DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2022

TEAM
CheckUR5
SENIOR DESIGN
FALL 2022

TEAM CHECKUR5
CHECKERS-PLAYING UR5 CO-BOT

NIMITA UPRETY
PATRICIA ROJAS
HOANG HO
KEVIN VU
JOANNA HUYNH

Revision History

Revision	Date	Author(s)	Description
0.1	10.31.2022	HH	document creation
0.2	11.13.2022	NU, HH, PR, KV, JH	complete draft
1.0	11.14.2022	NU, HH, PR, KV, JH	official release
1.1	05.09.2023	NU	final revisions
1.2	05.10.2023	NU, HH, PR, KV, JH	final revisions

Contents

List of Figures

List of Tables

1 Introduction

This section describes the purpose, use & intended audience for our programmed UR5 collaborative robot (co-bot). The UR5 co-bot will be programmed to be able to play the interactive & strategy-based game, checkers, against a human opponent. The robot will have an electropermanent magnetic attachment that will allow it to pick up and move the individual checkers pieces. The robot is not intended to beat the human opponent every time but rather just have the ability to play a full game of checkers against them.

1.1 Purpose and Use

Our project aims to showcase the UR5 co-bot's abilities as a means of promoting the UT Arlington College of Engineering to prospective students through the means of an interactive demonstration. Students visiting the College of Engineering are expected to be able to engage with the product in a game of checkers, which will provide the potential Mavericks with an enjoyable and educational experience.

1.2 Intended Audience

The intended audience of our product is the UT Arlington College of Engineering, Department of Computer Science & Engineering as well as the university as a whole to utilize the co-bot as a marketing strategy for prospective students. If this product were to be made commercially available it would be feasible and our primary customers would be other universities aiming to also recruit future engineers. Although the UR5 co-bot is intended to be operated by a student or faculty member of the university, the end user is a member of the public as the project is intended for general use.

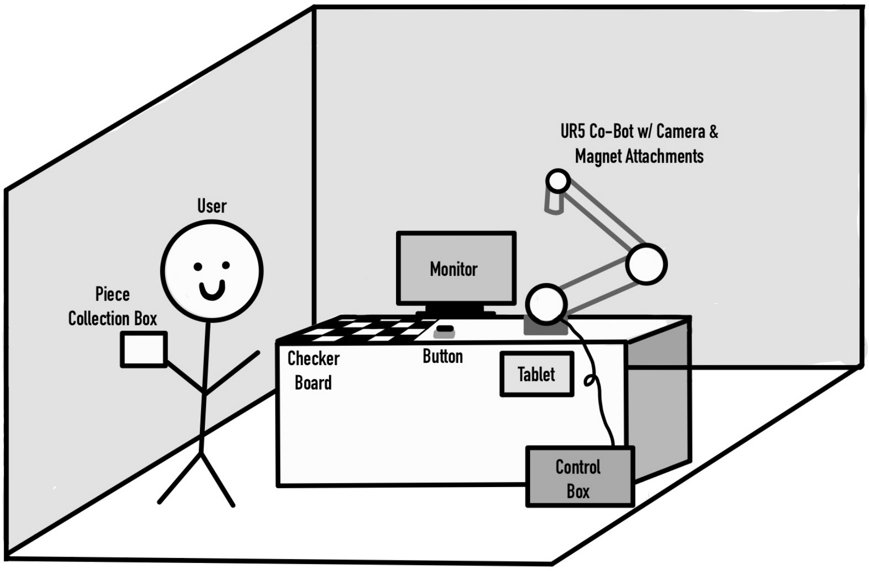

Figure 1: Conceptual Drawing

2 System Overview

The system for our product is currently simplified into 4 layers and 3 high-level components. Each layer/component will interact with one another in different ways.

