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

DETAILED DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2019

CHECKMATES
READY GO

JORDAN BURNES
RAHME BUTAINEH
SADAT HOSSAIN
MICHAEL SANTELLAN
DONGCHEN YE

Revision History

Revision	Date	Author(s)	Description
0.1	2.11.2020	SH	document creation
0.2	2.23.2020	JB, RB, SH, MS, DY	complete draft

CONTENTS

List of Figures

List of Tables

1 Introduction

"READY GO!" is a system that will allow for a person to play chess against a robot arm. The person will be able to select from various difficulties using a switch that has 3 modes for easy, medium, and hard. The robot will take input from the person using a camera and machine vision to discern the different pieces and their positions. The camera will be mounted on a pole and will only record position once a button is pressed by the user to signify that the person is done moving. Then the GUI provided will tell the user where the "READY GO" system wants to move the next chess piece and the game will continue.

2 System Overview

Our system will consist of four layers that work together to successfully play a game against a human. Our input layer will take input from the user. Our Machine Learning layer will help us understand where pieces are being moved. Our Logic layer will help us organize data into something useful. Finally, our output layer will emit messages in a format that is understandable to people playing against the system. This part will use the GUI and the speaker to explain the logic behind the movement used by the system.

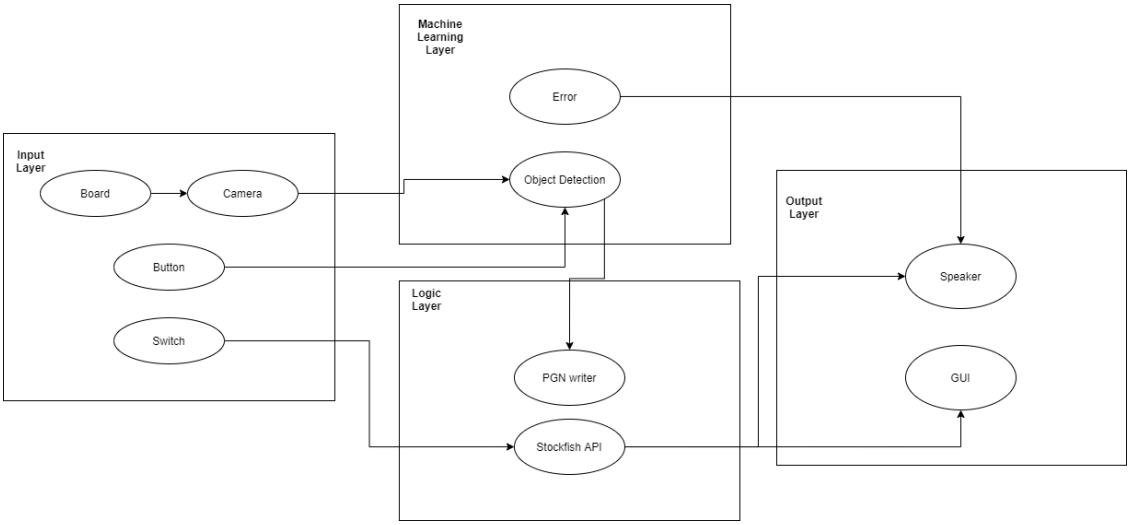

Figure 1: System architecture

3 Input Layer Subsystems

This layer is used for all the input by the user to the "READY GO!" system.The chess board has the pieces on top, the camera is used for image recognition. The button is a key on the keyboard for the GUI, this lets the system know that the user has made a turn. The last part is the switch which is used to choose the difficulty level for the chess algorithm.

3.1 Layer Hardware

This whole layer is just hardware. The board is a generic chess board. The camera is a 4k camera. The Button is a button on the keyboard. The switch is commercial 3 way toggle switch.

3.2 Layer Operating System

None in use

3.3 Layer Software Dependencies

None in use

3.4 Board Subsytem

This is a generic roll out chess board. This is where the user will make the chess moves for themselves as well as instructed by the GUI.

