Warehouse Drone Collision Avoidance Midterm Report

Tristan van Vegchel
Seacon Logistics

Venlo, October 21, 2019

Information Page Internship / Graduation Report

Fontys University of Applied Sciences
School of Technology and Logistics
Post Office Box 141, 5900 AC Venlo, Netherlands

Type of report:	Graduation report
Student name:	Tristan van Vegchel
Student number:	2762870
Study:	Software Engineering
Period:	September 2019 – February 2020
Company name:	Seacon Logistics
Address:	Celciusweg 60
Postal code + City:	5928 PR Venlo
Country:	The Netherlands
Telephone:	+31 (0)77 - 327 55 55
Company supervisor:	Kai-Arne Reiter
Supervising Lecturer:	Sander Bruinsma
External commissioner:	T.B.D.
Company Confidential:	*No*
Number of words:	3349
Coding:	*(see document "Coding of the Research". Logistic students only)*

STATEMENT OF AUTHENTICITY

Issued by the FHTenL Examination Board, September 2017

I, the undersigned, hereby certify that I have compiled and written this document and the underlying work / pieces of work without assistance from anyone except the specifically assigned academic supervisor. This work is solely my own, and I am solely responsible for the content, organization, and making of this document and the underlying work / pieces of work.

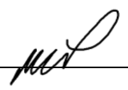
I hereby acknowledge that I have read the instructions for preparation and submission of documents / pieces of work provided by my course / my academic institution, and I understand that this document and the underlying pieces of work will not be accepted for evaluation or for the award of academic credits if it is determined that they have not been prepared in compliance with those instructions and this statement of authenticity.

I further certify that I did not commit plagiarism, did neither take over nor paraphrase (digital or printed, translated or original) material (e.g. ideas, data, pieces of text, figures, diagrams, tables, recordings, videos, code, ...) produced by others without correct and complete citation and correct and complete reference of the source(s). I understand that this document and the underlying work / pieces of work will not be accepted for evaluation or for the award of academic credits if it is determined that they embody plagiarism.

Name (in capital letters): TRISTAN VAN VEGCHEL

Student number: 2762870

Place / Date: Venlo / 21-10-2019

Signature: 

Contents

1 Introduction **1**
- 1.1 Background Information . . . 1
- 1.2 Context . . . 1
- 1.3 Reporting Structure . . . 2

2 Drone Simulator **3**
- 2.1 Preliminary Research . . . 3
- 2.2 Current State . . . 4
- 2.3 Future Plans . . . 6

3 Collision Avoidance **7**
- 3.1 Comparison of Approaches . . . 7
- 3.2 Current Status . . . 9
- 3.3 Future Plans . . . 9

4 General Adjustments **10**
- 4.1 Git Pages . . . 10
- 4.2 Pivotal Tracker Updates . . . 10
- 4.3 Risk Management . . . 10

5 Self Reflection **11**
- 5.1 Domain Specific Competences . . . 11
- 5.2 Research Skills . . . 12
- 5.3 Professional Behavior . . . 12

Appendix A Drone States **14**

Appendix B Simulation Class Diagram **15**

Appendix C Epics & Stories **16**

Appendix D Risk Management **17**

List of Figures

1.1 Roadmap of the cycle count drone project. 2

2.1 The PyQuadSim project running. (3) 3
2.2 Screenshot of the Unity simulator gameplay. 4
2.3 The Unity simulator in training mode. 5

A.1 State model of a drone performing inventory control 14

B.1 Partial class diagram of the code behind the simulation. 15

D.1 Risk matrix containing the risks with their mitigation plans. 17

List of Tables

C.1 Table of all epics with their major tasks. 16

List of Abbreviations

A.I. Artificial Intelligence.

Fontys Fontys University of Applied Sciences.

SSD Single-Shot Detector.

STG2 Graduation Project.

Definitions

Artificial Intelligence The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

Reinforcement Learning A set of genetic algorithms that tries to find the optimal state-action pairs in order to maximize the outcome of a predefined reward formula.

Chapter 1

Introduction

This introductory chapter provides the reader with background and contextual information regarding this project.

1.1 Background Information

This report is a mandatory deliverable for the Graduation Project (STG2) module, with the deadline being on the 20th of October 2019. It marks the end of the first half of the semester, and exists to document the progress made thus far and the relevant decision-making that came with it. This document is also an indication of what is yet to come, with respect to the current state of the project.

