PVA Automated Systems, Project codename Beezkneez

Mark Steijger
0938713

Kazimir Piek
0953725

Robert Karajev
0851997

Michael Francis
0963038

27 juni 2021

Inhoudsopgave

1 Inleiding

Dit document dient als een techniesch naslagwerk voor AutomatedBeezzzz project, hierin worden de verschillende aspecten die nodig zijn geweest voor het uiteindelijk complete product besproken. [1]

2 System/Subsystem Specification

2.1 Identification

Het volledige systeem bestaat uit de volgende subsystemen.

1. Camera
2. Server
3. Simulatie

2.2 System overview

Het doel van het systeem is het nabootsen van het gedrag dat bijen uitvoeren bij het verzamelen van eten. Hierbij wordt voor de bijen gebruik gemaakt van zowel hardware als gesimuleerde bijen. Deze bijen zullen vanuit hun start locatie, de korf, eten zoeken en hierna vervolgens andere bijen op de hoogte brengen van de locatie van deze voedselbron.

Het project zal uitgevoerd worden door de AutomatedBeezzzz projectgroep

2.3 Document overview

Dit document is geschreven om het gehele systeem van de drone simulatie die de bijen nabootst te omschrijven. Hierbij komt kijken de architectuur, eisen, designs en een testplan waar het systeem aan moet voldoen.

2.4 Referenced documents

Versies van dit document:

- Versie 1: 6-19-21

3 System/Subsystem Design Description

3.1 System wide design decisions

In dit hoofdstuk worden, zoals de tital al verraad, de design decisions behandeld. Hieronder vallen bijvoorbeeld input/outputs van het systeem, gedrag keuzes en andere onderdelen die voor het gehele systeem gelden

3.1.1 Camera

Aan de camera zit een aantal eisen. Het meest voor de hand liggende eis is dat de resolutie hoog genoeg moet zijn om bepaalde objecten te kunnen herkennen. Dit is niet de enige eis, zo moet de camera ook kleur kunnen herkennen en modulair zijn. Deze camera zal aangesloten moeten worden aan een microcontroller zodat de fotos die de camera maakt bewerkt kan worden. Dit betekent dat de microcontroller krachtig genoeg moet zijn om fotos op te vragen, verwerken en berekeningen erop uitvoeren. Ook moet hij vervolgens de verwerkte data doorsturen naar een centrale server. Dit betekent dat er een manier moet zijn op de microcontroller om deze data te verzenden. Dit zijn de hardware eisen aan de camera kant, als dit allemaal voldaan wordt is het mogelijk om de camera te gebruiken voor de simulatie van bijen.

3.1.2 Server

Omdat het een gecentraliseert systeem is worden de meeste acties bepaald door een centraal subsysteem, in dit geval is dat de server. Vanuit de server worden de drones aantgestuurd en wordt informatie van en naar de simulatie en camera verstuurd. Voor alle communicatie wordt een MQTT broker gebruikt.

Gedrag

Met gedrag worden de keuzes bedoelt die het systeem maakt, hiermee worden de bsis van het systeem bepaald.

1. Bij opstart word een drone naar de voedsel bron gestuurd.
2. Wanneer de drone deze locatie heeft berijkt zal hij trugkeren naar eht startpunt
3. De Drone danst om de locatie te delen met de andere drones
4. Meerdere drones worden gestuurd om het voedsel op te halen
5. Wanneer het laatste voedsel uit de bron is gehaald en alle drones binnen zijn worden een nieuwe scout gestruurd

Inputs/outputs

De server krijgt vanuit meerdere punten informatie binnen en verstuurd het ook naar meerdere subsystemen. Deze inputs en outputs worden uitgebreided in het hoofdstuk Interface Design Decisions.