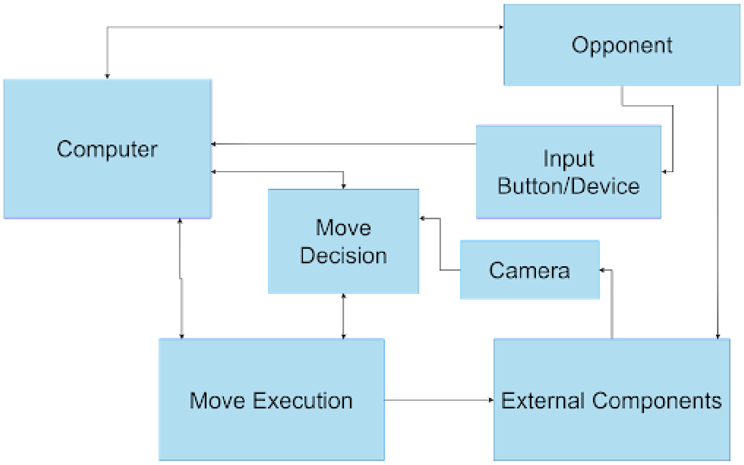

Figure 2: A simple architectural layer diagram

2.1 Opponent Description

The Opponent is the person that will be going against the robot during the checkers game. The opponent will interface with the Input Button/Device component, Computer layer, and External Components layer in order to progress and play checkers with the robot arm and make game moves.

2.2 Input Button/Device Description

The Input Button/Device is one of the ways the opponent will be able to facilitate the progression of the checkers match. When this component is pressed, it will notify the Computer layer that the human opponent has finished making their move.

2.3 Camera Description

The camera is the component that acts as the eyes into the real world for our system. The camera acts as a bridge that sends the AI layer the necessary information that it needs about the current state of the checkers board.

2.4 Move Decision Layer Description

The Move Decision layer is the layer that will make all of the move decisions that the robot will make. This layer consists of Computer Vision and AI. After receiving data about the current state of the board from the Camera, the Move Decision layer will formulate a move decision that will be sent to the Computer layer.

2.5 Computer Layer Description

The Computer layer is the central piece of our system. This layer contains the HUD, GUI, and Software components of our system. This layer acts as the facilitator of the checkers match, and receives input from the Opponent to progress the match, sends signals to the AI layer to grab board state and make a move, and signals the UR5 Robot Arm layer to execute those moves.

2.6 Move Execution Layer Description

The Move Execution layer is the layer that will physically execute the moves that our system will make during a checkers match. This layer consists of the Magnetic Gripper, the Arduino, and the UR5 Robot Arm. After receiving move data from the Computer layer, the Move Execution layer will interface with the components in the External Components layer to make its next move.

2.7 External Components Layer Description

The External Components layer is composed of all of the physical checkers board game pieces. Both the Opponent component and the Move Execution layer physically interact with the External Components layer when they have made their move decisions to progress the match. This layer is made up of the Playing Board, the Collection Box, and the Checkers Pieces.

3 Subsystem Definitions & Data Flow

Figure 3 describes the layers that the Checkers-playing co-bot comprises of and the subsystems that make up each layer.

The Computer layer, Move Decision layer, Move Execution layer, and External Components layer have subsystems within their layers. While these layers interface each other, the subsystems within them also interface with each other. Within the Computer layer, the software works with both the HUD and GUI to implement the user interface. Within the Move Decision layer, the computer vision component communicates with the AI to train and interpret information that is being captured by the camera. Within the Move Execution layer, the Arduino controller powers and instructs the magnetic gripper when it needs to be activated and deactivated. Within the External Components layer, the checkers pieces are manipulated across the playing board and disposed into the collection box.

As a whole, the Computer layer communicates with the opponent layer, the Move Decision layer, the Move Execution layer, and the input button/device layer. This is because the robot control system is a part of the software and the opponent interacts with the information displayed by the GUI. The Opponent layer also interfaces with the input button layer and External Components layer when it is their turn during the match. The Move Execution layer also interfaces with the External Components layer during the match. The Move Decision layer and the External Components layer interfaces with the camera where it captures the state of the game board and communicates it to the CV and AI components for data interpretation.