1.2 Context

The warehouse drone collision avoidance project is a graduation project assigned by Seacon Logistics. Its main purpose is to analyze, design, and implement a solution that is capable of providing drones with a means to avoid collision with other entities in warehouses. However, since this project is the first component of a larger project, providing advice on how to approach the future components based on this project is also a relevant part.

Seacon is looking to develop their own semi-autonomous solution for inventory and control management in their warehouses. According to the plan, this would be done by semi-autonomously flying drones that, among other things, check damage, count the contents of pallets, and scan barcodes. However, before all those tasks can be executed the drone is required to fly to a desired location without failure, which the collision avoidance solution will be used for.

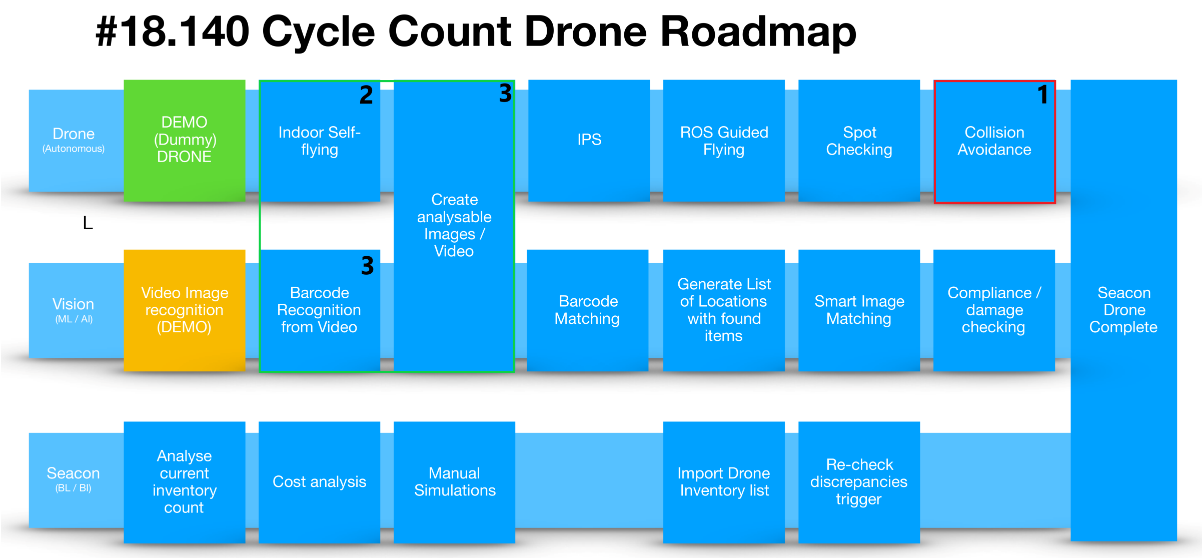


Figure 1.1: Roadmap of the cycle count drone project.

Due to the image size, the states the drone will traverse through can be found in appendix A. To display how tasks will eventually integrate, the scanning of barcodes has also been added. At a later point in time, however, this could be replaced with a task state containing all the specific tasks as substates.

1.3 Reporting Structure

The following subjects will come to pass: Firstly, the drone simulation is discussed. Secondly, the research done around the collision avoidance solution up until now will be described. Thirdly, the general adjustments made will be mentioned. Fourthly and finally, a self-reflection is given. All parts (aside from the general adjustments) will also include their respective future plans.

Chapter 2

Drone Simulator

This chapter describes the current state of the simulation, the preliminary research done, and the planning regarding further improvements.

2.1 Preliminary Research

As testing on live drones is generally not desired due to potential damaging costs and limited battery life, there was opted for a simulation.

When it comes to simulations, drones are commonly-used objects. With this in mind, the initial plan was to reuse an existing project. Since the collision avoidance algorithm will most likely to be developed using Python, it would be ideal to make use of a Python-native simulation. Ultimately, this led to project PyQuadSim by Github user "simondlevy". (4)

