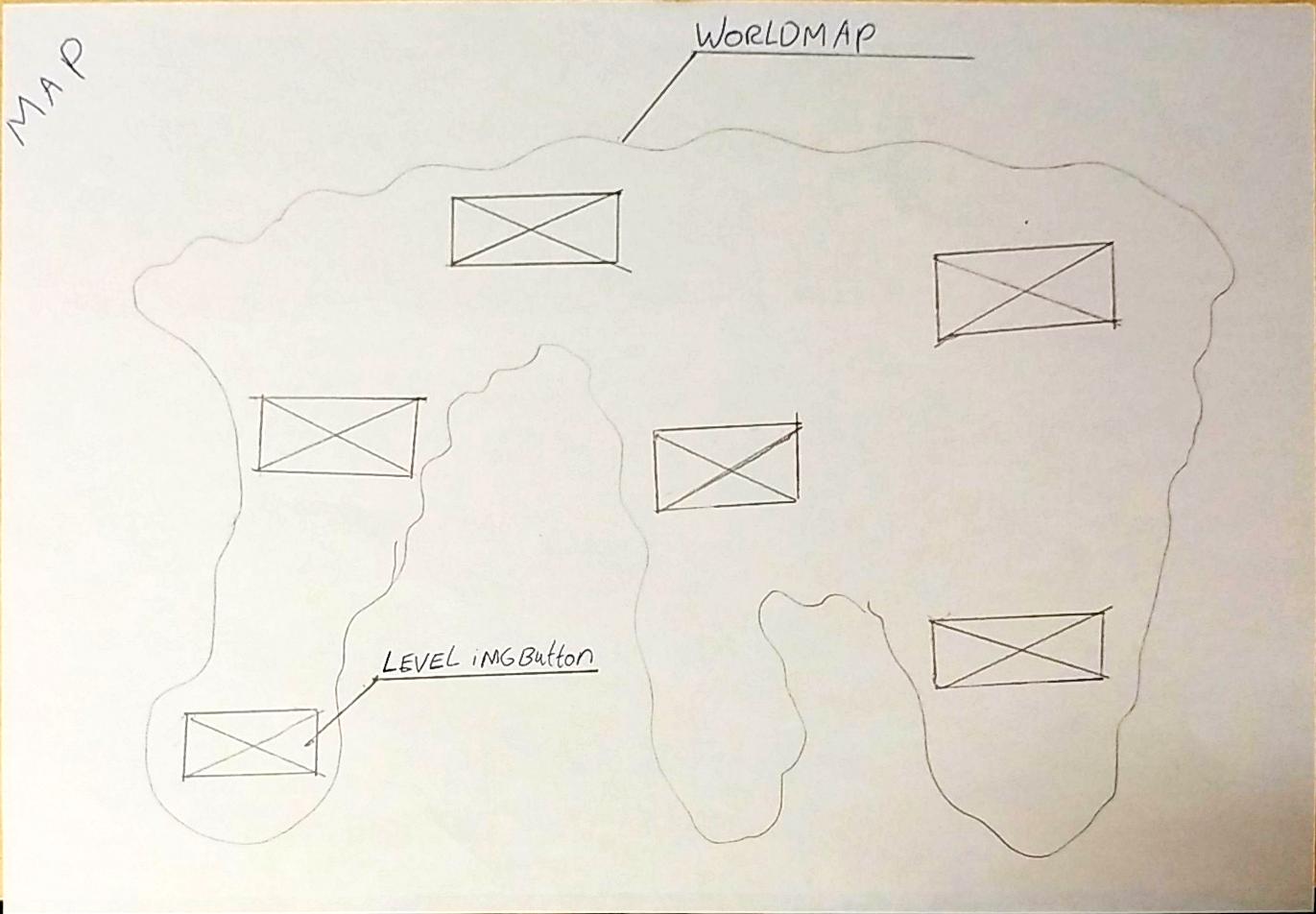
 via escape ga je terug naar het homescherm.