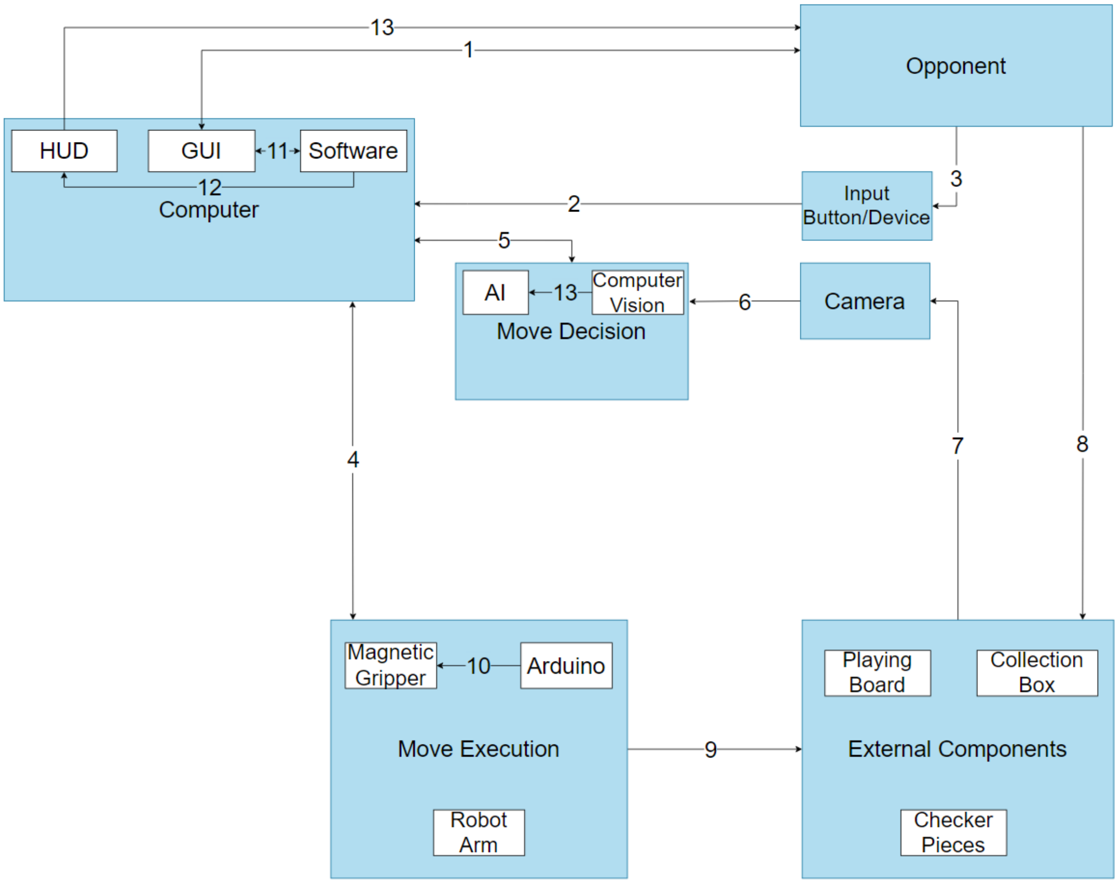

Figure 3: Checkers-Playing UR5 Co-bot architectural layer diagram

4 Move Decision Layer Subsystems

In this section, we describe the move decision layer. This layer is responsible for the decision making process of the UR5 robot for the Checkers game.

4.1 Artificial Intelligence

The Artificial Intelligence communicates with the camera and Computer layer subsystems to make decisions for the UR5 robot arm.