Input Camera informatie

Input Simulatie informatie

Output Drone gedrag

Output Simulatie informatie

3.1.3 Simulatie

De simulatie is een eenvoudig manier om een overzicht te krijgen over het gehele project. Dit komt omdat het net als met de echte server en drone, samenhangt met de rest van het geheel. Zo neemt de simulatie als input de coordinaten van de drones. Net als de echte server verwerkt de simulatie de coordinaten om uiteindelijk als output signalen door te geven aan de gesimuleerde drones. Deze signalen zijn de commandos die de drones uit moeten voeren. Dit gebeurt allemaal op een hoge snelheid gezien de simulatie op een sterke Windows laptop afgebeeld wordt. De limiet hieraan is dus de snelheid van de ingekomen locatie data. Om deze data veilig te ontvangen wordt er een MQTT server gebruikt die vergrendeld is met inloggegevens. Gezien de simulatie zelf lokaal op een PC is, zijn hier verder geen beveiligingsrisicos aan gekoppeld. Omdat het een simulatie is, wordt er een groot belang gehecht aan toegankelijkheid van data. De beste, en soms wel enige, manier om een simulatie te testen is door gebruik te maken van verschillende variabelen en states waar een simulatie zich in kan bevinden. Zo kan een simulatie gebruikt worden om een zo realistisch mogelijk gesimuleerde omgeving weer te geven.

3.2 System architectural design

3.2.1 System components

Hier wordt een opsomming gegeven van alle hardware en software componenten gegeven,

Software Camera

Software MQTT Broker

Software simulatie

Software Server

Hardware Picam

Hardware Crazyflie

3.2.2 System architecture

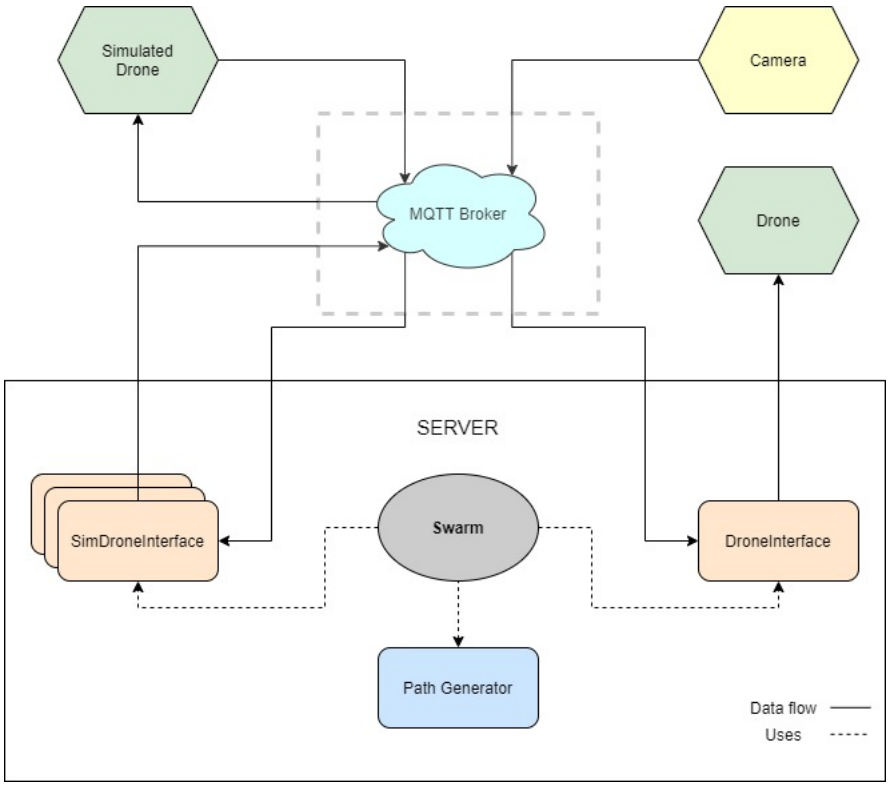
Diagram met hardware en software componenten van het gehelen systeem en de relatie tussen deze onderdelen