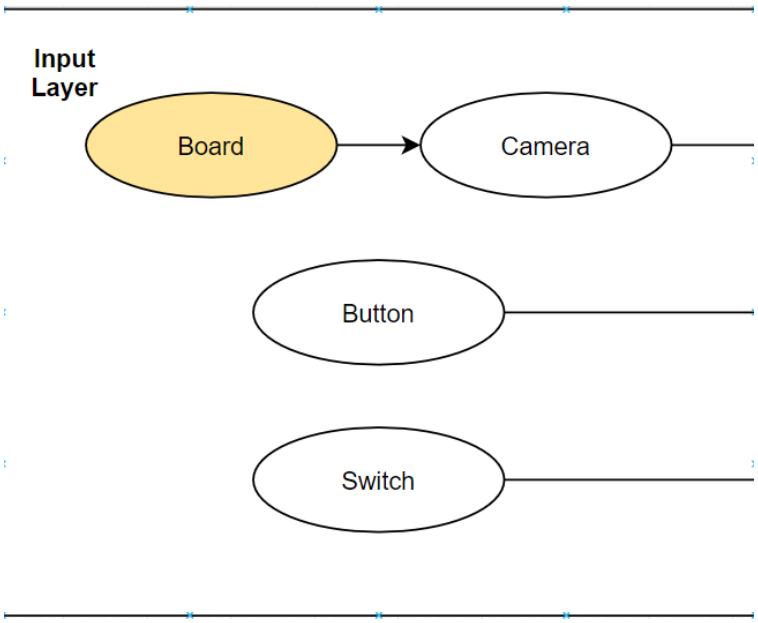

Figure 2: Board subsystem description diagram

3.4.1 Subsystem Hardware

8x8 dimensions with classic black and white pieces.

3.4.2 Subsystem Operating System

Not applicable

3.4.3 Subsystem Software Dependencies

Not applicable

3.4.4 Subsystem Programming Languages

Not applicable

3.4.5 Subsystem Data Structures

Not applicable

3.4.6 Subsystem Data Processing

Not applicable

3.5 Camera Subsystem

A 4K web camera used to capture the board and recognize the pieces and how they have moved

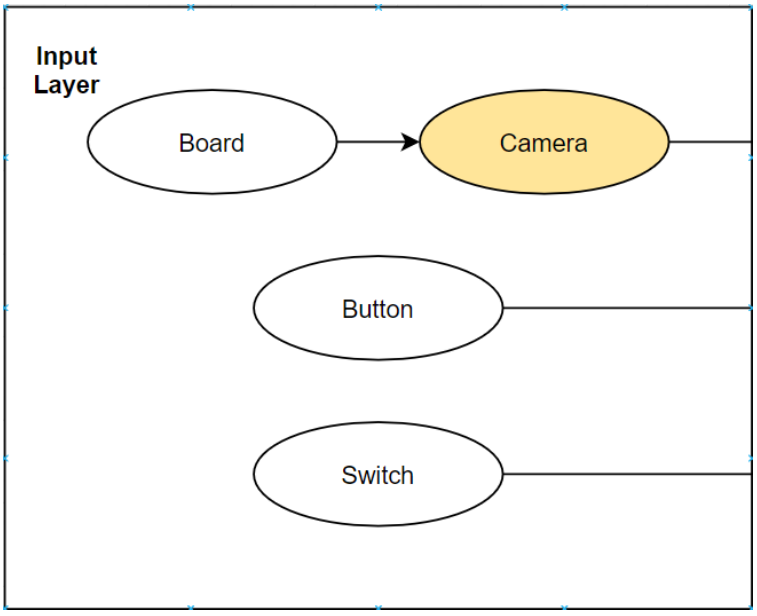

Figure 3: Example subsystem description diagram

3.5.1 Subsystem Hardware

This is a 4k web camera.

3.5.2 Subsystem Operating System

Not applicable

3.5.3 Subsystem Software Dependencies

Not applicable

3.5.4 Subsystem Programming Languages

Not applicable

3.5.5 Subsystem Data Structures

Not applicable

3.5.6 Subsystem Data Processing

Not applicable

3.6 Button subsystem

This is the spacebar or enter Key on the GUI. When pressed it will signify that the user has made their turn.