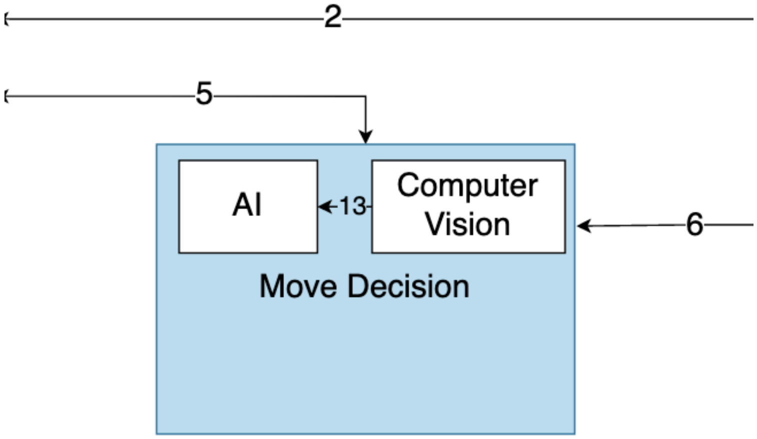

Figure 4: Move decision layer subsystems diagram

4.1.1 Assumptions

N/A

4.1.2 Responsibilities

The Artificial Intelligence calculates the next move for the Computer. It compares the best move by looking at the next two to three steps ahead by the appropriate algorithm. It also determines any invalid moves by the player.

4.1.3 Artificial Intelligence Interfaces

Details in Table 5 show the incoming and outgoing data elements.

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#5	Signal from the computer	N/A	N/A

4.2 Computer Vision

The Computer Vision interacts with the frames received from the camera to determine the player's move.

4.2.1 Assumptions

N/A

4.2.2 Responsibilities

It successfully identifies the movements of the Checkers pieces. It also identifies the current locations on the Checkers board.

4.2.3 Subsystem Interfaces

Details in Table 6 show the incoming and outgoing data elements.

Table 3: Computer Vision interfaces

ID	Description	Inputs	Outputs
#13	Send input to Artificial Intelligence	Computer Vision input	Checkers Move
#6	Receives camera frames	Camera Frames	Checkers board movement

5 Computer Layer Subsystems

In this section, the Computer layer is described in detail. This layer consists of the HUD, GUI, and Software subsystems. Overall, the layer is responsible for communicating with the players information about the state of the game, sending the robot arm the actions it needs to take, and handling the decisions after each move.

5.1 HUD Subsystem

The Heads-Up Display (HUD) subsystem communicates with the Software subsystem of the Computer layer to receive information about the status of the game.