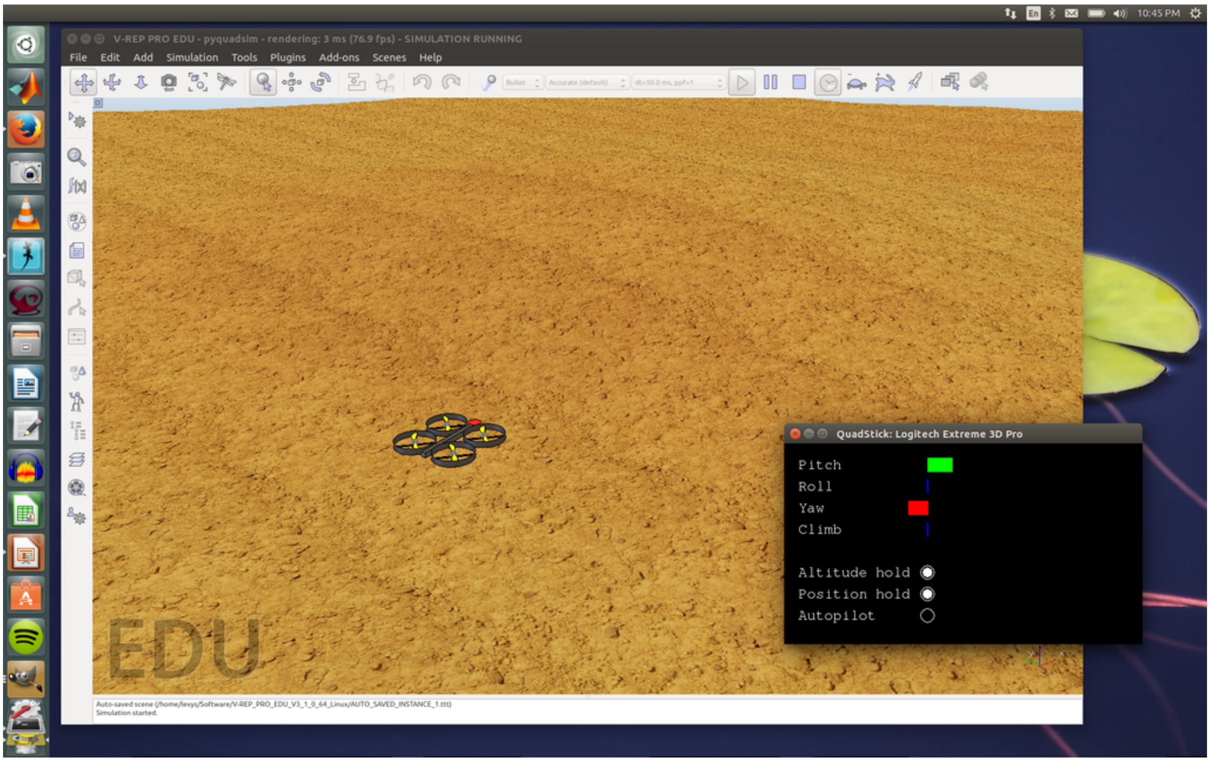


Figure 2.1: The PyQuadSim project running. (3)

One of the biggest problems of using most Artificial Intelligence (A.I.) algorithms, is transferring its knowledge to the real application. In the case of for example reinforcement learning, there are generally 2 approaches possible when it comes to real life applications: Train it in the real life environment, or use a simulation with transfer learning. Same as with this project, the former is often too expensive and come with too many risks. However, while simulations are the safer option, they come with the risk of not providing an identical environment. As it is difficult to determine or guide the algorithm on what states which actions should be taken, chances are that the algorithm will react improperly in the real environment if the simulation was not realistic enough. With this in mind, it was decided to look elsewhere for a solution that could provide a simulation more tailored to this case.

Based on the arguments above and on the advice from the company supervisor it was decided to develop a new simulation. The Unity game engine was chosen since it is easy to pick up, and 3D models for both the drone and the warehouse were available.

2.2 Current State

As of writing this document, a simulation has been made using Unity. It is capable of generating a variably-sized warehouse with interior and targets. It is designed to be controlled from the first person perspective of the drone, with as goal to collect the targets in the warehouse before the timer runs out. When training an algorithm the simulation should run at an increased speed.

Figure 2.2: Screenshot of the Unity simulator gameplay.

To facilitate this, a training mode has also been added to the simulation. This mode cuts out most of the aesthetics in order to decrease processing overhead. Using this in combination with running the game in a small resolution window should allow for faster game speeds.

