Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Spring 2022

Team Kratos
Tic-Tac-Toe Robot

Ayush Shrestha
Nabin Raj Bista
Che Justus Kwanga
Yan Wang

Revision History

Revision	Date	Author(s)	Description
0.1	04.16.2022	AS, NRB, CJK, YW	document creation
0.2	04.17.2022	NRB, CJK, YW	complete draft
0.3	04.19.2022	AS, NRB, CJK, YW	draft improvisation
1.0	04.22.2022	AS	official release

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

This robotic arm product is designed to play tic-tac-toe with humans. A gripper is attached to the main body of the robotic arm, which can hold a marker pen to write and play tic-tac-toe. It also holds an eraser so that it can erase the board and start over with a new game.

All the layers of the robotic arm are designed to ensure that it can draw the grid lines required for the game on the glass, playing the game of tic-tac-toe with the user from one side of the glass. During the game, it uses the built-in algorithm to randomly decide the order of tic-tac-toe games, and take strategic actions in turn to play against its human opponent.

2 System Overview

2.1 HMI Layer

The Human-Machine Interaction (HMI) layer is in charge of relaying information between the robot and human commands. It should be able to receive a start command to start the game. It should be able to turn on the LED with an appropriate color to indicate whose turn it is currently. It communicates in a two way form with the logic layer.

2.2 Vision Layer

The Vision layer is a two way communication layer of the robot in charge of translating positions of the board at the request of the logic layer. It will take a picture, convert this picture to text and send the information on the current state of the board to the logic layer for interpretation and for the system to make the next appropriate move.

2.3 Logic Layer

This layer is in charge of the overall game logic in deciding the move to make and the position that the robotic arm should move to. It also in charge of drawing the grid at the start after a command from the HMI layer. It is also in charge of deciding the shapes to be drawn and deciding at what point the game is over. It communicates with the navigation, vision and HMI layer in two way fashion depending on the current demands of the robot.

2.4 Navigation Layer

This is a two way communication layer in charge repositioning the robotic arm based on the command from the logic layer. It should be able to readjust the robotic arm in the x,y and z planes to facilitate three dimensional movement around the board.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section has the breakdown of the layer abstraction of all the layers. Below is the graphical representation of the logical subsystem and how it interacts between the system. In the Tic-Tac-Toe robot we will need human-machine interaction which is a very important step. The logical data elements that flow between the subsystems are explicitly defined.