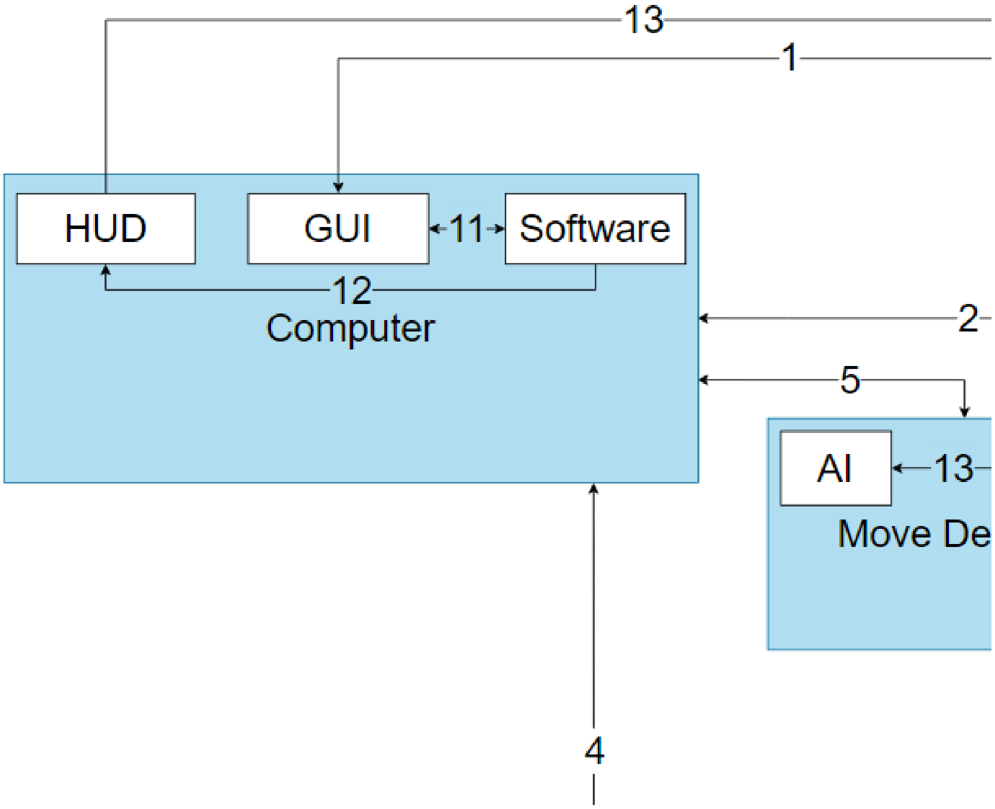

Figure 5: Computer layer subsystems diagram

5.1.1 Assumptions

N/A

5.1.2 Responsibilities

The HUD subsystem displays to the user the status of the game and any information needed for the user. In this case the user is the Opponent layer. Information on the status of the game includes the current scores and a display of the game board on the computer monitor. This information is provided by the Software subsystem. When the physical game board changes, such as when a player moves a piece, this will also be reflected on the screen through the HUD.

5.1.3 Subsystem Interfaces

Detailed in Table 2 are interfaces connected to the HUD subsystem. Incoming and outgoing data elements are detailed.

Table 4: HUD Subsystem interfaces

ID	Description	Inputs	Outputs
#12	Information received from the Software subsystem to display	Game status and board	N/A
#13	Display game status through the Computer to the Opponent	N/A	Game status and board

5.2 GUI Subsystem

The Graphical User Interface (GUI) subsystem is the main interactive component between the Computer and Opponent layers. It communicates with the Software subsystem within the Computer layer for its inputs and outputs.

5.2.1 Assumptions

The GUI only responds to keyboard inputs.

5.2.2 Responsibilities

The GUI is displayed on the monitor and can be interacted with by the use of keyboard keys by the player. These buttons will be able to change the state of the game for purposes such as new game, indicate their end of turn and quit game. Each time the opponent interacts with the GUI by sending a keyboard input for a specific function, this will interact with the computerized version of the game and update the game accordingly.

5.2.3 Subsystem Interfaces

Detailed in Table 3 are interfaces connected to the GUI subsystem. Incoming and outgoing data elements are detailed.

Table 5: GUI Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Opponent interacts with the GUI through keyboard inputs	Keyboard Input	Game state change
#11	Game state changes due to user interaction with the GUI	Keyboard Input	Game state change

5.3 Software Subsystem

The Software subsystem is the main program behind the actions the UR5 Robot Arm will execute. This subsystem not only includes the source code itself, but also software packages needed for the program.

5.3.1 Assumptions

The subsystem is able to send instructions to the UR5 robot arm.

5.3.2 Responsibilities

The Software subsystem is responsible for handling events from the GUI. The subsystem also communicates with the UR5 robot arm through the Computer layer. The subsystem handles data on the status of the robot arm received from the robot arm itself. It also communicates to the robot arm what programmed actions it will need to perform. These actions will be determined by its interactions with the Move Decision layer which consists of the AI subsystem. The Software subsystem is also responsible for handling inputs from the Input Button/Device layer through the Computer layer.

5.3.3 Software Subsystem Interfaces

Detailed in Table 4 are related interfaces to the Software subsystem. Incoming and outgoing data elements are detailed.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#2	Signal that a turn has been completed by the Opponent	Keyboard Input pressed signal	Signal to Move Decision layer
#4	Communication between Move Execution and Computer layers	UR5 robot arm status	Actions to be performed by the UR5 robot arm
#5	Communication between Move Decision and Computer layers	Next move to make	Game state change
#11	Game state changes due to user interaction with the GUI	Keyboard Input signal	Game state change

6 Move Execution Layer Subsystems

In this section, we describe the move execution layer. This layer is responsible for one of the key functionalities of our project which is the UR5 Robotic Arm's movement, which allows for the checkers game to take place between the user and the co-bot. The move execution layer is made up of three subsystems, the UR5 Robot Arm itself, the electromagnetic gripper, as well as an Arduino.