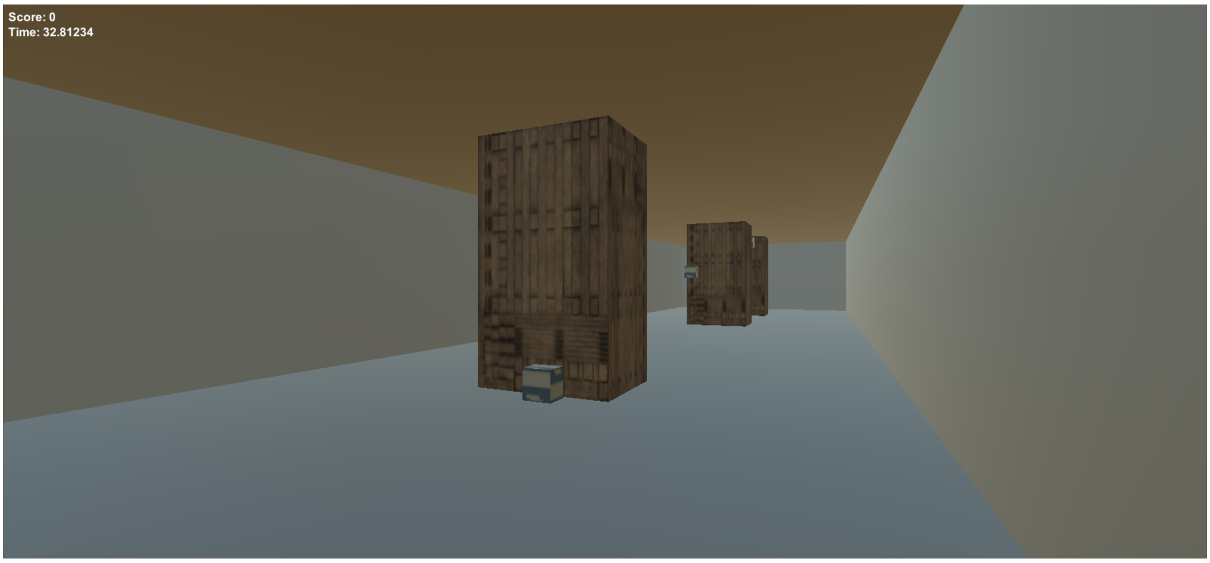


Figure 2.3: The Unity simulator in training mode.

The implementation of the simulation contains 4 major parts: generator classes, the Unity package, the ML-Agents package and its inheriting classes, and the class for reading a configuration file. Since the Unity package is the default package that all c# Unity projects use, it will not be discussed here. A class diagram can be found in appendix B;

Generator Generator is an abstract class that has concrete implementations for spawning the drone, warehouse, and the targets. Using the strategy design pattern, all the generators are registered and invoked by the DroneAcademy class.

DroneAcademy DroneAcademy inherits from the Academy class within the ML-Agents package. It is responsible for all operations around the environment during training, for example destroying or adding new objects to the scene. (2)

DroneAgent DroneAgent inherits from the Agent class within the ML-Agents package. The object containing an implementation of the Agent class is meant to be controlled by a neural network (or "brain" in the context of Unity) during training. (2)

ConfigHandler ConfigHandler reads an external configuration file that contains information about how to generate the scene. Examples include the size of the warehouse and whether or not to enable training mode (see figure 2.3).

2.3 Future Plans

To ensure that not too much time is spent on the simulator itself, the current version (aside from minor tweaks) will be made use of to run the first set of trainings. Based on the results of these, the simulator might receive additional changes. Potential improvements currently include more realistic drone movement by shifting to a propelled force movement system, more extensive and variable interior layouts, a class separating the generation part from the Academy object, and varying building/rack heights.

Chapter 3

Collision Avoidance

This chapter concerns the research part of the project regarding collision avoidance. The main research question states: "What is the feasibility and, if applicable, performance of A.I. techniques?" As the main priority of the entire drone project is to reduce costs, it is preferred to make use of an A.I. technique that could reduce or even remove the need for adding sensors to the drone.

3.1 Comparison of Approaches

By doing research into similar cases, a comparison of approaches has been formed. Each approach comes with a description, its advantages, and disadvantages.

Single-Shot Detection Single-Shot Detector (SSD) is an algorithm that only requires one picture to detect multiple objects. It is often trained on large datasets containing the majority of everyday objects. However, it can also be trained on custom datasets, or, alternatively, only a subset of the labeled objects from pretrained networks can be used. An approach for this project could be to use the pretrained SSD model on the Tensorflow Model Zoo Github and only detect the necessary labels such as vehicles/forklifts, humans, drones etc. This could then be combined with a distance sensor to predict whether a collision will happen.

The biggest advantage of SSDs is them being very lightweight in terms of processing power. Drones usually don't have high processing power on-board, and due to the sheer amount of constant data transfer a connection-based solution is unfavorable. Performance speed is essential in the context of real-time processing.

The only part done by the deep learning model is the object detection. This means that parts like distance calculation and recalculating the route in order to actually avoid collision have to be implemented manually. A standalone SSD requires a reference point with a known size to determine the distance from just visual footage. This would mean that every object should a reference point that is visible from every angle. This is very impractical and thus extra sensors are required.