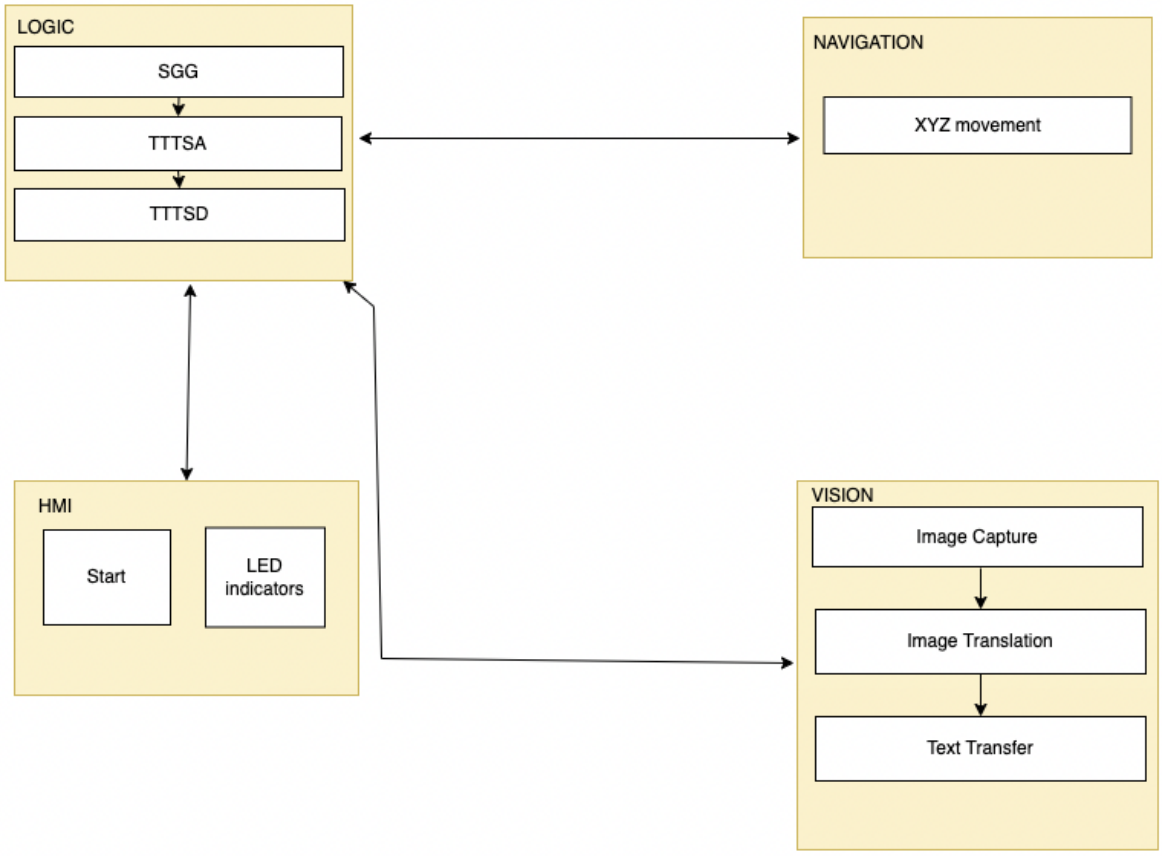

Figure 1: data flow diagram

4 HMI Layer

In this section, the Human Machine Interaction layer is described in details. As the main aim of our robot is to interact with humans and play interactive games, this layer is one of the most important parts.This layer and logic layer depend on each other for smooth functioning of the machine.

4.1 Start Command subsystem

This subsystem will be in charge of receiving the start command from the robot controller and sending it to the logic layer so that the game can start.

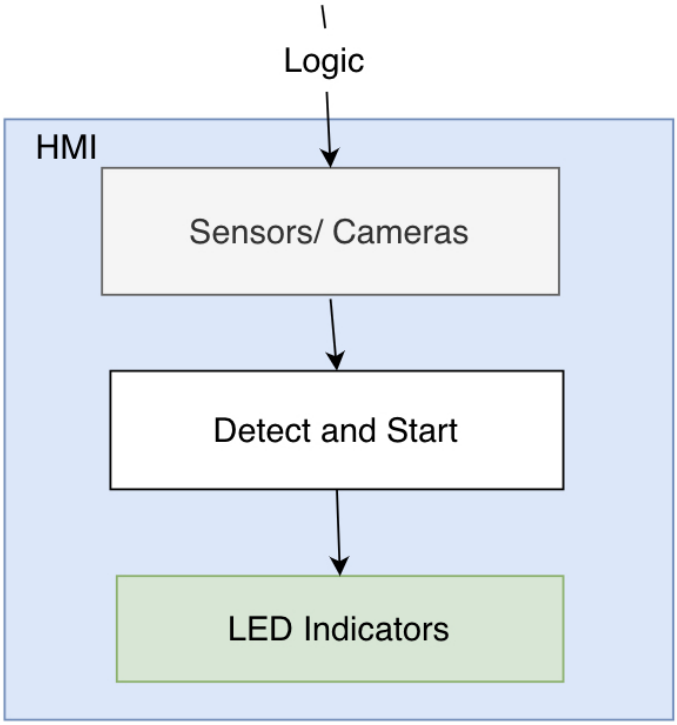

Figure 2: HMI Layer

4.1.1 Assumptions

It should be able to receive the start command from the robotic controller. It should be able to relay the information to the logic layer to start the game.

4.1.2 Responsibilities

Receive command from the robotic controller. Relay information to the logic layer.

4.1.3 Subsystem Interfaces

Table 2: Start interfaces

ID	Description	Inputs	Outputs
#01	Receive Command from robotic controller	Start Command	N/A
#02	Relay to the logic layer	N/A	Start Command

4.2 Human Detection System

The Human Detection System of the robot arm will be able to detect humans in the surrounding and start playing with them. The main aim of the system is to put the machine in stand-by method in case there is no one to play with.

