

# P&O Computerwetenschappen: Schrijfopdracht 1

Team Geel

## 1 Testbed

Het virtual testbed stelt een wereld met een doelwit en eventuele hindernissen voor. Dit testbed genereert de nieuwe posities van de drone, rekening houdend met fysische wetten, en toont een 3D-beeld. Anderzijds berekent de autopilot aan de hand van wat zijn camera ziet, de oriëntatie van de vleugels. Op deze manier kan de drone het doelwit bereiken. De beelden die de autopilot krijgt komen van het testbed.

## 2 Motion & vision

Dit onderdeel bestaat uit een aantal subroutines. Een eerste routine creëert een wereld met een doel en geeft een beginpositie voor de drone. De tweede routine is de camera, die op basis van de positie en oriëntatie van het vliegtuig een afbeelding genereert van de wereld. Deze afbeelding wordt op het scherm getoond en aan een beeldherkenningsroutine doorgegeven. Momenteel lijkt het ons doenbaar om zelf een algoritme voor beeldherkenning te bedenken. Indien de latere opgaves het moeilijker maken lijkt OpenCV een degelijk framework om ons verder te helpen.

Een motion planning systeem gebruikt de informatie uit de beeldherkenning om het traject van de drone te bepalen en de nodige correcties in te plannen. Voor de eerste deadline kunnen we recht naar de gekleurde kubus bewegen maar in latere onderdelen gaan we hiervoor een geavanceerder algortime moeten uitdenken. De geplande correcties worden door een controle-routine uitgevoerd. De bewegingen van de drone wijzigen wat de drone ziet. Dit geeft de drone feedback op zijn acties en laat het toe verder bij te sturen. Los van de eigenlijke simulatie zijn er nog twee routines. Om te zien wat er gebeurt is verder een graphics engine nodig die de drone en wereld visueel voorstelt. Eventueel kan een GUI handig zijn om parameters makkelijk te kiezen en informatie overzichtelijk weer te geven.

## 3 Rendering

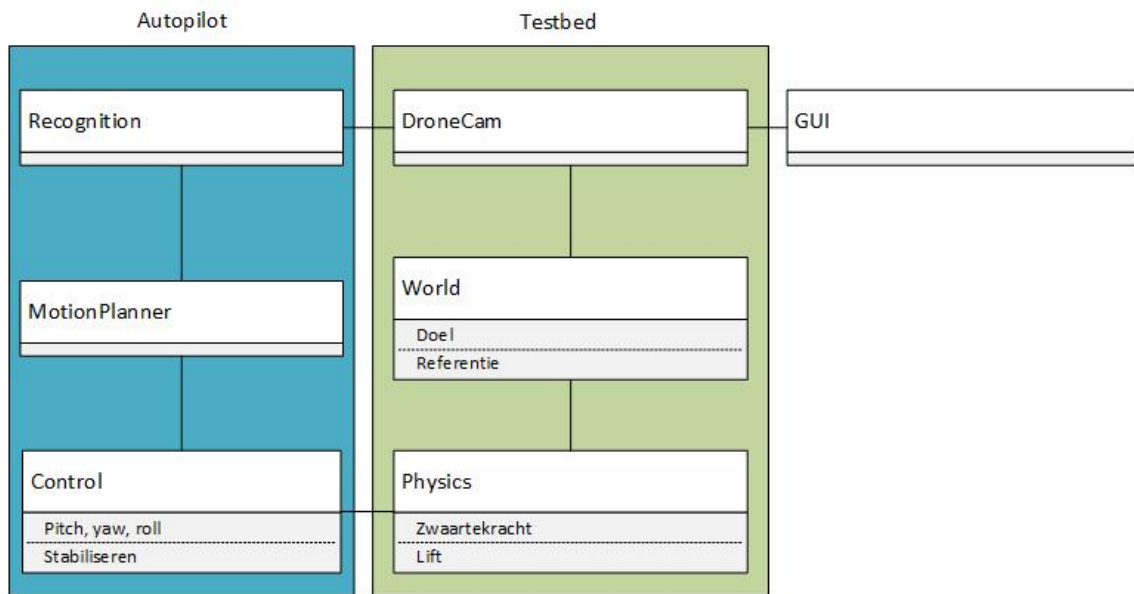
Het volledige programma is in Java geschreven. LWJGL zorgt voor het renderen van de graphics voor de gebruiker. LWJGL is immers heel licht en zeker krachtig genoeg voor de eenvoudige graphics waar het programma mee werkt. Andere opties zoals JavaFX en JMonkey lijken ons minder geschikt. JavaFX is trager maar makkelijker om mee te starten en JMonkey bevat te veel functionaliteiten die we niet nodig hebben waardoor we LWJGL's low level eigenschappen verkiezen.

Tabel 1: Huidige Verdeling

Onderdeel	Verdeling
<i>Physics Engine</i>	Flor, Florian, Tomas
<i>Visual Representation</i>	Arno (basics), Toon
<i>Image recognition</i>	Arno, Tomas
<i>Autopilot control</i>	Bernardo

Tabel 2: Eigen deadlines

Week	Visual representation	Image recognition	Autopilot	Physics
16/09	Voorstelling objecten	Degelijke research + hoog niveau	Manual control	Werkende draft versie
23/09	Basis implementatie van de wereld met camera input naar drone.	Basis werkend algoritme	Planning research + uitvoering	Nieuwe versie a.d.h.v visuele hulp
30/09	Tweaks en bugfixes	Polishing en optimalisatie	Finishing touches planning	Physics werkt volledig



Figuur 1: Voorlopig klassendiagram op zeer hoog niveau