Camera

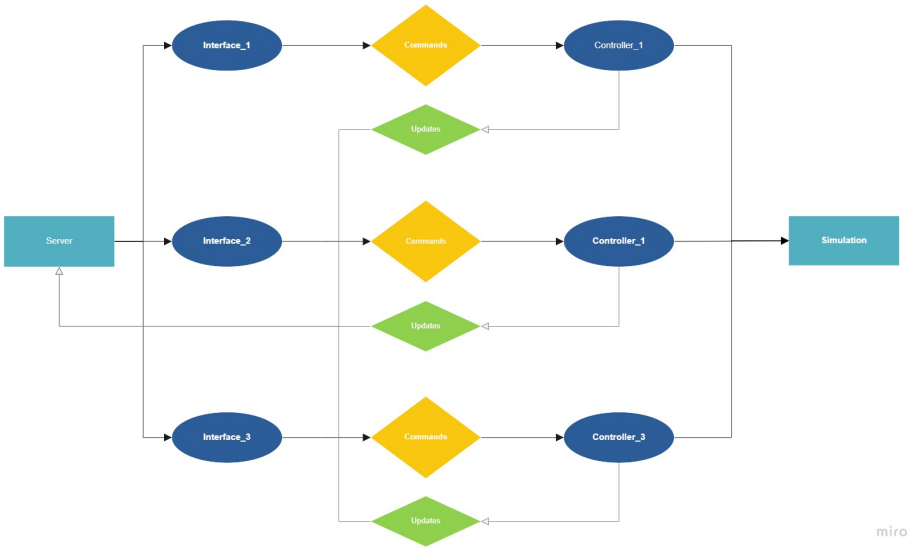
Aangezien het project inhoudt dat een aantal bijen gesimuleerd wordt, is het nodig om hun zintuigen ook te simuleren. Dit gebeurt door middel van een camera aan een raspberry pi. Hier zullen uiteraard eisen aan zitten, maar zullen in de SRS vermeldt worden. Om deze camera draadloos te laten werken is er ook een batterij pack nodig.

Server

Hieronder is de relatie te zien tussen de verschillende onderedelen van de Server. Hierin zijn ook de componenten meegenomen waar de Server mee communiceerd buiten zichzelf. Op deze manier is een overview van het volledige systeem weergegeven. De eerste communicatie begint wanneer de Camera begint met communiceren naar de MQTT broker.



Simulatie



Het diagram hierboven weergeeft de communicatie tussen de simulatie en de server. In de simulatie wordt het gedrag van bijen met behulp van 3 ontworpen drones gesimuleerd. Om dat mogelijk te maken zijn voor de input en de output de volgende ontwerp keuzes gemaakt:

INPUT:

1. De input voor de simulatie is altijd een opdracht naar een controller. Deze opdracht is een actie die een drone moet uitvoeren waarbij een positie meegegeven wordt naarr de volgende verplaatsingsstap. Dit wordt mogelijk gemaakt door te communiceren tegen een opgebouwde interface. Deze interface communiceerd dan vervolgens tegen een controller die een gesimuleerde drone bestuurd.

OUTPUT:

1. De output bij deze simulatie is verplaatsing van de gesimuleerde drone naar een geplannde positie.
2. Daarnaast is er ook een andere continue output van de simulatie. Dat is een bericht die afkomstig is van een controller. Een controller stuurt om de bepaalde tijd een bericht naar de server met de informatie over de positie van de bestuurde drone.

3.3 System architectural design

Uiteraard is het project een samenhangend geheel. Dit betekent dat er onderdelen zijn die afhankelijk zijn van andere onderdelen. Deze onderdelen zullen vermeld worden.

System components

Diagram met hardware en software componenten van het gehelen systeem en de relatie tussen deze onderdelen ¡ goeie, moeten we doen

Zoals op de afbeelding gebeeld wordt, is de camera afhankelijk van een raspberry pi. Dit komt omdat de beeld verwerkt moet worden door een microprocessor. Zo is er ook een samenhang tussen ...

4 Software Requirement Specification

In dit hoofdstuk word een opsomming en uitbreiding gegeven van de systeem requirements. Hieronder vallen onder andere de modus/states waarin het systeem zich kan bevinden en de communicatie die plaatsvind om het systeem te laten werken.

4.1 requirements

Camera

Aan de camera zit een aantal eisen. Het meest voor de hand liggende eis is dat de resolutie hoog genoeg moet zijn om bepaalde objecten te kunnen herkennen. Dit is niet de enige eis, zo moet de camera ook kleur kunnen herkennen om objecten van de achtergrond te onderscheiden en modulair zijn. Deze camera zal aangesloten moeten worden aan een microcontroller zodat de fotos die de camera maakt bewerkt kan worden. Dit betekent dat de microcontroller krachtig genoeg moet zijn om fotos op te vragen, verwerken en berekeningen erop uitvoeren. Ook moet hij vervolgens de verwerkte data doorsturen naar een centrale server. Dit betekent dat er een manier moet zijn op de microcontroller om deze data te verzenden. Dit zijn de hardware eisen aan de camera kant, als dit allemaal voldaan wordt is het mogelijk om de camera te gebruiken voor de simulatie van bijen.