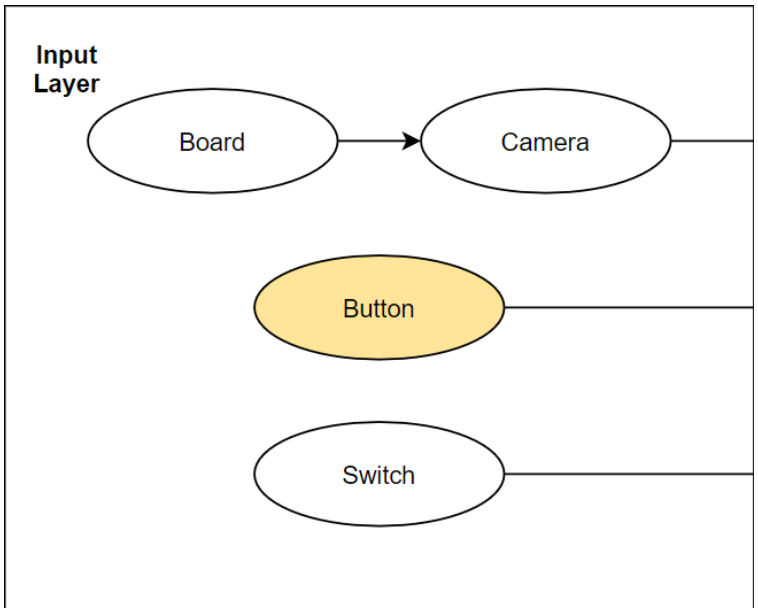

Figure 4: Example subsystem description diagram

3.6.1 Subsystem Hardware

Either the spacebar or the Enter key on the GUI

3.6.2 Subsystem Operating System

Not applicable

3.6.3 Subsystem Software Dependencies

It will require system to be functional and take in the keystroke.

3.6.4 Subsystem Programming Languages

Not applicable

3.6.5 Subsystem Data Structures

Not applicable

3.6.6 Subsystem Data Processing

Not applicable

3.7 Switch Subsystem

This is a switch that indicates the difficulty level for the "READY GO!" system.

3.7.1 Subsystem Hardware

A commercial grade 3 way toggle switch.

3.7.2 Subsystem Operating System

Not applicable

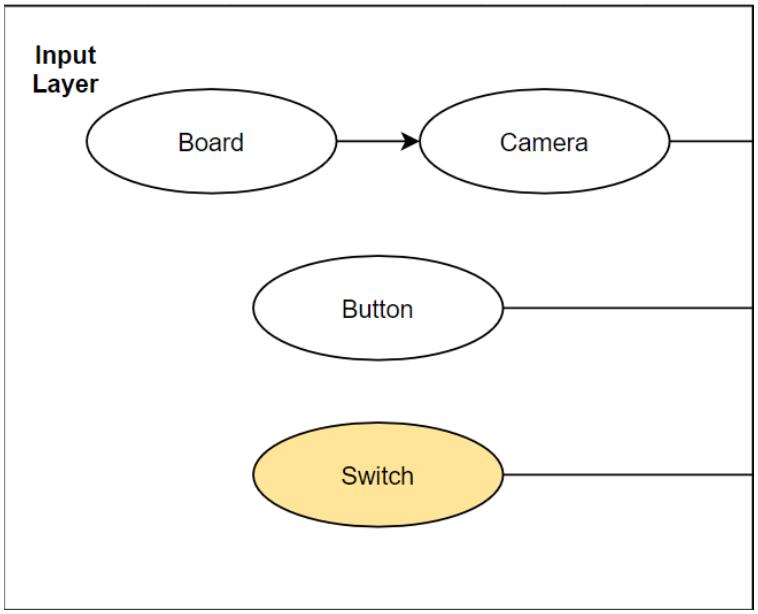

Figure 5: Example subsystem description diagram

3.7.3 Subsystem Software Dependencies

Not applicable

3.7.4 Subsystem Programming Languages

Not applicable

3.7.5 Subsystem Data Structures

Not applicable

3.7.6 Subsystem Data Processing

Not applicable

4 Machine Learning Layer Subsystems

This layer is responsible for the error and object detection. The camera sends the feed to the software to detect the chess pieces and where on the board the pieces are.

4.1 Layer Hardware

The program can run on any computer that is able to handle python,C# and unity.

4.2 Layer Operating System

Windows or Linux operating systems.

4.3 Layer Software Dependencies

The system requires Python and C#.

4.4 Error Subsystem

The error subsystem detects if the user has made an illegal move or has made the wrong move for the system as directed by the GUI.