# Functional requirements

1. De applicatie moet gaan werken op Windows
2. De applicatie moet gemaakt worden in Windows forms
3. De applicatie moet een homescherm hebben
4. De applicatie moet een map(speel map) hebben waar een level gekozen
5. De applicatie moet een personage hebben die kan rondlopen
6. De applicatie moet een personage hebben die kan veranderen doormiddel van items
7. De applicatie moet een personage hebben die een aantal levens heeft
8. De applicatie moet een personage hebben die dood kan gaan waardoor er ook een leven verloren wordt.
9. De applicatie moet een aantal levels hebben.
10. De applicatie moet vijanden bevatten

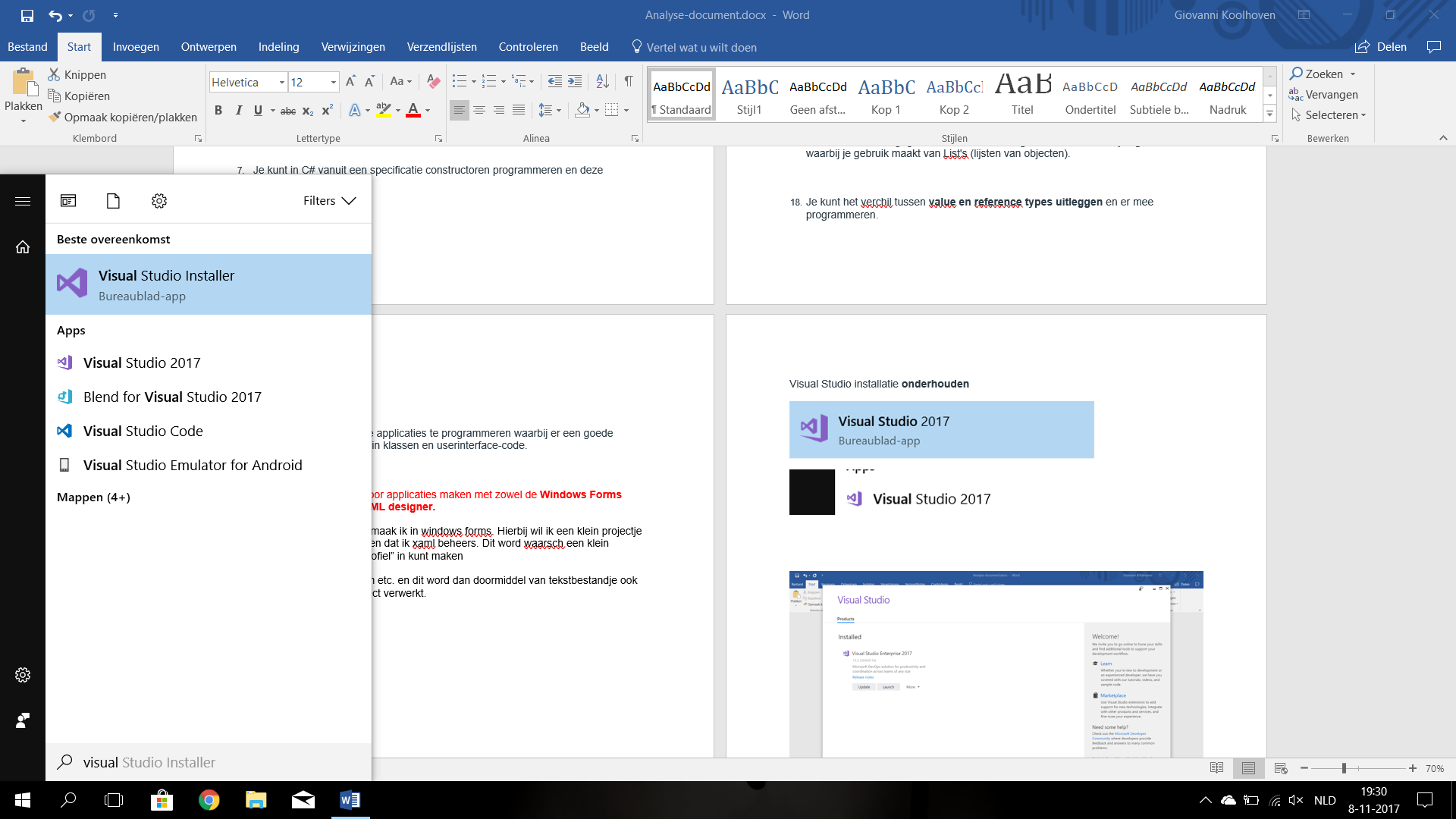
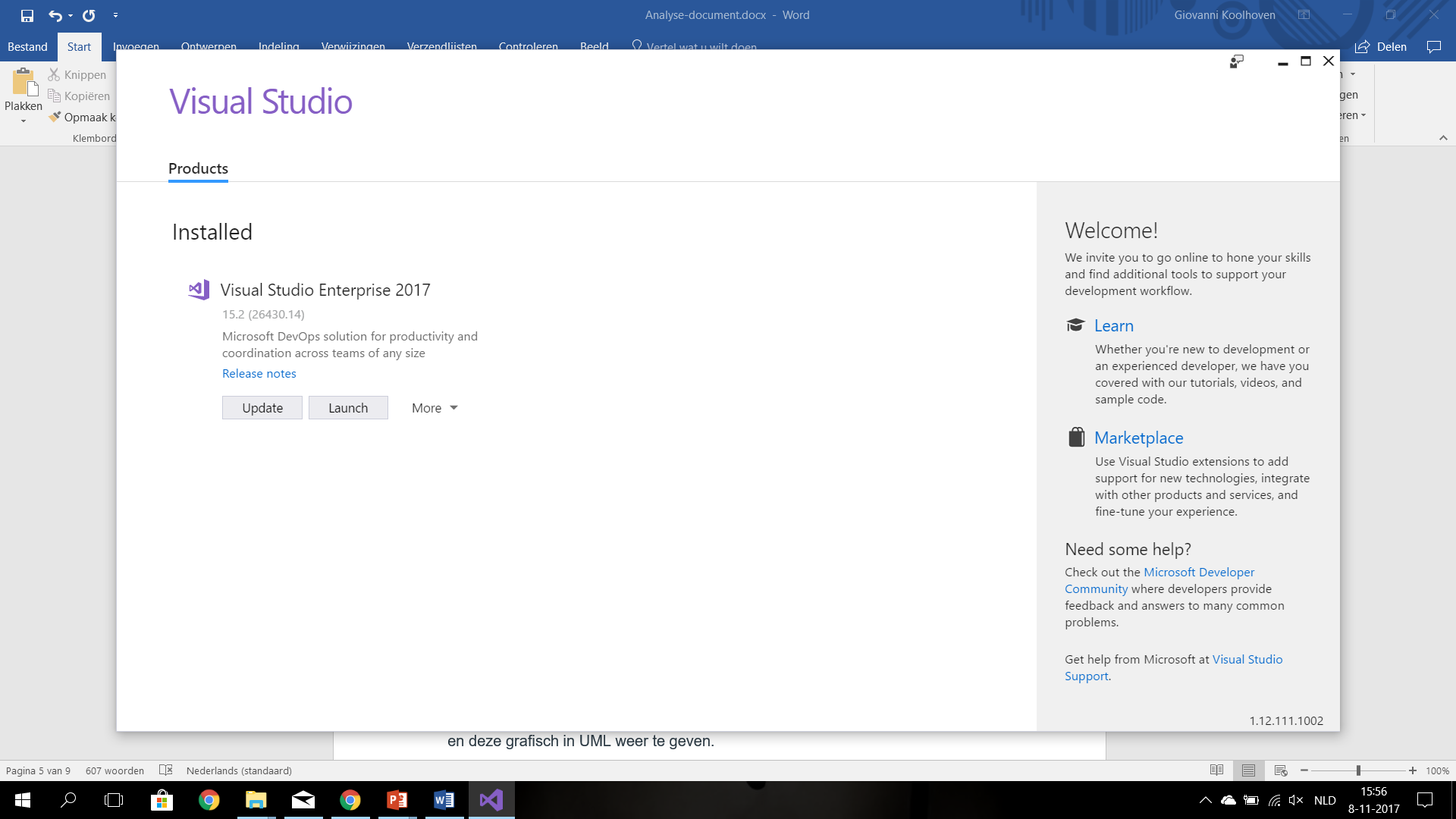
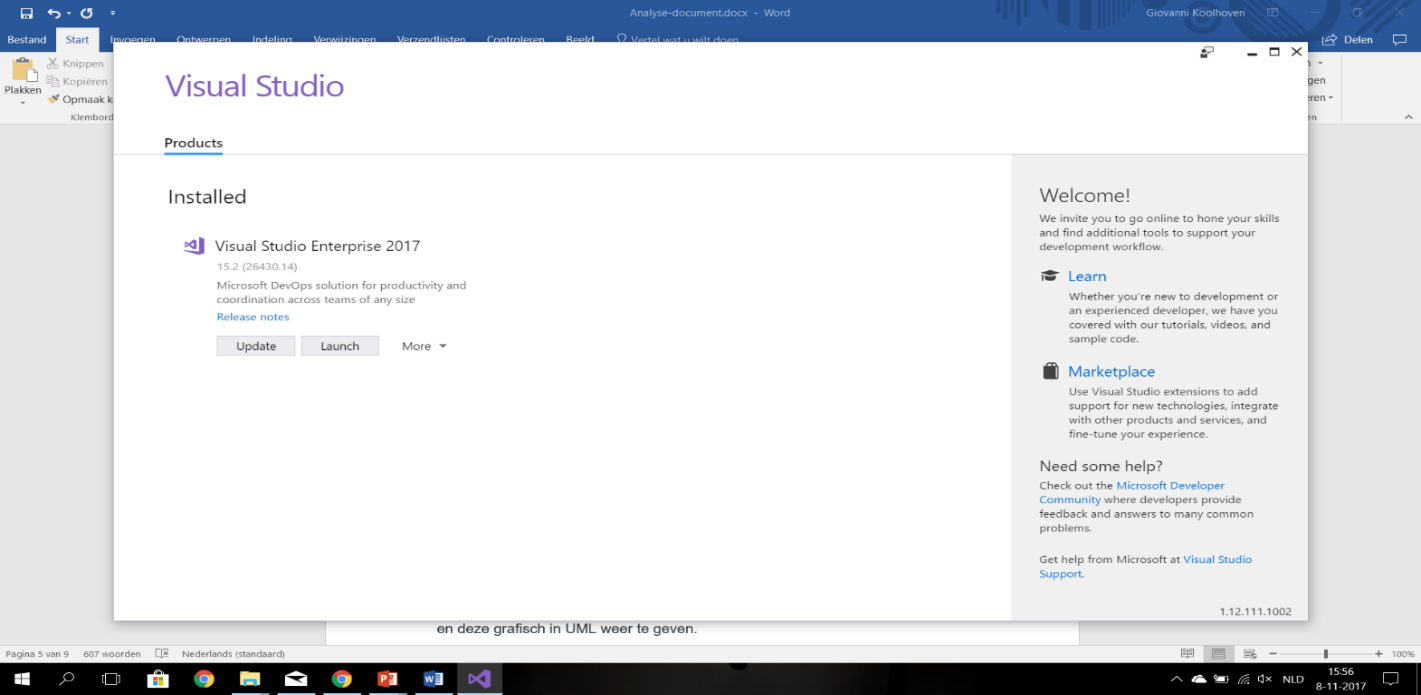
# Schets





# Visual studio installatie onderhouden

Via de setup van visual studio kan je visual studio natuurlijk installeren maar ook update. Ook kan je via de visual studio installer extra componenten toevoegen/downloaden als je die de eerste keer niet meegedownload hebt.



Ook kun je bepaalde updates en extenties installeren via Tools – extenties and updates. Daarna kan je via het volgende scherm links in het menu “updates” vinden. Hier kan je product updates en visual studio marketplace vensters vinden waar je bepaalde updates kan draaien.