Server

De server dient als het middelpunt van het gehele systeem, hierdoor zijn er hoop requirements die hieraan gebonden zijn. Allereest zijn er twee interfaces waarvan de server gebruikt maakt om de communicatie in het systeem te regelen. Voor het communiceren tussen de componenten van het systeem, de simulatie en de camera, wordt er gebruik gemaakt van een MQTT broker. Hiermee kunnen verschillende soorten berichten verzonden worden binnen het systeem en de juiste ontvanger kan op deze berichten subscriben om de relevante informatie te ontvangen. Voor de server geldt het dat deze zowel verstuurd als ontvangt. Hier zal verder op worden ingegaan in de interface paragraven van dit hoofstuk. Verder regelt de server ook alle communicatie naar de drones toe, ook hierop zal later uitgebreider worden ingegaan. Om de drones verschillend gedrag te laten uitvoeren voor de verschillende situaties die zich kunnen voordoen, wordt er gebruik gemaakt van verschillende modes waar deze drones zich in kunnen bevinden. Deze modes worden behandels in de paragraaf required stats and modes.

Simulatie

De simulatie is voor een deel een reflectie van de server, doordat deze ook de drones aanstuurd. Het processen van de data gebeurt op de server, maar de commandos worden vervolgens via MQTT doorgestuurd en uitgevoerd door de simulatie. Deze relatie word verder op in gegaan in internal interfaces. Verder heeft de simulatie eigen statussen die hij bijhoud die niet nodig zijn voor de hardware drone besturing om bij te houden.

4.2 Required states and modes

Camera

In het geval dat het systeem actief is, is ook de camera en de server actief. Aangezien de drone constant zijn locatie moet weten binnen het systeem, is het niet mogelijk om het gehele systeem idle te maken. Er is alleen een verschil tussen de berichten die worden verstuurd direct na de opstard en de berichten daarna. Allereerst word er namelijk de dimensies van de grid gestuurd en de locatie van de voedselbronnen. Alle berichten hierna zijn de locatie van de drones en de objecten daaromheen.

Server

Zoals eerder genoemd in de requirements, maakt de server gebruik van modes om de drones hun gedrag te laten uiten. De verschillende modes waarin de drone zich kan bevinden zijn, waiting, scouten, dansen en gatheren.

Waiting Een wachtende drone kan naar de scout status worden veranderd, wanneer hier nog geen andere drone mee bezig is. Als er al een drone aan het scouten is dan zullen de andere drones wachten totdat deze drone aankomt om de locatie door te geven, aangegeven door te dansen.

Scouten Een scoutende bij gaat opziek naar een voedselbron. Omdat er gewerkt wordt met object detectie is er voor gekozen om de drone direct naar de bron te laten vliegen. Er is geen meerwaarde in het rond laten vliegen van de drone wanneer het vinden van het doel al gebeurt is voor de vlucht. Bij het scouten wordt er door de drone naar het einddoel gevlogen en weer terug om de locatie door te geven in de volgende modus.

Dansen Wanneer de drone terug bij de hive is zal hij een dans uitvoeren, waarmee de bijen normaal gesproken de locatie en potentie van de voedselbron door communiceren. Door dit te doen worden de andere bijen naar de gather modus gezet en wordt de locatie doorgegeven

gatheren Wanneer de drones aan het gatheren zijn, zijn ze naar de voedselbron aan het vliegen totdat deze leeg en terug naar de korf totdat de voedselbron leeg is. Wanneer de het laatste voedsel uit de bron is gehaald, worden de drones opnieuw in waiting gezet.

Simulatie

De simulatie krijgt zijn commandos binnen nadat deze zijn uitgevogled door de server. Om deze reden maakt de simulatie geen gebruik van eigen modus, maar deelt hij deze met die van de server. Om deze reden kan er voor de modus naar de server modus gekeken worden, om te achterhalen wat de drones hun modus is binnen de simulatie. wel worden er statussen bij de gisimuleerde drones gebruikt om de huidige status van de drone bij te houden. Deze statussen zijn takeoff, moving, landing en of de drone aan staat of niet.

4.3 System internal interface requirements

Camera

De camera communiceerd door middel van een MQTT broker, hier worden meerdere dingen door naar buiten gecommuniceerd. Allereest wanneer het systeem opstart worden de buitenste rand en de etens objecten verstuurd naar de server. zodat deze regegistreerd kunnen worden. Tijdens het draaien van het systeem worden coordinaten van de objecten om een drone heen verstuurd samen met zijn eigen locatie. Hiermee kan de drone zorgen dat deze de obstakels om zich heen ontweikt.

