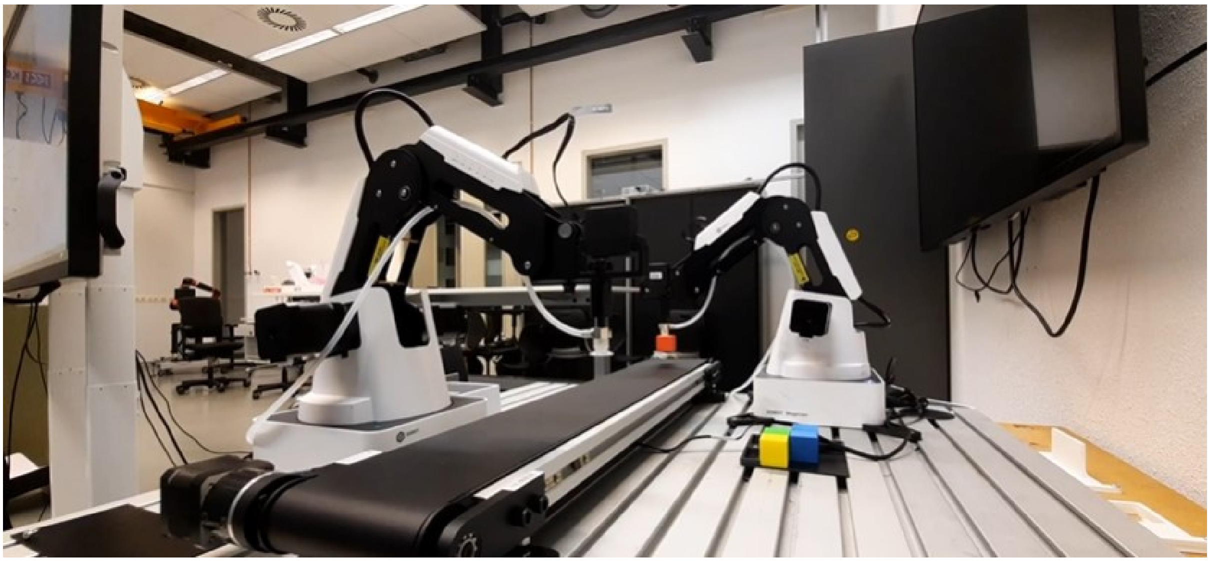
Software Design & Architecture

Final Assignment

Introduction



For the final assignment, to your group of students one Dobot, one camera and a part of a conveyor belt is provided. Your group should be able to use computer vision to pick up a geometry from a picking place and place it on the conveyor belt (or the other way around).

Prerequisites

You should have followed the course. You should have studied the course material and you should have made the exercises provided on Canvas. You should have understood UML, OOP, and vision. You should read the README.md file in the SDA3_final_assignment.zip, provided in Canvas.

Loading Assignment

For the final assignment, to your group of students one Dobot, one camera and a part of a conveyor belt is provided. Your group should be able to use computer vision to pick up a color shape **from a picking place (loading area) and place it on the conveyor belt.**

Requirements:

1. You must perform some user stories (at least 1).
2. You must implement a class diagram of your system.
3. You must implement at least two sequence diagrams for either your detection, localization, or picking process.
4. You must implement a state machine for your entire detection, localization and picking process (the entire program), hint: (pytransitions/transitions: A lightweight, object-oriented finite state machine implementation in Python with many extensions (github.com)).
5. Your implemented code must represent your UML design, in other words your implemented classes must be the same as the classes in your class diagram.
6. You must implement computer vision (OpenCV) that recognizes at least three different color-shapes (where different means: different color, or different shape, or both) placed on a certain picking location (black surface, at your side of the conveyor belt).
7. You must transform the detected shape from the pixel coordinates into the dobot coordinates (mm).
8. You must be able to pick the detected shape with the conveyor loading dobot.
9. You must place the picked shape on the conveyor belt.
10. You should be able to move the placed shape with the conveyor belt to the opposite side.
11. You could design a UI (User Interface), where the possible shapes to be picked are shown and the user can select one for the entire process.
12. You could let the UI randomly (color and shape) select which shapes (at least three) must be placed on the loading area and from those the user must be able to choose one of them in the UI.

Unloading Assignment

For the final assignment, to your group of students one Dobot, one camera and a part of a conveyor belt is provided. Your group should be able to use computer vision to pick up a color shape **from a picking place on the conveyor belt to the unloading area at the right of the conveyor belt.**

Requirements:

1. You must perform some user stories (at least 1).
2. You must implement a class diagram of your system.
3. You must implement at least two sequence diagrams for either your detection, localization, or picking process.
4. You must implement a state machine for your entire detection, localization and picking process (the entire program), hint: (pytransitions/transitions: A lightweight, object-oriented finite state machine implementation in Python with many extensions (github.com)).
5. Your implemented code must represent your UML design, in other words your implemented classes must be the same as the classes in your class diagram.
6. You must implement computer vision (OpenCV) that recognizes at least three different color-shapes, that are placed on the conveyor belt (where different means: different color, or different shape, or both).
7. You must transform the detected shape from the pixel coordinates into the dobot coordinates (mm).
8. You must be able to pick the detected shape with the conveyor unloading dobot.
9. You must place the picked shape on the unloading area at the right of the conveyor belt.
10. You should be able to move the placed shape with the conveyor belt to the opposite side.
11. You could design a UI (User Interface), where the possible shapes to be picked are shown and the user can select one for the entire process.
12. You could let the UI randomly (color and shape) select which shapes (at least three) must be placed on the conveyor belt for the unloading robot to pick and place on the unloading area, and from those the user must be able to choose one of them in the UI.

Deliverables

For this assignment, all of the UML deliverables have to be delivered in any of the following formats: .pdf, .png, .plantuml, .plant, .uml, .iuml, .puml, or .pu. From the deliverables described below, you must get at least 55% and perform the video (deliverable number 9) to be invited to the oral assessment.

1.	At least one user story (UML).	5%
2.	At least one Class Diagram of your system (UML).	15%
3.	At least two Sequence Diagrams (UML).	5%
4.	At least one State Machine Diagram for the entire process (UML).	20%
5.	Your Python code shared in GitHub and Canvas (For those, who cannot use GitHub, an individual report must be written explaining his/her contribution in the assignment).	15%
6.	Proof of a working dobot loading or unloading (picking and place) solution, hint: it can be done blindly (video).	10%
7.	Proof of a working solution (video), where the shape is detected.	10%
8.	Proof of a working solution (video), where the shape is located for the dobot coordinate system. The proof must show a comparison between the real millimeters from the dobot location to the shape location and the calculated transformation in millimeters (Be aware! For getting this percentage/points, you cannot position the object in the same location always).	5%
9.	A video showing your entire solution.	Required
10.	A video, illustrating moving the shape on the conveyor belt.	2%
11.	A video, illustrating the UI, where the user can select one shape to be picked and that shape must be used in deliverables 6, 7, and 8. In other words, it must be integrated in your entire solution.	10%
12.	A video, illustrating that the UI randomly select which shapes (at least three) must be placed on the conveyor belt for the unloading robot to pick and place on the unloading area, and from those the user must be able to choose one of them in the UI. Show and explain the code, that performs this, in the video.	3%

Note: If your deliverable number 9 shows the entire process and solution, you can skip making a video for deliverable numbers 6 and 7.