|  |
| --- |
|  |
| Design document |
| Igus robot arm project |

|  |
| --- |
| Rik van Heesewijk, Luuk Aarts, Dimitar Dyulgerov, Jorn Kersten  6-16-2023 |

Table of Contents

[Introduction 2](#_Toc137803783)

[Unity 2](#_Toc137803784)

[Class diagram 2](#_Toc137803785)

[Scene layout 3](#_Toc137803786)

[TwinCAT 4](#_Toc137803787)

# Introduction

In this document you will find all created designs relevant to our project. One Class diagram was created for Unity, in addition there is also a section on why the objects in Unity are laid out and designed this way.

For TwinCAT, there are a number of state diagrams and....

# Unity

This chapter explains how everything communicates with each other and why the scenes are laid out the way they are. First you see the class diagram, there follows a brief explanation of what talks to each other and what is used next.

A screenshot of a computer screen

Description automatically generated with medium confidence

## Class diagram

Above is the class diagram for Unity, Unity's classes are: ADS, ConveyorBelt, DetectCube, Spawner, DestroyObjects, MainMenu and OptionsPauseMenu. There are 4 classes connected to each other, the only one of those four that calls other classes is the ConveyorBelt class. The ConveyorBelt class reads some variables and methods from the other 3 classes (ADS, DetectCube and spawner).

From ADS a number of methods are called, RobotArmContorler, MoveArmToPosition, ReadDefaultPosition and SendHasNewData with these methods a number of variables are read from the PLC and also variables are set from the PLC. From DetectCube, inTrigger are read to check if there is a cube in the detection area. In addition, objectsInTrigger is also read to read the location of the first cube that is in the trigger. Finally, the variable maxZdetectonArea is read to pass the correct position of the cube relative to the detection area to the robot arm. From Spawner, the ConveyorBelt class sets the boolean doSpawn to true or false so that no more cubes are spawned when the conveyor belt is stopped.

## Scene layout

The scenes are divided into two parts, the main menu section and the game scene. In the main menu scene, a menu was created with a number of input boxes on it. These input boxes are split into two sections, on the left side is all about the conveyor belt and on the right side is all about the spawner.

A screenshot of a computer

Description automatically generated with low confidence  
When start is clicked, the game scene is loaded. A deliberate choice was made here to spawn cubes, this is a simple object that can be moved without rolling or making strange movements. The conveyor belt has been kept simple so that the full focus can be on its functionality. In addition, a colour has been given to the conveyor belt so that it is clearly different from the rest of the scene.  
The terrain was also deliberately left simple, it is only there so that it looks a little neater as a floating conveyor belt.  
A picture containing sky, screenshot, design, table

Description automatically generated

# TwinCAT