Subsumption Architecture Originally described by Rodney Brooks, it could be considered the opposition of A.I.. This is due to the fact that subsumption architecture makes use of sensory input to layer competences, instead of being guided by mental/behavioral-based algorithms like A.I. algorithms typically are.(1)

Scalability is one of the major strong points. Since subsumption architecture makes use of a bottom-up approach, theoretically it should be possible to create a device that only needs additions when extending the solution.

Subsumption architecture is based on sensory input, which requires the drone to be equipped with necessary sensors. This is generally not desired as this is both not a cost-efficient solution, and will affect the battery life/performance of the drone. Another major disadvantage is the complexity when it comes to designing a scalable system.

Curriculum Learning Curriculum learning is subset of reinforcement learning. As the name implies, it is based on the principle of following a course or curriculum where the complexity of the assignment is gradually increased.

By layering the training into multiple difficulties, the risk of getting stuck in a local optimum or getting misbehavior will be smaller. By dividing it, it will be easier to determine which parts the neural network understands well and which it does not understand at all.

Gradually increasing the training means that extra work needs to be done in order to ensure that each part is understood and carried over properly before it moves on next difficulty. This will most likely require extra time necessary for training, as well as for writing the implementation. Moreover, since curriculum learning is a subset of reinforcement learning, all problems are inherited too to a certain degree. Notable examples include not defining the rewards and punishment criteria properly, or the difficulty with understanding and making decisions for longterm results.

Imitation Learning Imitation learning is another form of reinforcement learning. This time, however, the definition of a reward/punishment formula will not be necessary. Imitation learning is a form of supervised-learning that generates its own intrinsic reward formula based on imitating human behavior. In the context of the drone simulator (or any other game), this would require an expert to play and record the simulation multiple times, and in turn feed that gameplay to the algorithm. Based on the input, the algorithm would then determine what the human expert thinks is important in order to maximize the performance, and apply that to its own gameplay.

Imitation learning thrives in contexts where human-like behavior (and thus optimal performance is not necessarily important) is desired. Also, whenever it is not certain what actions exactly the network should be rewarded or punished for, imitation learning is likely to outperform its competitor algorithms. Finally, as seen in the video above, with certain tasks the algorithm learns to copy the human at a very fast rate. There are 2 major disadvantages when it comes to imitation learning. Firstly, it is supervised. This means that clean and proper data is required for an algorithm to be trained properly. Clean data should not only consist out of a good performance, but also scenarios where the agent could learn how to avoid mistakes should be included. Since the algorithm is trying to copy the expert's (human) behavior, it is bound to copy (minor) mistakes as well.

3.2 Current Status

As of writing this report, one approach has partially been prototyped: curriculum learning. This was chosen first because it does not need any extra sensors, and, while the set up is similar to that of imitation learning, that of curriculum learning takes less time and has more documentation available, which is ideal for creating an initial prototype. However, even before having implemented the entire curriculum a problem already occurred. Namely, the problem with making decisions based on longterm results. Concretely, while the current software does allow an algorithm to train itself on it, it will not ever move. This is due to the fact that no reward is given for just moving around. The danger with giving a reward to something that is not a goal, is that the algorithm might misinterpret it as its main task. Even worse, the algorithm might find a way to get a higher score doing trivial tasks than actually performing as it was supposed to.

3.3 Future Plans

As the main focus of this project is to satisfy the graduation internship requirements, it is important to define a point of no return. This is a point where the final decision on what approach to use during this time scope has to be made. Especially in the context of A.I. development, time is often a big risk.

Curriculum learning will most likely not produce any significant results within this time span, and will thus be dropped. As imitation learning does not suffer from the same reward function problems, it will be the next approach to take. However, since A.I. approaches generally take a lot of time, it will only be looked at very shortly. If it does not show enough potential for the remainder of this project by the midterm presentation, it will be swapped for an approach using extra sensors (e.g.: SSD).

Chapter 4

General Adjustments

Here any major changes that are not necessarily directly relevant to one of the (end) products are mentioned here.

4.1 Git Pages

In order to facilitate agile development, a website has been developed using Jekyll and Git Pages. Through this site new builds of the simulation are published with patch notes, as well as other relevant documentation such as a comparison of approaches for collision avoidance.

4.2 Pivotal Tracker Updates

Small adjustments were made to the pivotal tracker. These changes mainly occur due to some tasks being obsolete or replaced by other tasks. As the project progresses requirements and feasibility change from their original forms, hence why the Pivotal Tracker should be adjusted accordingly. The new list can be found in appendix C.