4.2.1 Assumptions

This subsystem assumes that the instruction sent are reliable and realizable.

4.2.2 Responsibilities

The Human Detection System will be using a sensor/camera to sense any humans standing in front of the game board. After the robot senses the human, it should be able to start the game. Similarly, the robot should automatically go to standby mode if it does not detect any humans. It should be able to prevent unnecessary machine movement and utilization of power.

4.2.3 Subsystem Interfaces

Table 3: Camera/ Sensor interfaces

ID	Description	Inputs	Outputs
#1	New instructions for arm movement by sensor/camera	Instructions	Arm Movement

4.3 Human-Machine turn Indicator Lights

Th Indicator lights will be able to indicate whose turn it is to play the game. As it not fixed that who plays first, the machine will randomly pick whose turn it is. After that, the lights will indicate whose turn it is. Here, red light will indicate it is the robot's turn to play while green light will indicate it is the human's turn to play. The main aim of this is to avoid confusion between the players.

4.3.1 Assumptions

The robot will indicate green for the player's turn and red for its own turn.

4.3.2 Responsibilities

The Human-Machine turn indicator light is responsible to avoid confusion between the robot and the player. Most of the times in Tic-Tac-Toe it conflicts who goes first. To avoid this, the machine will be picking it randomly to avoid biases.

4.3.3 SUBSYSTEM INTERFACES

Table 4: LED interfaces

ID	Description	Inputs	Outputs
#1	Decision on which light to turn on	Random picking at first then player and machine in turns	Green/Red light

5 VISION LAYER

In this section, the vision layer is described in detail with the explanation of its subsystems. Each of the subsystems is in a separate subsection of this chapter. The Vision layer is a two way communication layer of the robot in charge of translating positions of the board at the request of the logic layer. It will take a picture, convert this picture to text and send the information on the current state of the board to the logic layer for interpretation and for the system to make the next appropriate move.

5.1 IMAGE CAPTURE SUBSYSTEM

This subsystem will capture the state of the board at the time of request from the logic layer. It then relays this image to the image translation subsystem within the vision layer.

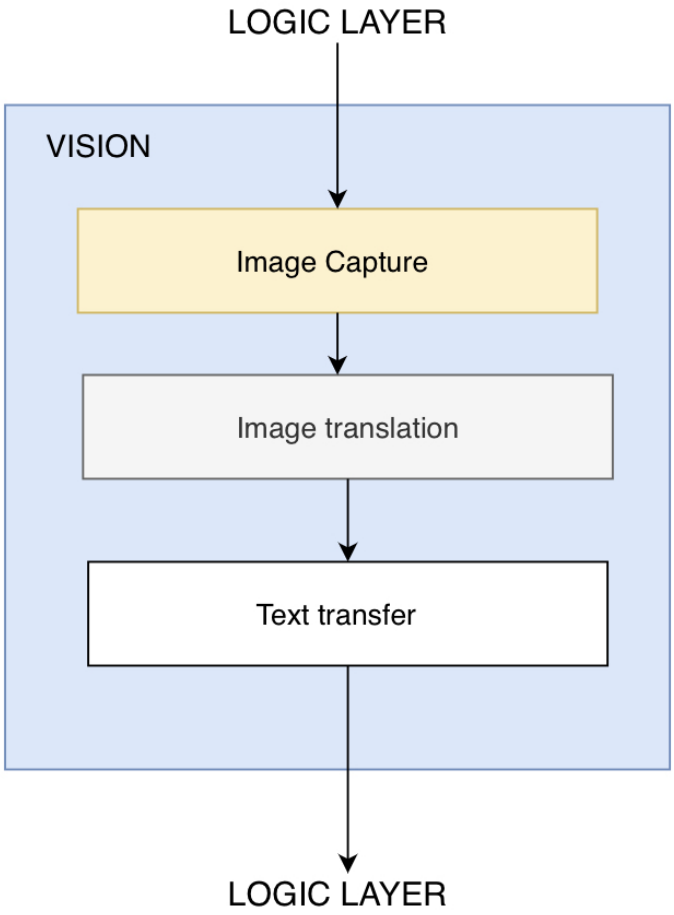

Figure 3: Vision Layer

5.1.1 ASSUMPTIONS

The camera will be able to take at least a black and white photo of the board. It should be able to do it in a short frame of time. It will be able to receive commands from the logic layer and respond.