6.1 UR5 Robot Arm

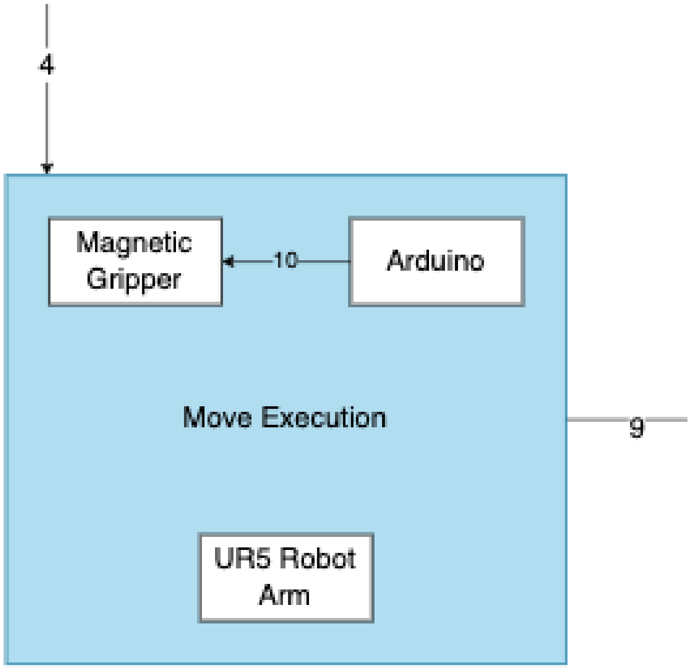

Figure 6: Move execution layer subsystems diagram

6.1.1 Assumptions

Here our biggest assumption is the fact that the UR5 arm has already been connected to the software functionality and therefore can be programmed to make specific moves.

6.1.2 Responsibilities

The UR5 co-bot arm must be able to follow the instructions as asked by the software in order to hover over the correct location on the checker board. The robot must be able to move to fixed locations as predetermined by ROS and our custom software in order to not only move to specified pieces on the board but also to the collection box, and a waiting position.

6.1.3 Subsystem Interfaces

Table 7: UR5 Robot Arm interfaces

ID	Description	Inputs	Outputs
#4	Receive input from computer/software	Location of next move	UR5 Arm will move to expected location

6.2 Arduino

6.2.1 Assumptions

N/A

6.2.2 Responsibilities

The Arduino system has the primary responsibility of being the micro-controller of our magnetic gripper. We will be integrating the Arduino into our software so that we can have the ability to turn our electromagnet on and off whenever we are moving our checkers pieces around.

6.2.3 Subsystem Interfaces

Table 8: Arduino interfaces

ID	Description	Inputs	Outputs
#4	Receive input from computer/software	Signal indicating on/off	Arduino will store on/off signal to send to magnetic gripper
#10	Send signal to magnetic gripper	Stored on/off signal in Arduino	Send on/offsignal to magnetic gripper

6.3 Magnetic Gripper

6.3.1 Assumptions

We will assume that the magnetic gripper has been attached to the "hand" or tip of the UR5 Robot Arm, via a base that has been 3D printed.

6.3.2 Responsibilities

The Magnetic Gripper allows for game play interaction to take place as it will be the component that physically picks up and releases each individual checkers piece throughout the game.

6.3.3 Subsystem Interfaces

Table 9: Magnetic Gripper interfaces

ID	Description	Inputs	Outputs
#10	Receive signal from Arduino	On/off signal from Arduino	Magnetic gripper will turn on/off accordingly

7 External Components Layer Subsystems

In this section, the External Components layer is described in detail. This layer is comprised of the physical parts that make the checkers game, this includes the checkers playing board, the checkers pieces, and the collection box.