4.3 Risk Management

In order to facilitate risk management, a risk matrix has been made. Each of the risks contains a score, a mitigation action/plan, and a updated score after the mitigation plan has been invoked. The matrix can be found in appendix D.

Chapter 5

Self Reflection

This self reflection is divided into 3 parts: the specific activities/competences according to the domain description for IT bachelors, research skills, and professional behavior. Each part will include a short definition, a reflection on how this project is progressing with respect to that part, and a plan to improve on the current status.

5.1 Domain Specific Competences

According to the official regulations for graduation internships, it is required for an intern to display an equivalent to the N3-level amount of skill within the internship period. This roughly means that the selected activities should have a generally complex and unstructured nature, which should be fulfilled by the intern independently. The selected activities have to be 3 out of the following: manage, analyze, advise, design, implement. Furthermore, these activities have to be within at least 1 architectural layer (user interaction, business processes, infrastructure, software, hardware interfacing). The activities for this project are within the software layer, and are as follows:

- **Advise:** Provide research-based advice on the most suitable approach for collision avoidance when keeping the requirements and wishes in consideration.
- **Design:** Design an adjustable/scalable simulation environment for testing and/or training the solution.
- **Implement:** Implement the simulation and train (if applicable), optimize, and tailor the algorithm for collision avoidance. The algorithm should also be connected to the simulation.

Starting off, the advice part is making good progress. Evidence is gathered based on comparisons, already existing papers/research, and own research/prototypes. Unfortunately, the design and implementation are not up to standard yet. As explained in section 3.2, the current approach is unlikely to be completed within the remaining time scope, and while the simulation software is functional, does not fulfill the requirements for being a N3-level design activity. Having this happen was a known risk, yet it was still chosen based on the company's wishes and requirements and in order to satisfy the research part. Looking back the most time-costly mistake was to not make a prototype of the simulation to train on first. This was considered before starting, but based on research done the fact that the training environment needs to be very similar to the real one came out, which in combinations with the wishes of the company led to the current decision.

As ultimately the most important task is to pass, the assignment will get some minor adjustments so that all competences can be adequately fulfilled. This will be done during the weeks of the midterm reports and presentations, so that both the feedback of the presentation as well as the wishes of the company can be incorporated into the new plan.

5.2 Research Skills

When assessing one's research skills, thing such as ability to use state-of-the-art techniques and technologies, providing sound research with proper research questions and unbiased conclusions, criteria-based decision making, and making use of relevant sources are considered.

The research question is: "What is the feasibility of using reinforcement learning-based approaches for collision avoidance as opposed to computer vision and sensor-based approaches?" While the current implementation will not be enough to satisfy the requirements in the previous section, it does provide a lot of research products. It makes use of state-of-the-art techniques (curriculum/imitation learning) in a very relevant context (collision avoidance and autonomous drone control). Currently, the research will be continued by doing research into the feasibility of imitation learning. Based on this in combination with other relevant sources a conclusion will be made. This will then be followed by documenting the whole research, with properly formulating the research and sub questions, and describing the whole research process. The main improvement point is defining the research sub questions earlier. Due to the uncertain nature of the approach, this was postponed until the project took more shape. However, looking back making a preliminary list of sub questions would have been better.

5.3 Professional Behavior

Professional behavior aims at skills such as problem orientation, creativity, project management, productivity, self-reflection, and communicating within a professional environment.

In order to maximize productivity, small quality improvements were introduced. First of all, as my attention span drops relatively fast, I prefer switching between small tasks a lot than to work on one long task for a long time. Thus, the daily planning has been designed so that there are always 2 different kinds of tasks available to avoid exhaustion from 1 type of task and thus reduced quality. Usually this would be a combination of reading/writing documentation and programming/designing. Working with this kind of system often allowed for additional online research time. Secondly, I consider structure an important thing. Because of this, aside from the long-term pivotal tracker planning made, a daily backlog with open to-dos has been made. While the company did have weekly stand up meetings, these were not very beneficial as most colleagues are not involved with my project. After communicating this with the supervisor, a private weekly progress meeting has been set up to show deliverables and notify him of important updates, as well as to ask questions.

Improvement points here are finding a better balance between the graduation internship requirements and the company's wishes. An example of this being the scalability of the simulation. Also, I have difficulty determining what parts are up to the necessary standard when it comes to the activities, which sometimes leads to making things too complex. Starting simple and then building up should be the way to go.