5.1.2 Responsibilities

Capture the state of the board. Relay the information to the image translation subsystem. Operate on request from the logic layer.

5.1.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Capture state of Board	Logic command	Board Image
#02	Relay to Image Translation subsystem	N/A	Board Image

5.2 Image Translation

This subsystem will receive an image from the Image Capture subsystem and will be responsible for it's conversion into useful information in the form of text that the logic layer can utilize.

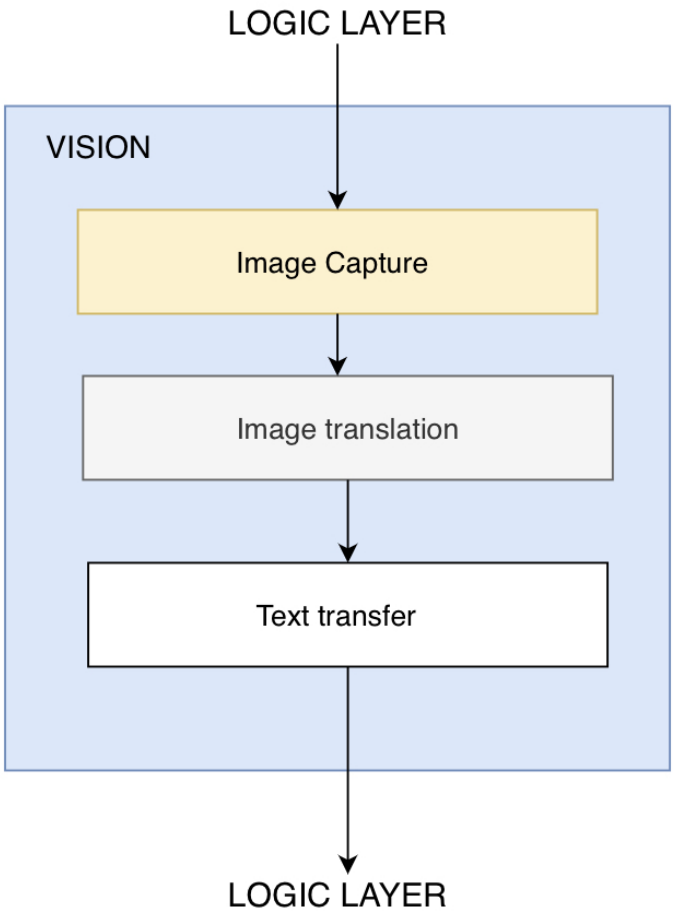

Figure 4: Vision Layer

5.2.1 Assumptions

The camera will be able to carry out this conversion into text as a functionality of the software that will be used with it. It should accomplish this in a short frame of time for fast processing. It should be able to translate into an appropriate text format use able by the logic. It should then be able to transfer this information to the information transfer subsystem.

5.2.2 Responsibilities

Translate the image into text information for the logic layer to interpret. Relay information to the information transfer subsystem.

5.2.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Image Translation	Board Image	Text information
#02	Relay to Information Transfer subsystem	N/A	Text Information

5.3 Information Transfer

This subsystem will be in charge of receiving information from the Image Translation as text and transferring this text to the logic layer over the appropriate communication protocol and utilizing an efficient baud rate.

5.3.1 Assumptions

The camera should have the ability to utilize an appropriate communication protocol. It should transfer the data as received from the image translation without any changes during transfer. It should be able to utilize an adequate baud rate

5.3.2 Responsibilities

Receive text from the Image Translation subsystem. Relay the information to the logic layer. Relay the information over an appropriate communication protocol and using an appropriate baud rate.

5.3.3 Vision Interfaces

Table 7: Vision interfaces

ID	Description	Inputs	Outputs
#01	Receive text from Image Translation	Text Information	N/A
#02	Relay to Logic layer	N/A	Text Information
#02	Utilize an appropriate communication protocol and baud rate	N/A	N/A

6 Logic Layer

The Logic Layer is the most basic module among all the modules of the robot arm. It plays the role of the core brain as robot, and is responsible for the tic-tac-toe game algorithm, how to draw game marks, and how to communicate and coordinate the other three layers. The Logic Layer has three subsystems: setup game grid subsystem, tic-tac-toe strategy algorithm subsystem, and tic-tac-toe symbol drawing system.

