# CU6051 Artificial Intelligence
# Assignment 2
# Group Coursework

**SEBASTIAN, 17024627**
**ABDIRIZAK, 11063786**

**Project Topic:** Sheep Herding

# Contents

**Contents** 2

**Design of AI Program** 3
- **Use Case Diagram** 3
- **High-level Use Case Descriptions** 3
- **Class Diagram** 4
- **State Transition Table** 5

**Research and Challenges of Topic** 5

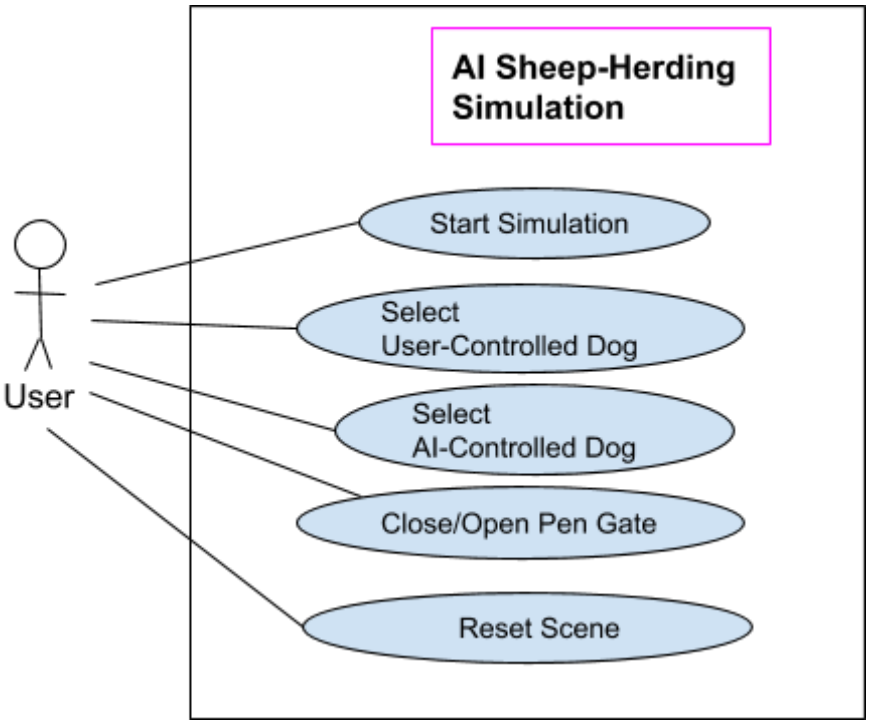
**Ideal Solution** 5
- **P.E.A.S of Ideal Solution: AI Robot Sheepdog** 6

**Prototype of Ideal Solution** 7
- **P.E.A.S of Prototype: AI Virtual Sheepdog** 7

**Bibliography** 8

# Design of AI Program

## Use Case Diagram




## High-level Use Case Descriptions

**Use Case: Start Simulation**
Actors: User
Description: The user starts the simulation of the sheep-herding by loading the unity project and then pressing the play button.

**Use Case: Select User-Controlled Dog**
Actors: User
Description: The user chooses the dog to be user-controlled by pressing the corresponding button on the screen.

**Use Case: Select AI-Controlled Dog**
Actors: User
Description: The user chooses the dog to be AI-controlled by pressing the corresponding button on the screen.