References

[1] Brooks, R. (1985), 'A robust layered control system for a mobile robot'.
URL: *apps.dtic.mil/dtic/tr/fulltext/u2/a160833.pdf*

[2] Juliani, A. (2018), 'Unity: A general platform for intelligent agents'.
URL: *https://github.com/Unity-Technologies/ml-agents*

[3] Levy, S. (2014), 'Pyquadsim: A python quadrotor simulator'.
URL: *https://www.youtube.com/watch?v=oQejljPA6ys*

[4] Levy, S. D. (2019), 'Github simondlevy'.
URL: *https://github.com/simondlevy*

Appendix A

Drone States

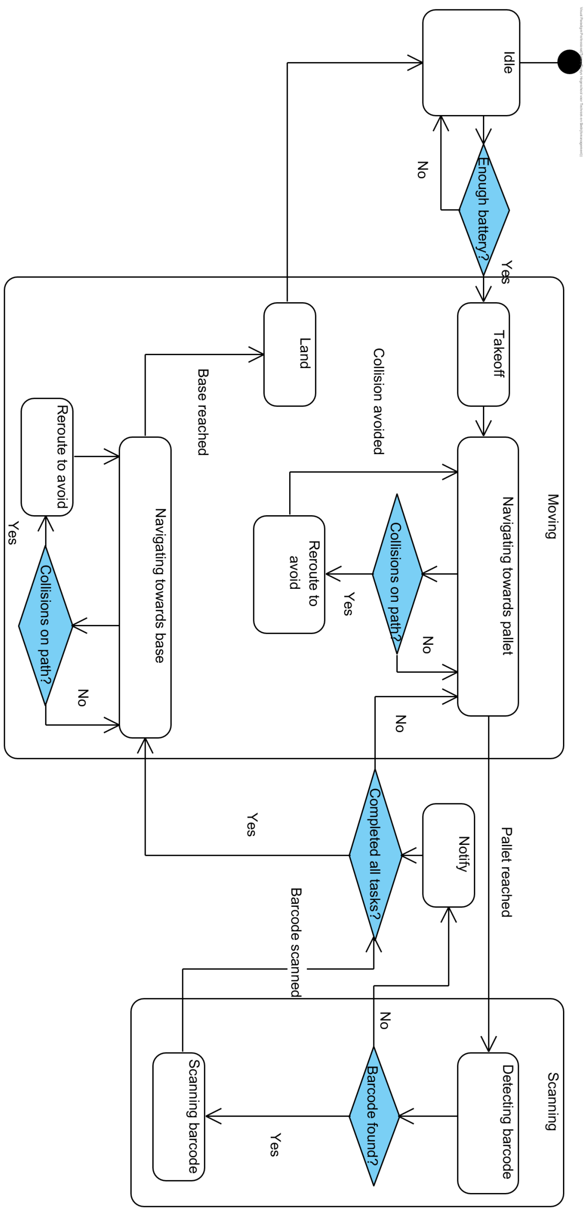


Figure A.1: State model of a drone performing inventory control

Appendix B

Simulation Class Diagram

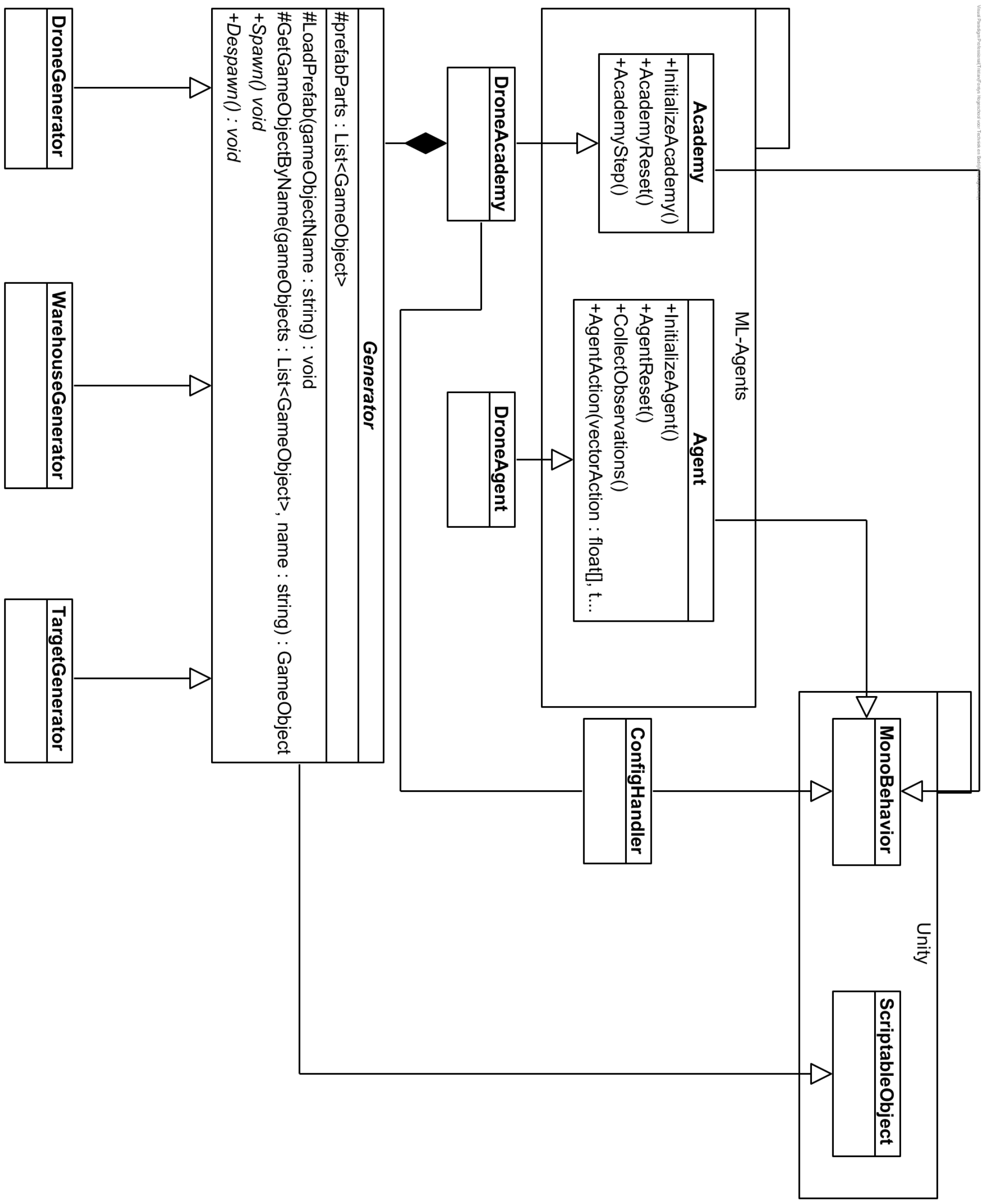


Figure B.1: Partial class diagram of the code behind the simulation.

Appendix C

Epics & Stories

Epic	Task	Level
School Tasks	Project plan	2
	Mid-term report	1
	Mid-term presentation	2
	Thesis report	3
	Presentation & defence	2
Showcasing and Agile	Select easily accessible showcasing environment	1
	Add simulation footage/build	2
	Add documentation	1
Drone Control	The drone should be able to obey scripted commands	1
	The script should be able to handle continuous input from ext sources	1
	The drone should be able to accept NN model based input	2
Demonstration and Testing	A fairly realistic warehouse environment should be made	1
	The rewards and punishment criteria should be defined	1
	Add targets that the drone needs to fly to	1
	Create a flyable drone	2
	3D model for forklifts should be made	1
	Add simulation footage/build	2