6.1 Setup game grid subsystem

Setup game grid subsystem obtains permission through the HMI Layer after the robot receives the command from human, and it then starts the Navigation Layer to help robotic arm move to the front of the glass plate in certain location, and uses the algorithm written inside the setting game grid subsystem to draw a game grid consisting of two vertical and two horizontal lines on the glass plate.

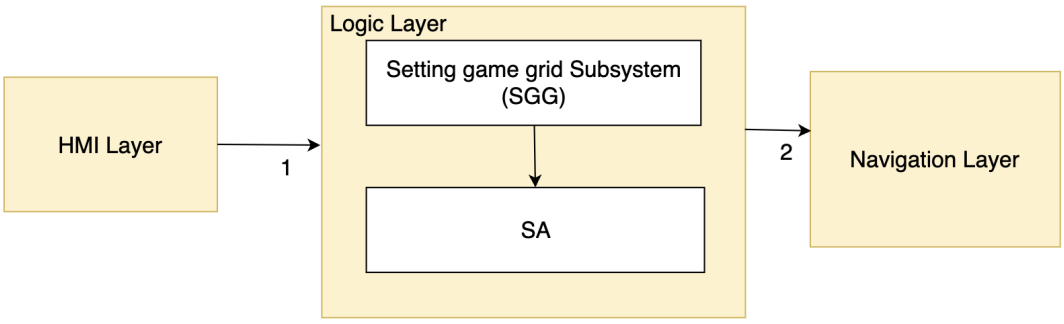

Figure 5: Example subsystem description diagram

6.1.1 Assumptions

Setup the game grid subsystem starts only after receiving the program start command forwarded by the HMI Layer. Its startup also marks the beginning of work on the Logic Layer.

6.1.2 Responsibilities

The initial command to start the tic-tac-toe game is sent by human to the HMI Layer, and then HMI Layer forwards the command to the Logic Layer. Specifically, setup game grid subsystem is set to be executed first. Setup game grid subsystem then wake up the Navigation Layer to move to a specific xyz space point, and use a marker to lay out two horizontal and two vertical game grids on the glass plate.

6.1.3 Subsystem Interfaces

Table 8: Setup Game Grid Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Start command from HMI Layer	input 1	N/A
#2	Initiation bus to Navigation Layer	N/A	output 2
#3	Confirmation to tic-tac-toe strategy algorithm subsystem	N/A	output 3

6.2 TIC-TAC-TOE STRATEGY ALGORITHM SUBSYSTEM

Tic-tac-toe strategy algorithm subsystem is the core subsystem of the Logic Layer. It randomly determines the player of the first round after the game starts, coordinates the Navigation Layer and the Vision Layer, and make marks on board through the tic-tac-toe symbols drawing system to truely complete the task of playing tic-tac-toe with human.