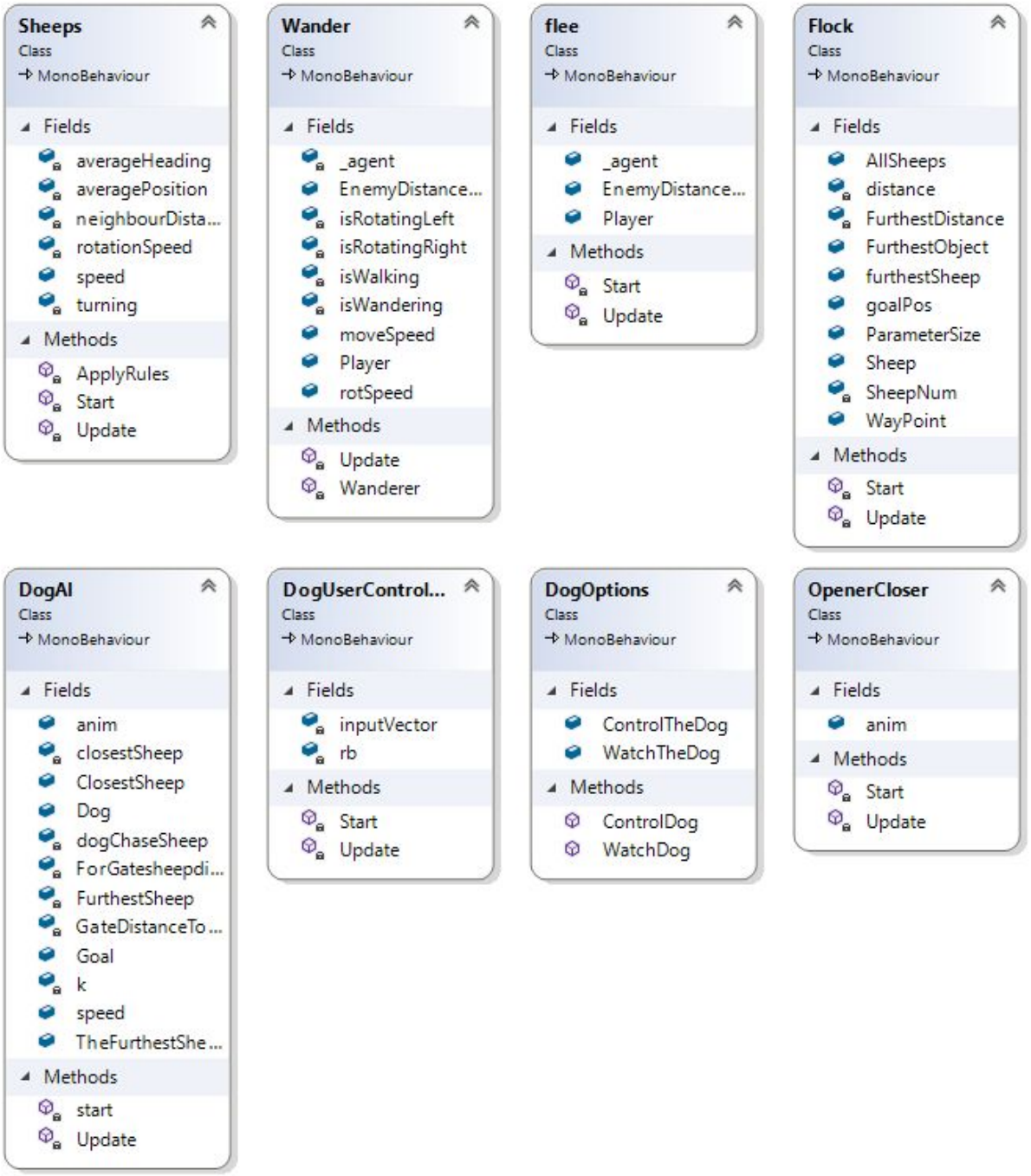
**Use Case: Close/Open Pen Gate**
Actors: User
Description: The user opens or closes the pen gate (depending on gate's current state) by clicking the "E" key on the keyboard.

**Use Case: Reset Scene**

Actors: User

Description: The user resets the scene of the sheep-herding by clicking the “R” key on the keyboard.

## Class Diagram



### State Transition Table

There are two AIs in the project, the dog and the sheep. The dog only really has 1 state when its in AI mode, which is to herd the sheep towards the pen. The dog can be user controlled too. The sheep, however, will have different states:

| SHEEP | | |
|---|---|---|
| **Current State** | **Condition** | **State Transition** |
| Wander | If dog is too close | Flee & Flock |
| Flee & Flock | If dog is far enough | Wander |

## Research and Challenges of Topic

As Unity was completely new to us, we had to do a lot of research and constantly try to look things up to find out how to do certain things or solve our issues. Using C# with unity introduces a lot of new functions and syntax that we had no experience with. One thing we wanted to know was how exactly we could make the dog position itself so that the sheep would run away from the dog, but at the same time be running towards the pen. The initial solution we came up with was to subtract the coordinates of the pen away from the coordinates of the sheep to get the coordinates or vector that the dog should be on to make the sheep run away to the pen. This solution was partly correct, as it positioned the dog on the correct vector, however the sheep only runs if the dog is within a certain range, but the dog was too far from the sheep with just that one calculation. We considered a function Vector3.Normalize, which would shrink the vector, then we could multiply that shrunk vector/coordinate to whatever we wanted so that we could choose exactly how close the dog would position itself to the sheep. We ended up using angles combined with vector3 to determine where the dog approaches from.

## Ideal Solution

We decided to base our project around sheep herding. When thinking about where an AI could be used in a sheep herding scenario, we thought of two things: the sheep and the sheepdog. In a real-life scenario, there would not be much use for an AI sheep except for possibly training a real dog for herding, so we decided the AI that would be most useful is an AI dog.

Our ideal solution would be an AI robot sheepdog that would herd real sheep.

A robot dog would have a lot of differences compared to a real dog. The robot dog is a mechanical object, while a real dog is a conscious animal. The robot dog will only follow the instructions it has been given in its code, so will be very efficient at herding sheep, while a real dog would take quite a bit of training to learn how to herd sheep, and even then it could possibly get distracted in lots of ways. Also once the program has been completed, it can be reused to reproduce a robot dog very fast, while a new dog would need to be trained from scratch. The maintenance of the two types of dogs is also different. The robot may only require charging with electricity to function, and it can be kept anywhere without needing to

pay attention to it. A real dog needs to eat and drink to live, and as a conscious animal it has feelings that attention should be paid to. It would also need exercise through activities such as walks. Something that a real dog has that a robot dog would not is the ability to be a companion to a person. As a person and a real dog both have a conscience, they can build a relationship with each other. This would not really be possible with a robot dog.