Server

De server ontvangt de gegenereerde data van de camera doormiddel van de MQTT broker, vervolgens wordt deze informatie doorgegeven aan de relevante drones. Een combinatie van informatie en commandos door naar de draaiende simulatie gestuurd via de MQTT, zodat de simulatie ook weet waar de objecten zijn en heen moeten. Hiernaast worden de drones ook aangestuurd door de server, hiervoor wordt de 2.4Ghz radio antenne gebruikt.

Simulatie

Net als de server moet de simulatie vliegen van drones regelen, communicatie voor de locatie van deze objecten gaat om deze reden ook hetzelfde. De simulatie ontvangt commandos die de drones moeten uitvoeren en informatie van van waar de objecten zich bevinden via de MQTT.

4.4 Design and construction constraints

De grootste beperkingen binnen het project is het budget. Ondanks dit is het mogelijk geweest eromheen te werken en een kostenanalyse op te zetten die positief uitkomt. Verdere beperkingen zijn aantal drones die geleverd zijn, oftewel het beschikbare hardware. Hierdoor zou er een mogelijke compromis gesloten moeten worden om een aantal drones te simuleren. Ook is er niet volledige kennis over een systeem als deze voor elke groepslid, waardoor er kennis opgedaan moest worden en uiteindelijk niet genoeg tijd overbleef. De kennis met tijd balans is dus een grote beperking binnen het project.

5 Software Test Plan

De test plan zal worden opgesplitst in een software-matige kant en een hardware-matige kant. Zo kan voor ieder testomgeving omschreven worden wat er gebeurt binnen de omgeving, en de resultaat van de testen.

5.1 Test environment

Software

Om de testen op te zetten is het natuurlijk nodig om voor de drones bijbehorende software te maken. Dit betekent dat er een vorm van aansturing moet zijn, oftewel een interface om de drones mee aan te sturen. Deze interface bestuurd de benodigde commandos naar de individuele drones. De server draait een python server om de interface te runnen. Het is mogelijk om dit te doen op een laptop die Windows draait. Deze code wordt via python ook verbonden met een MQTT server, die dan verbonden is met een raspberry pi of Windows laptop waar een camera aan hangt. Hierop wordt code uitgevoerd om beeld te verwerken. Dit gebeurt ook in python. Ten slotte is er nog een simulatie aanwezig van twee of drie drones die op een scherm afgebeeld worden in WeBots. Dit gebeurt op een Windows laptop die in staat is deze software uit te voeren. Alle code is geschreven door de gehele project team.

Hardware

Voor de hardware wordt er gebruik gemaakt van twee drones van de merk CrazyFlie 2.0. De architectuur van deze drones is te vinden in de documentatie van CrazyFlie. De gehele simulatie zal deze drones aan moeten staan, aangezien dit de visuele indicatie is van de gedrag van bijen. De centrale server is een python script die op een Windows pc draait. Deze laptop hoeft niet krachtig te zijn, gezien zijn voornaamste taken locaties ontvangen, omrekenen en commandos versturen naar de drones zijn. Ook deze heeft geen down-time tijdens de simulatie, want anders weet een bij/drone niet wat het moet doen. Ten slotte is er een apparaat met een camera nodig. Er zou gekozen kunnen worden voor een raspberry pi met een PiCam v2, maar het werkt net zo goed met een laptop met Windows waar een goedkope webcam aan is gesloten. Daarom is er uiteindelijk gekozen voor een laptop met een webcam. Deze webcam kan fotos en films in full HD maken. Beide de laptop en de webcam zijn constant aan, gezien de drones ten alle tijden zijn locatie moet weten.

5.2 Test identification

De testen zullen betrekkeing hebben tot de subsystemen van het project. Om het volledigen systeem te testen, zullen de subsytemen op hun individuele onderdelen getest worden. Onder deze onderdelen valt het volgende, in/uitgaande communicatie en gedrag dat het subsysteem uit gebaseerd op deze informatie. De in en uitgaande communicatie zal plaatsvinden in een test die losstaat van de subsystemen, omdat de toepassing hiervan hetzelfde is van de systemen.

5.3 Planned tests

In deze paragraaf worden de testen neergezet die bij de test resultaten behandeld zullen worden. Deze testen zijn niet directe testen maar zullen gegroepeerde systeemfuncties testen.