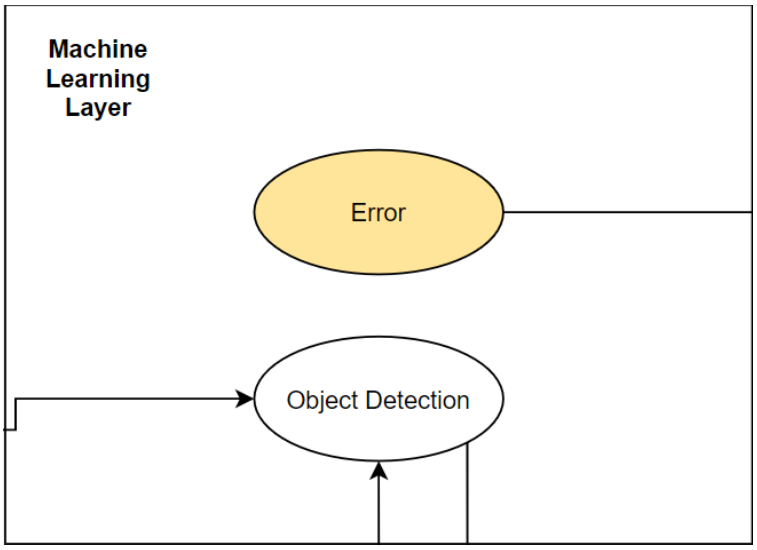

Figure 6: Example subsystem description diagram

4.4.1 Subsystem Hardware

Not applicable

4.4.2 Subsystem Operating System

Windows or Linux systems

4.4.3 Subsystem Software Dependencies

Depends on chess algorithm software to detect the error.

4.4.4 Subsystem Programming Languages

Python and C#

4.4.5 Subsystem Data Structures

Error classes that are made by the chess algorithm.

4.4.6 Subsystem Data Processing

Stockfish API does the error processing

4.5 Object detection

This part is responsible for distinguishing between pieces, and black and white squares.

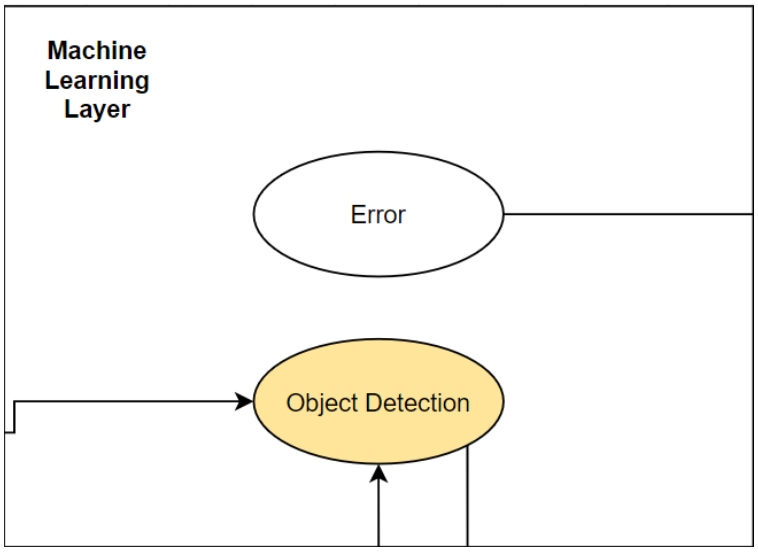

Figure 7: Example subsystem description diagram

4.5.1 Subsystem Hardware

Not applicable

4.5.2 Subsystem Operating System

Windows or Linux

4.5.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

4.5.4 Subsystem Programming Languages

Python and C#

4.5.5 Subsystem Data Structures

Classes for the pieces and how to recognize them from the pictures.

4.5.6 Subsystem Data Processing

Using machine learning to teach the system what the pieces look like from the top view.

5 Logic Layer Subsystems

The logic layer is the main part of the system. This is where the chess algorithm decides what moves to make based on information it gets from the object detection subsystem and the camera feed. Once the decision has been made it will send the information to the GUI.

5.1 Layer Hardware

Not applicable

5.2 Layer Operating System

Windows and Linux

5.3 Layer Software Dependencies

Stockfish API required.

5.4 PGN writer

This subsystem will be responsible for structuring the game data into PGN format for the purpose of easy parsing and generation by computer programs. The output PGN code consist of a set of "tag pairs" (game information), followed by the "movetext" (chess moves).

