

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 generen van 3D-beelden zijn er drie kandidaten: Blender, JMonkeyEngine en OpenGL.

Blender heeft als voordeel dat voorwerpen gemakkelijk gegeneerd kunnen worden door de gebruiksvriendelijke interface. Daarnaast moet er voor Blender kennis zijn van Python, aan deze voorwaarde is niet door iedereen voldaan. Ook wordt er hierdoor een extra moeilijkheid gecreëerd, namelijk de communicatie tussen JAVA en Python. Vooral door 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 alles gecodeerd moet worden. Toch kan dit deels omzeild worden door figuren te importeren vanuit Blender. Deze optie zal dan ook gebruikt worden.

2 Autopilot

De drone autopilot moet zijn positie tegenover het doel kunnen bepalen aan de hand van twee gekregen 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).¹ Zie appendix B.

Om te vliegen, oriënteren we eerst de drone naar het doel. Hiervoor moet er opeenvolgend de bewegingen pitch, roll en yaw onder een bepaalde hoek uitgevoerd worden. Deze hoeken kunnen worden opgevraagd uit de interface. Vervolgens wordt naar het doel toegevlogen via een combinatie van thrust en pitch. Deze berekeningen worden na iedere draai of na een bepaalde vliegafstand telkens opnieuw uitgevoerd. Als pixel waarop de berekeningen gebaseerd worden, kiezen we de pixel 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

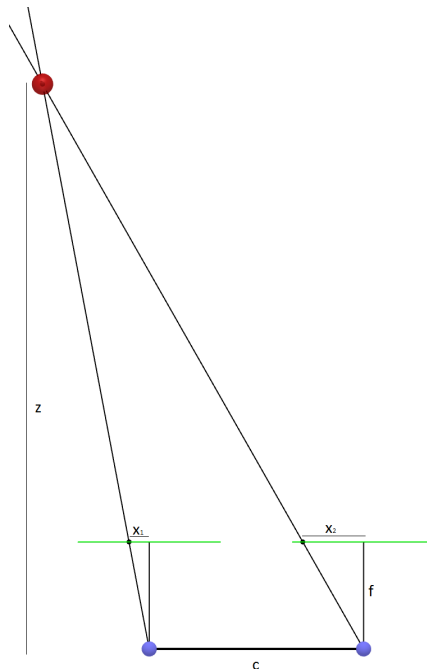
De GUI zal gezamenlijk gemaakt worden, zodanig de vereisten van beide subgroepen verwerkt kunnen worden.

¹Deze 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 | x | | | | | | |
| Git in orde brengen | x | | | | | | |
| Wire protocol bekijken | | x | x | | | | |
| Drone autopilot: | | | | | | | |
| Berekening dieptezicht uit pixels | | x | | | | | |
| Berekening hoeken naar gedetecteerde voorwerp | | x | | | | | |
| Berekening middelpunt bol | | x | | | | | |
| Programmeren berekeningen | | x | x | | | | |
| GUI | | | | x | | | |
| Berekening voltooiingsgraad | | | | x | | | |
| Testen | | | | | x | | |
| Virtual testbed: | | | | | | | |
| Keuze platform | | x | | | | | |
| Visualisatie basisomgeving, beweging en zwaartekracht | | x | | | | | |
| Positie bepaling | | | x | | | | |
| Collision detection | | | x | | | | |
| GUI | | | x | | | | |
| Problemen oplossen + reserve | | | | x | | | |
| Testen | | | | | x | | |
| Ontwerp/oplossingssuggesties verslag schrijven | x | | | | | | |
| Tussentijdsverslag | | | | | | x | |
| Tussentijdse demo | | | | | | | x |

B Diepte-berekening



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

PAGE LIMIT