5.3.1 MQTT Communicatie

A	Test objective	MQTT comuunicatie
B	Test Level	System wide
C	Test type	API testing
D	Requirements	Camera MQTT communicatie, Swarm MQTT communicatie, Simulatie MQTT communicatie
E	Invoer	Message: drone_id, state, position
F	Verwachte uitkomst	positie per drone_id

5.3.2 Camera Object tracking

A	Test objective	Plaatsbepaling van object
B	Test Level	Camera Subsystem
C	Test type	Unit test
D	Requirements	Camera plaatsbepaling van objecten, omvormen van pixel naar grid locatie
E	Invoer	Camera gemonteerd aan plavond, drone geplaatst in beeld van camera
F	Verwachte uitkomst	Border om locatie drone, grid locatie drone

5.3.3 Drone foodsearch

A	Test objective	Drone vliegt naar de locatie van eten
B	Test Level	subsystem
C	Test type	Unit tests
D	Requirements	Path generation, Path navagation,
E	Invoer	Een drone URI voedsel locatie
F	Verwachte uitkomst	De aangewezen drone verplaatst zich naar de voedselbron en komt dan terug naar de korf om te dansen

5.3.4 Drone foodget

A	Test objective	Drone vliegt continue tussen eten en korf
B	Test Level	subsystem
C	Test type	Unit tests
D	Requirements	Path generation, Path navagation, collision detection
E	Invoer	Een drone URI
F	Verwachte uitkomst	De aangewezen drone verplaatst tussen de voedselbron en de korf, zonder dat deze tegen andere drones botst.

5.3.5 Crazyflie weergeven in simulatie

A	Test objective	Het weergeven van de locaties van hardware drones in de simulatie
B	Test Level	subsystem
C	Test type	UI testing
D	Requirements	Hardware weergeven in simulatie
E	Invoer	MQTT message met drone_id, postitie
F	Verwachte uitkomst	De positie van de crazyflie drone word weergegeven in de simulatie

6 System Test Report

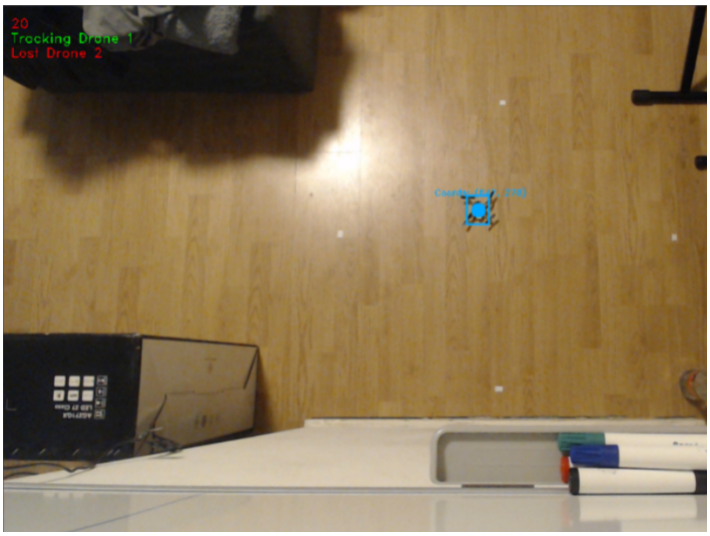
In dit hoofstuk worden de testresultaten behandeld, de testplannen hiervoor staan in het vorige hoofdstuk genaamd System test plan.

6.1 MQTT communicatie

Voor het testen van de communicatie MQTT, zijn aparte test files geschreven. De messages worden gevormd in de ene test file genaamd test_publish.py. Deze kunnen vervolgens succesvol worden uitgelezen in test_recieve.py.

6.2 Camera object tracking

De camera kan in een live beeld de drone onderscheiden ende locatie van deze drone aangeven. Vervolgens worden deze pixel coordinaten omgevormd naar grid coordinaten die doorgestuurd worden. In de afbeelding is te zien hoe de drone wordt onderscheiden van zijn omgeving en coordinaten aan hem worden toegepast.