The ideal solution should be a mechanical sheepdog capable of moving quicker than the sheep so that it can easily get them to their destination instead of chasing them forever. The mechanical sheepdog should be stable so it does not collapse on terrain, although the field would likely be flat and obstacle-free.
The sheep should be scared of the robot sheep and want to run away, so a way of doing this, apart from the visual design, would be to make it sound a loud noise that will make the sheep run from it.
It should have a way to know where all the sheep are. A motion sensor would be able to detect them, however it could possibly detect something that isn't a sheep, or the sheep may get too far from it. The best way would be a GPS tracker. All sheep could be fitted with a tracking chip that allows the robot sheepdog to know where all sheep are at all times. With the coordinates of the pen the sheep should be directed to, the robot sheepdog could find out if certain sheep are straying further away from the pen, or which sheep is the furthest away. It will also know when the sheep are in the radius of the pen. The robot sheepdog will use the gps information to position itself where the sheep will run away from it and towards the pen, as well as focusing on putting the furthest straying sheep back on course.
There should be a way to manage the robot sheepdog, like turning it off or making it follow the user. This could be done from a mobile device with an app. The robot sheepdog could be made to track and follow the mobile device to lead it home. The volume of robot sheepdog's noises should be able to be manually triggered or its volume altered too, incase it is required due to some sheep not being scared enough or external factors like weather. The user could also keep track of battery levels.

## P.E.A.S of Ideal Solution: AI Robot Sheepdog

**Performance Measure:**
Whether the dog and the sheep can make it to the destination or not
The speed the dog puts the flock away
The amount of sheep that get lost

**Environment:** Farm/Field

**Actuators:** Mechanical limbs, coded instructions

**Sensors:** Wireless/bluetooth receiver, GPS reader

# Prototype of Ideal Solution

While our ideal solution's general concept isn't something incredibly complex, actually developing our ideal solution would require extreme amounts of time and resources/investment, so the exact ideal solution is currently infeasible. Our AI project is a small demo version or prototype of our ideal solution. It is a virtual AI dog in a virtual environment. As it is a virtual project, a virtual field (containing a plane and pen) and virtual sheep needed to be created, so an AI had to be developed for the sheep too. There was a lot more to the sheep than initially thought. Wandering was programmed to have the sheep walk randomly, to try to imitate real sheep. Fleeing was programmed so that the sheep run away from the dog when it comes within a certain distance of the sheep, and when they are far enough away from the dog, they will resume wandering. The ideal solution would use sound to scare the sheep, however in the demo just the distance between the dog and the sheep will be used to determine if the sheep is scared and will flee. Flocking was also programmed so that the sheep attempt to stick together, also trying to imitate real sheep. Programming the virtual sheepdog's functions has some identical logic to if it were the robot sheepdog being programmed, however it could get extremely complex trying to program around mechanical limbs. For the demo, the dog will just be translated from point to point. Initially, a user-controlled dog was made that allowed movement of the dog with the arrow keys. This way mainly used to test the behaviour of the sheep. After the sheep AI was working as intended, the sheepdog AI was developed. Our virtual sheepdog will have access to the positions of all existing sheep. It will use these positions to calculate distances between the sheep and the pen. The dog will prioritise the sheep furthest away from the pen, and will calculate using vectors the position it should approach from which will make the sheep run towards the pen.

When all the sheep have successfully been herded into the pen, the pen's gate will automatically close. A feature was also programmed to allow the user to manually open and close the gate from the click of a keyboard key, used to trap the sheep. This would represent a real-life shepherd closing the gate.

## P.E.A.S of Prototype: AI Virtual Sheepdog

**Performance Measure:**

Whether the dog and the sheep can make it to the destination or not
The speed the dog puts the flock away
The amount of sheep that get lost

**Environment:** Virtual Field

**Actuators:** Modelled Limbs(unity) with C# backend functions

**Sensors:** The reading of variables by the program

# Bibliography


- Answers.unity.com. (2014). *Find angle between two gameobjects? - Unity Answers*. [online] Available at: https://answers.unity.com/questions/678369/find-angle-between-two-gameobjects.html [Accessed 7 Jan. 2020].

- YouTube. (2016). *Flocking Fish in Unity 5: Creating Schooling behaviour with simple AI.*. [online] Available at: https://www.youtube.com/watch?v=eMpI1eCsIyM&t=1320s [Accessed 2 Jan. 2020].

- Answers.unity.com. (2011). *How do I convert angle to vector3? - Unity Answers*. [online] Available at: https://answers.unity.com/questions/54495/how-do-i-convert-angle-to-vector3.html [Accessed 8 Jan. 2020].

- YouTube. (2018). *Unity Navmesh AI Tutorial : Flee from Enemy*. [online] Available at: https://www.youtube.com/watch?v=Zjlg9F3FRJs [Accessed 29 Dec. 2020].

- Technologies, U. (n.d.). *Unity - Scripting API: Animator.SetTrigger*. [online] Docs.unity3d.com. Available at: https://docs.unity3d.com/ScriptReference/Animator.SetTrigger.html [Accessed 9 Jan. 2020].