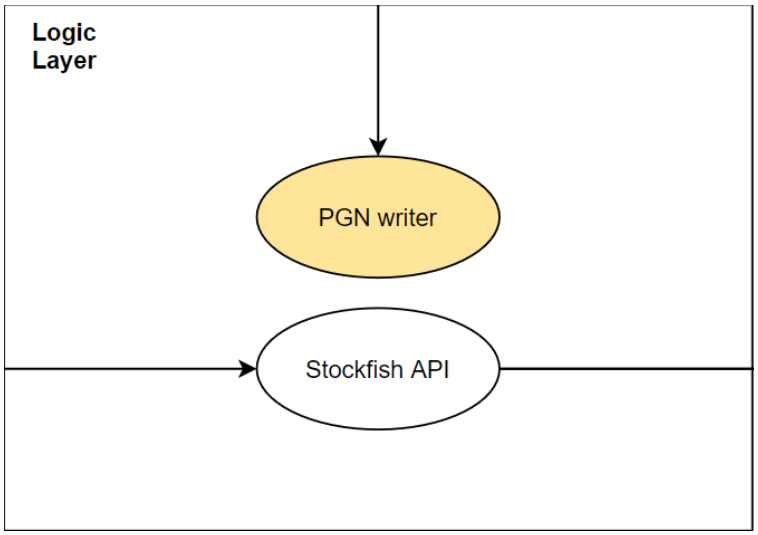

Figure 8: Example subsystem description diagram

5.4.1 Subsystem Hardware

Not applicable

5.4.2 Subsystem Operating System

Windows and Linux

5.4.3 Subsystem Software Dependencies

Requires PGN files and UNITY integration

5.4.4 Subsystem Programming Languages

Python and C#

5.4.5 Subsystem Data Structures

PGN files have the game data encoded in them.

5.4.6 Subsystem Data Processing

Only used as a representation to do the logic an another layer.

5.5 StockFish API subsystem

This subsystem will be responsible for determining the best place for the next move based on the difficulty setting. It will use the Stockfish API which is a free and open-source UCI chess engine. It could trigger the speaker in the "Output" layer when an illegal movement is detected. The decisions of the next movement made by the chess engine will be passed down to GUI. A movement could also trigger the speaker if a checkmate happens or the game is finished.

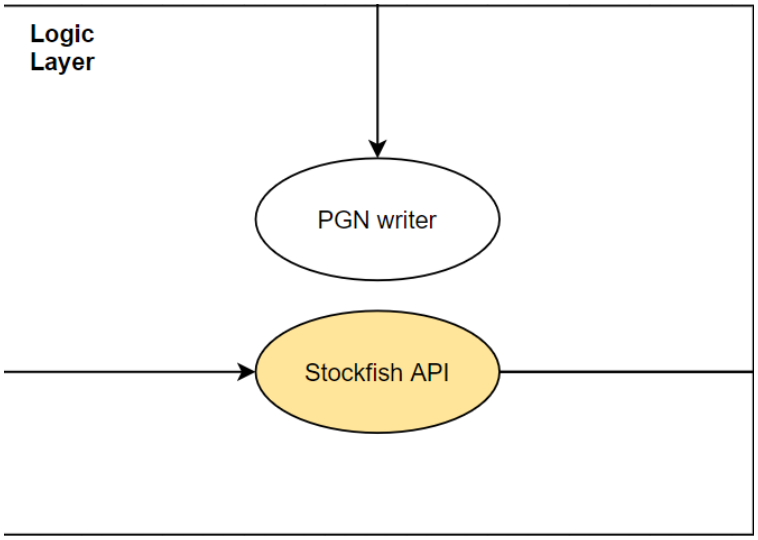

Figure 9: Example subsystem description diagram

5.5.1 Subsystem Hardware

Not applicable.

5.5.2 Subsystem Operating System

Windows or Linux

5.5.3 Subsystem Software Dependencies

Unity

5.5.4 Subsystem Programming Languages

Python and C#

5.5.5 Subsystem Data Structures

Data will be processed through the Stockfish API in order to determine the next move based off the difficulty. This data will be sent to the visual layer for movement.

5.5.6 Subsystem Data Processing

Stockfish uses it's own algorithm to figure out the next move to be made.

6 Output Layer Subsystems

The output layer subsystem handles the actual outputs to the GUI and audio from the speakers. The GUI will show the movements on unity API. The speaker will output audio after encountering an error from the machine learning layer subsystem or from the Stockfish API in the logic layer subsystem.

6.1 Layer Hardware

The speaker is a generic audio device. The GUI is on unity displayed to a screen.

6.2 Layer Operating System

Windows or Linux

6.3 Layer Software Dependencies

Unity

6.4 Speaker Subsystem

The responsibilities of the speaker component of the output layer subsystem is to output audio to the users. The speaker can output error messages if illegal or incorrect moves are made. The speaker can also output messages if the game has ended by either a checkmate or time based victory.