6.3 Drone Foodsearch

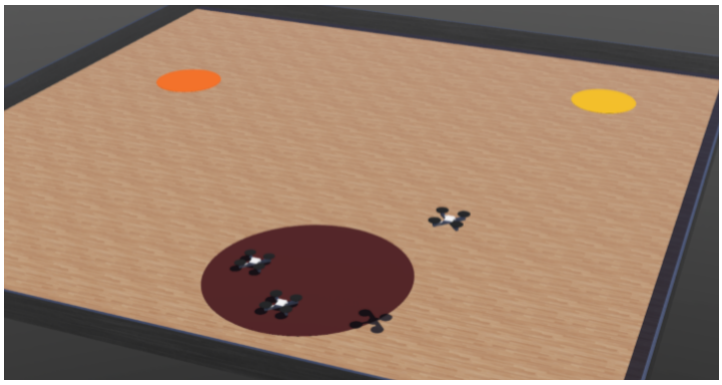
De crazyflie heeft soms problemen met opstijgen door een lage batterij, maar zolang de batterij minstens 25 procent batterij heeft is er niks aan de hand en vliegt de drone, via kleine, snelle stappen, naar de voedselbron volgens de route die pathfinding geplanned heeft. Zodra de drone geland is op de voedselbron stijgt hij weer op om terug te keren naar de korf, als geariveerd begint hij met dansen door 360 graden links en dan 360 graden rechts te draaien. Na het dansen land hij in de korf, mogelijk is hij een beetje verplaatst door het dansen, maar niet zoveel dat hij buiten de korf zou komen. Deze test is ook te zien in filmpje dat hoort bij de opleverset.

6.4 Drone Foodget

Het gedrag van de drone is voor het eerste vlieggedeelte hetzelfde als bij de Foodsearch. Echter zodra hij bij de korf arriveert land hij en zolang de voedselbron nog niet leeg is stijgt hij weer op om het te herhalen. Dit gedeelte van de test werkt, maar de collision avoidance werkt op het moment van het inleveren van dit rapport helaas nog niet, dit zal in de toekomst nog gedaan worden.

6.5 Crazyflie weergeven in simulatie

De informatie voor de locatie van de Crazyflie wordt doorgegeven door middel van MQTT messages. Met de binnengekomen berichten wordt het verschil in thuis en verplaasting uitgerekend zodat de beweging in de simulatie vloeiend is. Deze test is te zien in de bijgeleverde video in de opleverset.



7 Risico Analyse

Nr.	Risico's	Kans	Gevolg	Maatregelen	Kans	Gevolg
1	Niet genoeg budget voor het realiseren van een Swarm.	8	middel	Alternatieve oplossing vinden voor het realiseren van een swarm	1	middel
2	Breken van propellers tijdens vliegen van drone.	5	middel	Nauwkeurig te werk gaan en de drone gecontroleerd besturen	3	middel
3	Webots omgeving is niet compatabel met programmeer taal Python.	7	hoog	Uitzoeken of Webots mogelijk te gebruiken is met Python.	5	middel
4	Te weinig kennis/vaardigheden voor het ontwikkelen van kleur detectie algorithme.	7	middel	Uitzoeken of het mogelijk is om gebruik te maken van kant en klare libraries.	1	laag
5	Niet op elkaar afgestemde losse gebouwde componenten.	9	hoog	Vooraf afspreken hoe het gehele systeem architectuur uit komt te zien.	4	middel
6	Besturing van de drone is niet genoeg om het volgens grid coordinaten systeem te laten vliegen.	6	middel	Uitzoeken of crazyflies iets hebben waarbij positie bepaling of stabilizatie mee gedaan wordt.	2	middel
7	Niet bruikbare MQTT communicatie tussen simulatie en de server.	6	laag	Uitproberen welke tijdsintervallen het beste prestatie leveren.	1	laag
8	Inadequate swarm gedrag realiseren.	5	hoog	De tijd nemen om swarm gedrag uit te denken. De scenario die de swarm uit moet voeren in schema zetten.	1	middel
9	Te lange wachttijd bij levering van hardware componenten.	8	hoog	Langs andere groepen gaan om te kijken of bij hun wat te lenen valt.	4	middel
10	Niet op tijd inleveren van opleverset	5	hoog	Ruim op tijd een dag afspreken om er voor te zorgen dat alle documentatie voor de swarm project af is.	4	hoog

Tabel 1: Risico analyse voor project AutomatedBeezzzz voor Tinlab Automated Systems

8 Conclusie

Referenties

[1] Reza Akbari, Alireza Mohammadi, and Koorush Ziarati. A novel bee swarm optimization algorithm for numerical function optimization. *Communications in Nonlinear Science and Numerical Simulation*, 15(10):3142–3155, 2010.