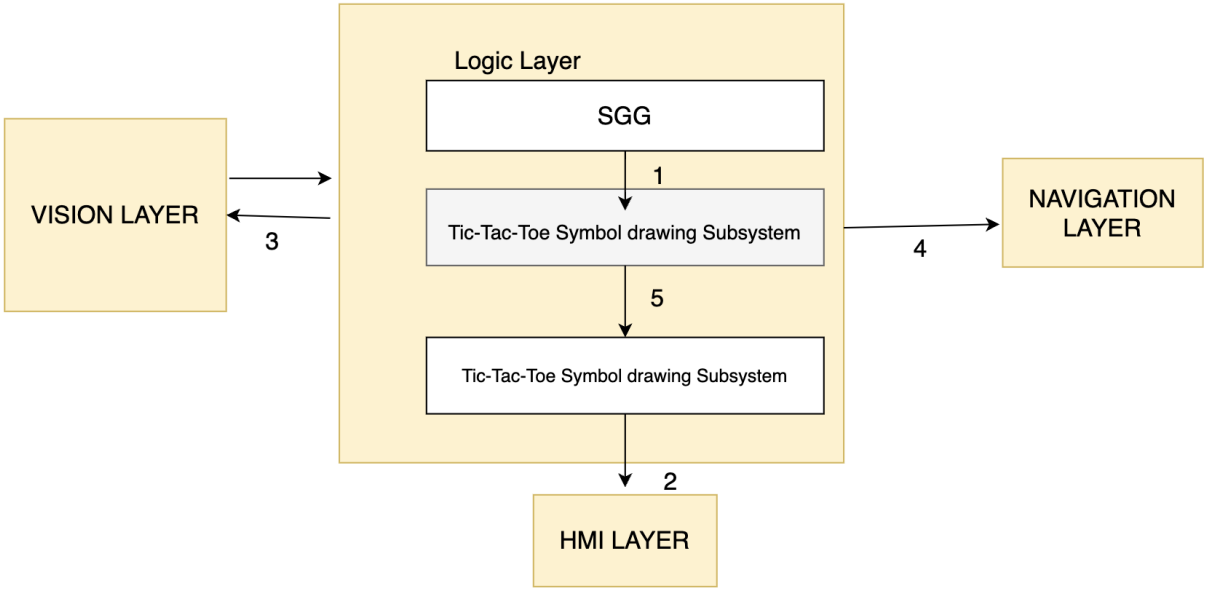

Figure 6: Example subsystem description diagram

6.2.1 ASSUMPTIONS

Feedback from setup game grid subsystem is a signal for the activation of the tic-tac-toe strategy algorithm subsystem.

6.2.2 RESPONSIBILITIES

After setup game grid subsystem completes its task, a feedback is sent from the setup game grid subsystem to the tic-tac-toe strategy algorithm subsystem. The tic-tac-toe strategy algorithm subsystem will determine if the first turn of the game is taken by the machine side or the human side. We assume that the tic-tac-toe strategy algorithm subsystem decides that the human side will play the first turn. In this case, the tic-tac-toe strategy algorithm subsystem will communicate with the HMI Layer through the Logic Layer, indicate that the human draws the first mark by lighting a green LED, and tell the Vision Layer to monitor changes on the game board , which converts the collected picture information into data that can be processed by Logic Layer. When the tic-tac-toe strategy algorithm subsystem determines that its turn is coming, it will determine the strategy for the next round through the built-in tic-tac-toe algorithm, move the position of the robotic arm through the Navigation Layer, and enable the tic-tac-toe symbol drawing subsystem to complete the action.

6.2.3 Subsystem Interfaces

Table 9: Setting Game Grid Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Feedback from setup game grid subsystem	input 1	N/A
#2	Command to HMI Layer	N/A	output 2
#3	Command to Vision Layer	input 2	output 3
#4	Command to Navigation Layer	N/A	output 4
#5	Command to tic-tac-toe symbol drawing subsystem	input 3	output 5

6.3 Tic-Tac-Toe Symbol Drawing Subsystem

The tic-tac-toe symbol drawing subsystem is a combination of pixels and algorithms. Its task is only to record the corresponding symbols that should be marked by its own turn at the beginning of each game, and to draw the mark at a specific position when the tic-tac-toe strategy algorithm subsystem sends order.

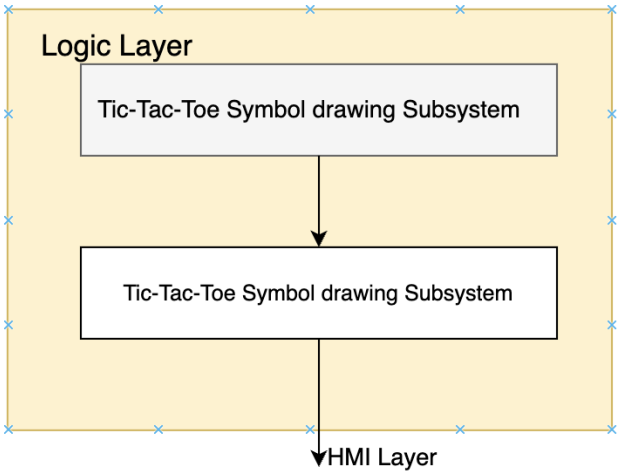

Figure 7: Example subsystem description diagram

6.3.1 Assumptions

The tic-tac-toe symbol drawing subsystem draws after getting the command of the tic-tac-toe strategy algorithm subsystem, and prompts the current player of the game round through the HMI Layer.