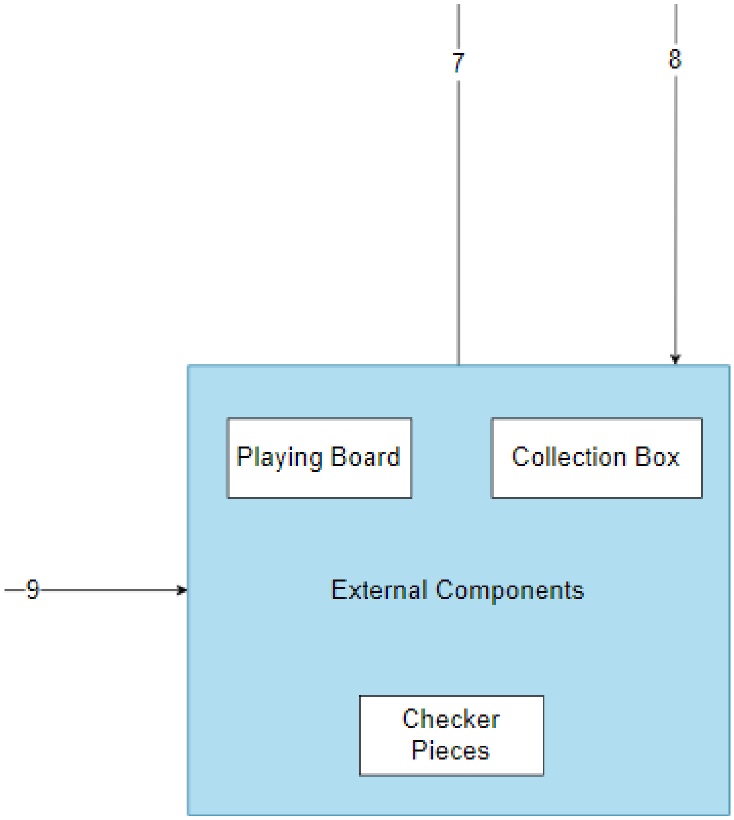

Figure 7: External components subsystem description diagram

7.1 Playing Board

The Playing Board is the physical platform with distinct cells which the checkers game is played upon.

7.1.1 Assumptions

Both the robot arm and human opponent interact with the playing board by moving the checkers pieces.

7.1.2 Responsibilities

The opponent and the UR5 robot arm will be executing the checkers match upon the playing board. The grid on the playing board sets the boundaries for which movements each player can make. The playing board will be a fixed entity on top of the toolbox and serves as the domain that the camera will be focusing on.

7.1.3 Subsystem Interfaces

Table 10: Playing Board interfaces

ID	Description	Inputs	Outputs
#8	Camera focuses on the state of the playing board	N/A	State of the playing board

7.2 Checkers Pieces

The checkers pieces are the focus of the two players as they are the items that will be manipulated during the match.

7.2.1 Assumptions

Both the robot arm and human opponent manipulate the checkers pieces by picking up their respective pieces and moving them to a new location.

7.2.2 Responsibilities

The checkers pieces are the main objects of the checkers match as both the UR5 arm and opponent will be making decisions on the movement of the pieces. Based on these decisions, both players will be making contact with the checkers pieces in order to manipulate their location. These pieces allow a match to take place.

7.2.3 Subsystem Interfaces

Table 11: Checkers Pieces interfaces

ID	Description	Inputs	Outputs
#7	Opponent manipulates the position of a checkers piece	Opponent moves a piece	New location of the checkers pieces
#8	Camera views the pieces on the playing board	N/A	State of a checkers piece
#9	UR5 robot arm and magnetic gripper manipulates the position of a checkers piece	Opponent moves a piece	New location of the checkers pieces

7.3 Collection Box

The collection box stands off to the side ready to hold the checkers pieces.

7.3.1 Assumptions

The collection box is large enough to carry all checkers pieces and is accessible to both the robot arm and human opponent.

7.3.2 Responsibilities

The collection box serves to collect the checkers pieces that are eliminated from the playing board during the checkers match by both the robot arm and human opponent.

7.3.3 Subsystem Interfaces

Table 12: Collection Box interfaces

ID	Description	Inputs	Outputs
#7	Opponent eliminates the robot arm's checkers piece	Checkers piece disposed into collection box	N/A
#9	Robot arm eliminates the opponent's checkers piece	Checkers piece disposed into collection box	N/A