

Verslag: 3D in WPF

Inhoud

[1. Inleiding 3](#_Toc483253585)

[2. Keuze Monument 4](#_Toc483253586)

[3. WPF 5](#_Toc483253587)

[4. WPF Monument 6](#_Toc483253588)

[3.1 Begin 6](#_Toc483253589)

[3.2 sokkel 6](#_Toc483253590)

[3.2.1 opbouw 6](#_Toc483253591)

[3.2.2 probleem sokkel 6](#_Toc483253592)

[3.3 Basis 6](#_Toc483253593)

[3.4 trappen 6](#_Toc483253594)

[3.5 Pilaar 6](#_Toc483253595)

[3.6 Piramide 7](#_Toc483253596)

[3.7 Plakkaat 7](#_Toc483253597)

[5. WPF Camera 8](#_Toc483253598)

[4.1 basiscamera 8](#_Toc483253599)

[5.Bronnen 9](#_Toc483253600)

# Inleiding

Voor het vak Game programming wordt door de studenten een 3D-applicatie wordt gemaakt met WPF en XAML. Deze applicatie moet een monument naar keuze uitbeelden. Als laatste moet er ook nog een camerabeweging bij de applicatie die het monument in zijn volledigheid toont.

# Keuze Monument

Het in de presentaties gegeven voorbeeld van monument is het Atomium. Mij leek het echter dat velen dit gingen pakken, ik heb dus een ander monument gekozen: Het Wellington Monument in Dublin.

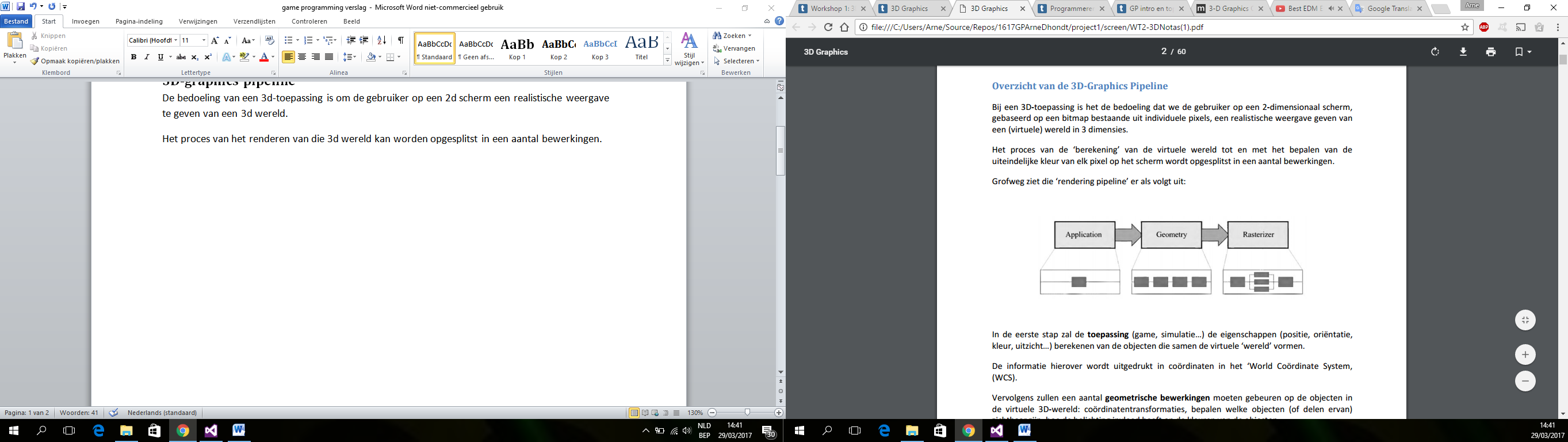
Dit monument is een obelisk. Deze obelisk heeft ook nog een kubus als sokkel voor de obelisk en een trap die naar de sokkel opbouwt. De voorkant van het monument heet ook nog een metalen plaat met daarop een afbeelding.

# WPF

## 3-D Graphics

In deze toepassing wordt een 3D figuur in 2D gezet. Om een realistische weergave te krijgen moet men rekening houden met verschillende eigenschappen zoals positie en belichting.

Het proces van het renderen van die 3d wereld kan worden opgesplitst in een aantal bewerkingen.



In de eerste stap zal de toepassing de positie en belichting eigenschappen opnemen en die uitdrukken in het World Coördinate System om ze daarna makkelijker te verwerken.

In de tweede stap als alle eigenschappen van alle objecten binnen zijn gekomen zal er berekend worden welke objecten zichtbaar zijn in he 2D vlak en hoe deze belicht worden.

In de derde stap zal men dan deze objecten in 2D uittekenen in rasters.

## Polygonen

De objecten worden opgebouwd aan de hand van veelhoeken. Hoe kleiner de veelhoek, hoe meer detail er in het object zit. Men gebruikt ook de simpelste vorm van een veelhoek: een driehoek. Per object worden er coördinaten bijgehouden die daarna worden gebruikt als hoekpunten voor deze driehoeken.

.

# WPF Monument

## 4.1 Begin

Als begin kreeg men buiten presentaties en een pdf met een intro ook nog een zip met 3 voorbeeldbestanden. Deze 3 bestanden waren een standaard driehoek, een form in WPF en een houten kubus die rond zijn as roteert. De laatste is de handigste om uit te leren wat we voor het project moeten gebruiken.