6.3.2 Responsibilities

After the tic-tac-toe strategy algorithm subsystem determines that its own round has started, of course, after moving to a predetermined position through the Vision Layer and Navigation Layer, it sends the drawing command to the tic-tac-toe symbol drawing subsystem. The tic-tac-toe symbol drawing subsystem then begins drawing with the stored symbol type, cross or circle. During this process, the HMI

Layer will be coordinated to tell the human side not to take any action until the robot side's action is over.

6.3.3 Subsystem Interfaces

Table 10: Setting Game Grid Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Feedback from tic-tac-toe strategy subsystem	input 1	output 1
#2	Command to HMI Layer	N/A	output 2

7 Navigation Layer

This layer mainly deals with navigation of robotic arm. Our robotic arm is able to move in all direction in 3D space.

7.1 Grid Drawing

Once the game starts robotic arm navigate in a location where it starts to draw grid using the pen attached to the gripper of the robotic arm.

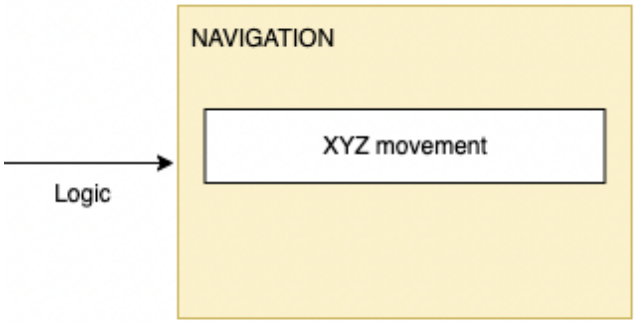

Figure 8: Grid drawing subsystem description diagram

7.1.1 Assumptions

In this subsystem we assume that the glass screen is in a fixed position and robotic arm will always be able to draw grid in same location every time.

7.1.2 Responsibilities

We cannot start to play game until and unless our robotic arm draw grid in glass. Therefore, grid is needed at the beginning of every game.

7.1.3 Navigation Interfaces

Input is provided when the user is ready to play game which is generally recognized by vision layer. Once, grid drawing is completed, it send the output to logic layer to start the game.

Table 11: Navigation interfaces

ID	Description	Inputs	Outputs
#01	Interface between robotic arm and player	input from vision layer	output to logic layer

7.2 Game mode

This subsystem help to navigate robotic arm during game in proper coordinate to draw 'X' or 'O' so that it can play game. Logic layer helps to navigate through out the game.

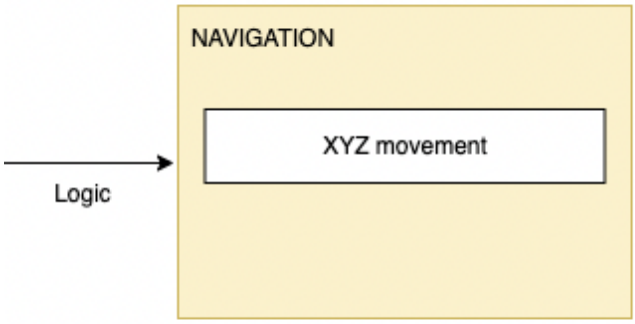

Figure 9: Game mode subsystem description diagram

7.2.1 Assumptions

In this subsystem we assume that robotic arm to move and draw inside the grid. We also assume that once game is finished it erase the glass and be ready for next game. This sub-system is assume to find the best possible grid to draw so robot can be winner.

7.2.2 Responsibilities

The main responsibilities of this subsystem is to keep the game flowing. This subsystem should draw clearly and in proper grid location so it can be easier to track and take input of the game by the vision layer.

7.2.3 Subsystem Interfaces

This subsystem interface with both vision and logic layer.

Table 12: Game interfaces

ID	Description	Inputs	Outputs
#01	Interface between logic layer and navigation layer	input from logic layer	N/A