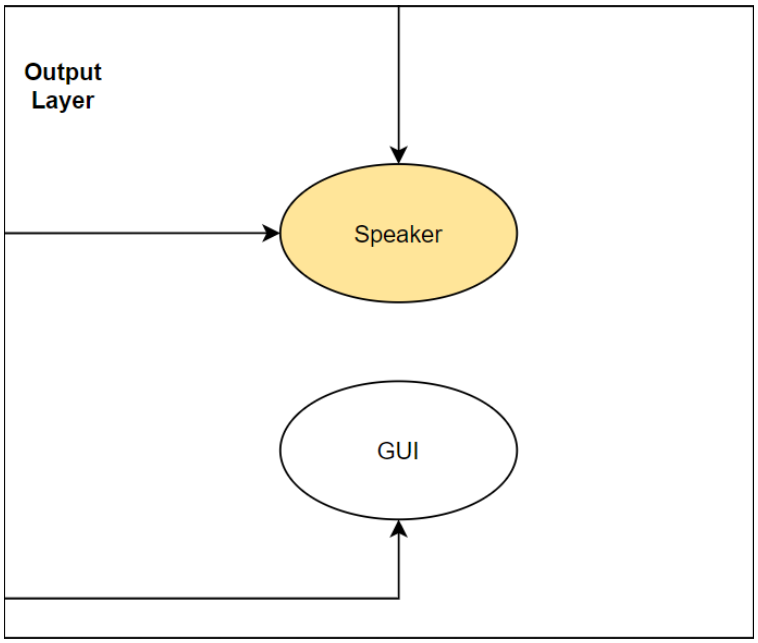

Figure 10: Example subsystem description diagram

6.4.1 Subsystem Hardware

A generic speaker.

6.4.2 Subsystem Operating System

Not applicable

6.4.3 Subsystem Software Dependencies

Unity integration

6.4.4 Subsystem Programming Languages

C#

6.4.5 Subsystem Data Structures

Not applicable

6.4.6 Subsystem Data Processing

Not applicable

6.5 GUI layer

The GUI layer is where the output is shown to the user. It also has a live feed of the chess pieces as they move.

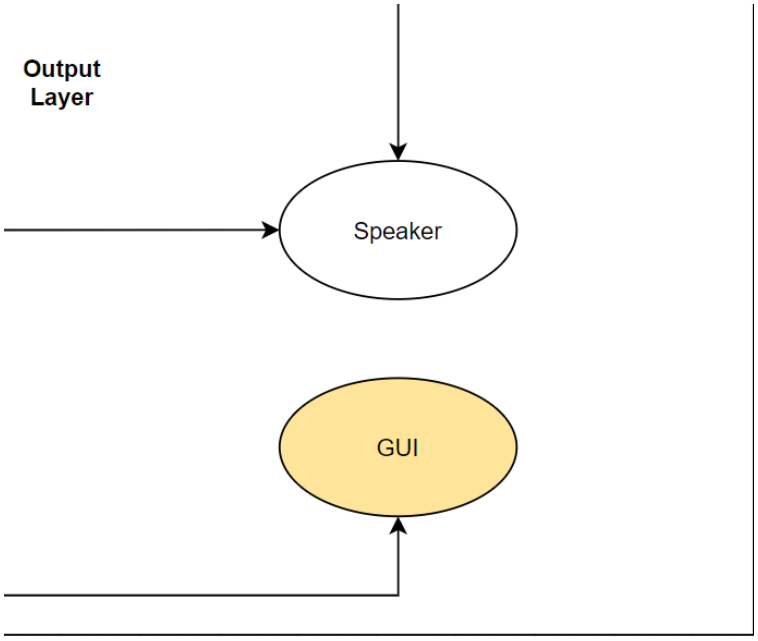

Figure 11: Example subsystem description diagram

6.5.1 Subsystem Hardware

Not applicable

6.5.2 Subsystem Operating System

Windows or Linux

6.5.3 Subsystem Software Dependencies

Unity

6.5.4 Subsystem Programming Languages

Python and C#

6.5.5 Subsystem Data Structures

The data is being sent from the stockfish api to the gui. It will move the pieces on the GUI.

6.5.6 Subsystem Data Processing

This GUI is based on the Stockfish algorithm and the user's moves.