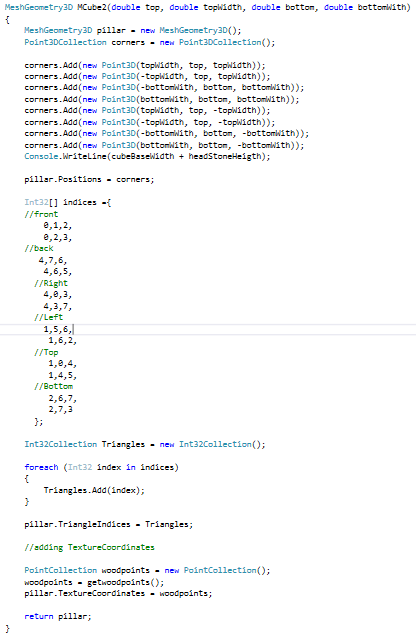
## 4.2 sokkel

### 4.2.1 opbouw

Als sokkel werd als basis de kubus gebruikt gegeven in het voorbeeld. Na uit te vogelen hoe het coördinatensysteem , dat gebruikt wordt om de hoekpunten aan te duiden, werkt kan men extra punten toevoegen die de uitsnijding van de kubus vormen.

Vervolgens kan men aan de hand van de extra punten de driehoeken maken die de uitsnijding vormen. Dit wordt ook op de manier gedaan die in her voorbeeld staat.

Nadat de driehoeken worden aangemaakt kan men de driehoek initialiseren in de Window\_Loaded methode die het canvas start. Om de 3D-vormen te kunnen zien moeten we hierbij ook een materiaal meegeven. Vanaf het begin wordt er in dit project een JPG meegegeven meet een marmer texture.



### 4.2.2 probleem sokkel

Om de figuur goed langs alle kanten te bekijken wordt de figuur om zijn as gedraaid. Hierbij wordt een probleem ondervonden. Sommige driehoeken zijn alleen zichtbaar langs een kant. Langs de andere kant zijn ze doorzichtig.

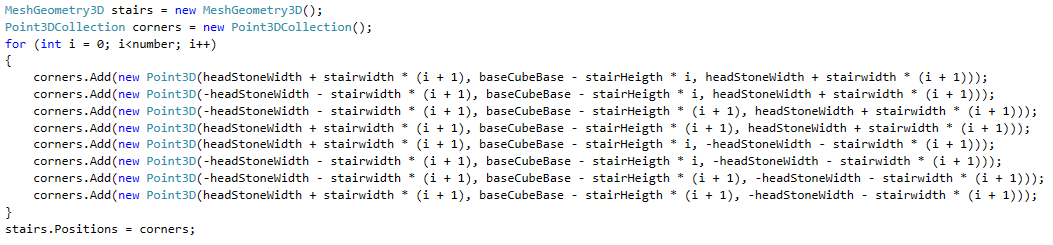
Een oplossing hiervoor word gevonden door de texture niet enkel mee te geven als material maar ook als backMaterial. Hierdoor wordt de figuur zichtbaar van alle kanten.

## 4.3 Basis

De sokkel krijgt nog een basis die gemaakt is uit een piramide zonder top. Deze figuur wordt gemaakt als een kubus waarbij de X-en Z-waarden van de top punten dichter naar het middelpunt liggen.

## 4.4 trappen

De trappen zijn grotendeels hetzelfde als het bouwen van een standaard kubus. De methode om deze figuur te maken wordt echter in een for-lus gestoken die bij elke nieuwe lus een trede lager gaat en verbreedt in de X-en Z-richtingen. Aangezien dat er van de sokkel is begonnen en die als basis op 0 is gezet, worden de trappen van 0 naar beneden gemaakt.



## 4.5 Pilaar

De pilaar die het belangrijkste deel van de obelisk vormt wordt op dezelfde manier gebouwd als de basis maar dan met een groter Y-waarde.

## 4.6 Piramide

De top van de pilaar is een kleine piramide. Dit kan ook samen in de pilaar worden gemaakt. Men moet maar één extra punt toevoegen om dit te doen. Hier is de piramide echter apart gemaakt. De piramide heeft maar 5 hoekpunten en aangezien de bodem bedekt wordt door de pilaar moeten enkel de hoekpunten verbonden worden met de top.

## 4.7 Plakkaat

Als laatste wordt er een bronzen plakkaat aan de voorkant van de sokkel geplaatst. Dit is weeral een soort kubus maar deze wordt niet met de Y-as als center geplaatst. Voor dit plakkaat wordt ook een extra texture toegevoegd.

# WPF Camera

## 5.1 basiscamera

De camera zorgt ervoor dat men het monument kan bekijken. Nu is het ook de bedoeling dat men het monument van alle kanten kan bekijken. Omdat de camera met vectoren verplaatsen ingewikkeld is, wordt de camera in dit project gewoon geroteerd rond de Y-as. Dit is tevens de centrale aas van het monument. Om dit te doen stoppen en starten wordt hieraan nog een button event toegevoegd.

# 5.Bronnen

http://www.i-programmer.info/projects/38-windows/273-easy-3d.html?start=1