

# P&O Computerwetenschappen: Schrijfopdracht 1

## Team Zilver

Ons team bestaat uit Bram Vandendriessche, Matthias Van der Heyden, Jef Versyck, Vincent Vliegen, Arne Vlietinck en Laura Vranken. Bram werd aangeduid als CEO en Arne zal de taak van CAO op zich nemen.

Om op een efficiënte manier het geheel te kunnen realiseren, wordt het team in twee verdeeld. De ene groep (Bram, Jef en Arne) buigt zich over het virtual testbed. Het andere team (Matthias, Vincent en Laura) zorgt voor de drone autopilot.

Een gedetailleerde planning kan teruggevonden worden in appendix A.

## 1 API keuze

Als API voor het genereren van 3D-beelden selecteerden we drie kandidaten: Blender, JMonkey-Engine en OpenGL.

Blender heeft als voordeel dat voorwerpen gemakkelijk aangemaakt kunnen worden door de gebruiksvriendelijke interface. Blender gebruikt echter Python. Hierdoor wordt er een moeilijkheid gecreëerd, namelijk de communicatie tussen Java en Python. Blender zou dan vanuit Java moeten worden gestart, wat in een omslachtig proces resulteert. Vooral omwille van deze laatste eigenschap zal er geen gebruik gemaakt worden van Blender.

JMonkeyEngine is gebruiksvriendelijker dan OpenGL, maar door gebrek aan uitgebreide communities wordt er niet voor geopteerd.

Onze keuze gaat uit naar OpenGL ondanks het ontbreken van een grafische interface. Dit nadeel kan wel gedeeltelijk omzeild worden door figuren te importeren vanuit Blender. OpenGL is reeds bruikbaar in Java en heeft bij problemen meer hulppaginas dan JMonkey.

## 2 Autopilot

De autopilot van de drone moet zijn positie ten opzichte van het doel kunnen bepalen aan de hand van twee ontvangen camerabeelden. Eerst wordt de diepte bepaald op basis van de brandpuntsafstand, de afstand tussen de zichtbare rode bol op het frame en het middelpunt van het frame (in pixels) en de afstand tussen de camera's (in meter).<sup>1</sup> Zie appendix B.

Om te vliegen, oriënteren we eerst de drone naar het doel. Hiervoor moeten opeenvolgend de rotaties (pitch, roll en yaw) onder een bepaalde hoek uitgevoerd worden. Vervolgens wordt naar het doel toe gevlogen via een combinatie van thrust en pitch. Om te kunnen controleren of de drone nog in de juiste richting vliegt, worden de berekeningen na iedere draai of na een bepaalde vliegafstand opnieuw uitgevoerd. Als pixel waarop de berekeningen gebaseerd worden, kiezen we diegene die het zwaartepunt van de gedetecteerde vorm weerspiegelt. Dit zwaartepunt bepalen we door de coördinaten van alle rode pixels op te slaan en hieruit de gemiddelde x- en y-coördinaat te berekenen.

## 3 GUI

De GUI zal gezamenlijk gemaakt worden, zodanig dat de vereisten van beide subgroepen verwerkt kunnen worden.

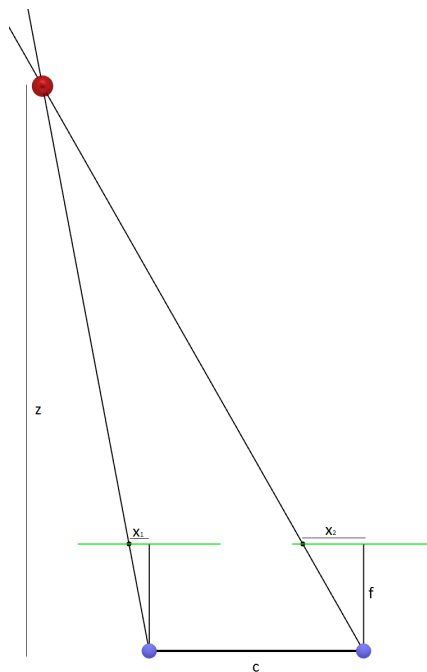
---

<sup>1</sup>Deze formule en de grafische weergave kan teruggevonden worden op: (NASA tech briefs : A Guide to Stereovision and 3D Imaging, <http://www.techbriefs.com/component/content/article/14925>, geraadpleegd op: 5/10/2016.)

## A Gantt chart

Semesterweek	2	3	4	5	6	7	8
Data	3/10/2016	10/10/2016	17/10/2016	24/10/2016	31/10/2016	7/11/2016	14/11/2016
Planning opstellen							
Git in orde brengen							
Wire protocol bekijken							
<b>Drone autopilot:</b>							
Berekening dieptezicht uit pixels							
Berekening hoeken naar gedetecteerde voorwerp							
Berekening middelpunt bol							
Programmeren berekeningen							
GUI							
Berekening voltooiingsgraad							
Testen							
<b>Virtual testbed:</b>							
Keuze platform							
Visualisatie basisomgeving, beweging en zwaartekracht							
Positie bepaling							
Collision detection							
GUI							
Problemen oplossen + reserve							
Testen							
Ontwerp/oplossingssuggesties verslag schrijven							
Tussentijdsverslag							
Tussentijdse demo							

## B Diepte-berekening



$$z = \frac{c \cdot f}{x_1 - x_2}$$