	3D model for random pallets	1
	3D model for human(oid)	1
Collision Avoidance	The rewards and punishment criteria should be defined	1
	The drone should use its camera to predict collisions	3
	The drone should maintain altitude	1
	The drone should avert collisions without diverting too far	3
Pathing	Add targets that the drone needs to fly to	1
	The drone should maintain altitude	1
	The drone should avert collisions without diverting too far	3

Table C.1: Table of all epics with their major tasks.

Appendix D

Risk Management

Cause	Consequence	Likelihood (1-5)	Impact (1-5)	Score (likelihood * impact)	Risk mitigation	New likelihood (1-5)	New impact (1-5)	New score (new likelihood * new impact)
Approach requires too much time	Incomplete products	4	4	16	Weekly progress meetings and defining (time)limits	2	4	8
Approach does not fit requirements	Unsatisfactory products	3	3	9	Weekly progress meetings	1	2	2
Miscommunication on requirements	Unsatisfactory products	1	4	4	Weekly progress meetings and documentation	1	1	1
Illness	Stagnation of project	2	2	4	Creating a planning with small buffer for risks and communicating with supervisors on how to continue	2	1	2
Project too big/complex	Subpar solution	3	3	9	Properly defining a scope with supervisor	2	2	4
Drone stops functioning properly	No proof of concept can be shown	2	2	4	Creating a simulation	1	1	1
Loss of files/data	Loss of progression	1	4	4	Making use of a repository	1	1	1
				50				19

Figure D.1: Risk matrix containing the risks with their mitigation plans.