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

SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2019

CHECK-MATES

CHECK-MATES
READY GO

JORDAN BURNES
RAHME BUTAINEH
SADAT HOSSAIN
MICHAEL SANTELLAN
DONGCHEN YE

Revision History

Revision	Date	Author(s)	Description
0.1	10.18.2019	JB	document creation
0.2	10.20.2019	JB, RB, SH, MS, DY	complete draft
0.3	10.21.2019	RB	edit document
1.0	10.21.2019	JB	official release
2.0	05.11.2020	JB	official release

CONTENTS

List of Figures

1 Product Concept

This section describes the purpose, use, and intended user audience for READY GO. READY GO is a chess playing GUI designed to use computer vision and a machine learning algorithm to play chess against a human player. READY GO is meant to be used by anyone, particularly students, interested in computer science, robotics, and machine learning.

1.1 Purpose and Use

READY GO will use computer vision, a subfield in the evergrowing field of machine learning, to play chess against a human player. The primary component of the product is a GUI that will be used to display the moves that the computer has made, and also reflect the moves of the player. It will make use of a camera to detect the pieces and a machine learning algorithm to classify the pieces accurately. Finally, it will utilize the Stockfish chess engine API in order to make the correct moves. READY GO is meant to be used as a demonstration tool to inspire the next generation of engineers. READY GO will be demoed at local high schools in order to encourage students to get involved in computer science, robotics, and machine learning.

1.2 Intended Audience

Our intended customers are high school students who have an interest in STEM. The goal is to demo the system playing chess against a human player and talk about the process of designing the software system. As our product is aimed toward students, the product would have very few applications commercially. High schools with computer science programs would be our target market. A very niche market would be chess clubs interested in having an AI player as part of their club. However, since READY GO is primarily for educational purposes, it is not meant for general use by the public. We want to use it as a tool to give insight into computer science, machine learning, and robotics to those interested in pursuing such career paths.

2 PRODUCT DESCRIPTION

This section provides the reader with an overview on READY GO, its components, its features, and its usage. In addition to describing each component of the design, the product's interaction with the user will be outlined here. How the user is to interact, maintain, and adjust READY GO will be described in detail.

2.1 FEATURES & FUNCTIONS

The quintessential function of READY GO is to play chess against a human player. The product consists of three primary components: the gui, the machine learning object detection model, and the Stockfish chess engine. In addition to these three primary subsystems, there are several auxiliary components such as a mounting board, video camera, camera stand, and a computer that are required in order for READY GO to perform.

The GUI will be made using pythons TKinter library as it will be easy to integrate with our other components that also utilize the python language.

YOLOv3 (You Only Look Once) is the object detection model that we will implement in order to detect the pieces and board squares. It was designed specifically for real-time object detection and performs well on a camera stream input.

The final major component of READY GO is the Stockfish chess engine API that will be utilized to determine the best move to play. The API requires .pgn (Portable Game Notation) files that capture single board states and evaluate the position at a configurable depth. READY GO will have three difficulty settings, and the Stockfish engine has settings for any ELO level.

The auxiliary features can be split into required and QoL sets. A HD camera is required in order to capture images of the board state. From previous experience, the YOLO model works extremely well (95% accuracy) on high quality images, and significantly worse on lower quality images. A camera mount is required in order to ensure consistency of the angle the images are taken from. As an additional feature, we will have output from the computer speaker say "check" or "checkmate" when the board state arrives to such a position.

2.2 EXTERNAL INPUTS & OUTPUTS

Given the simplicity of the READY GO objective and of the rules of chess, the only mission critical requirement is that the player plays legal and clear moves. The model will not be able to detect the board state, for instance, if the player places pieces on the edges of the board or at the center of four squares. Further, the Stockfish engine will not be able to evaluate the board if a king is missing from the board, or if a player is in check and it is not his/her turn.

The flow of data is outlined in the high level state diagram displayed in Figure 1 and will be described here. The user input is simple: the player makes a legal chess move. The live feed of the camera captures frames at 30 frames/second. If the machine learning model detects a change in board state (i.e. the human player has made a legal chess move), it will update the board state data object and create a .pgn file of the new board state. Portable Game Notation is the ubiquitous format used in the world of online chess that denotes the board state using a series of letters and numbers to identify the piece that most recently moved and its new position. For example *1.d4 d5* means that the last move made was the black player moving his pawn to d5. Once the board state is in PGN format, it can be interpreted by the Stockfish engine. Stockfish will then, depending on the level of difficulty set by the player at the beginning of the game, determine the best move at a certain depth. The model will then pass the best move to the GUI which will reflect the move chosen.

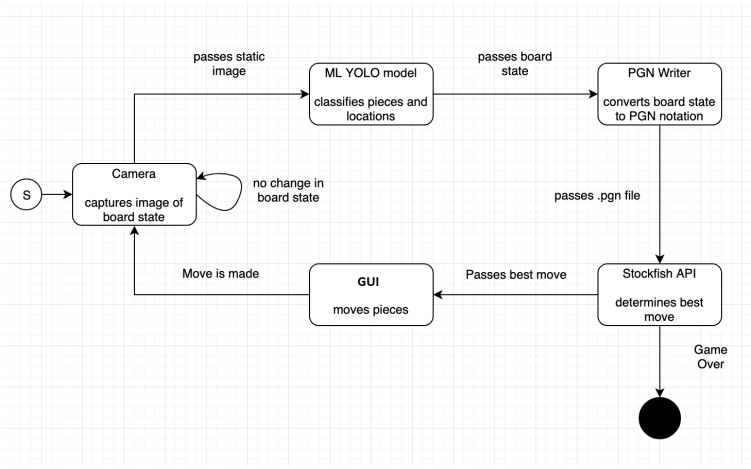

Figure 1: high level state diagram of data flow

2.3 PRODUCT INTERFACES

The product interface for READY GO is incredibly simple. The player will obviously and most frequently interact with the player board. In the first iteration of READY GO, the model will be trained on a single board. Future iterations will be trained to play on any board the player provides. The player will rarely interact with the GUI directly. At the beginning of the game, the player will set the difficulty level using the GUI.

3 Customer Requirements

Included below are the overarching requirements for the READY GO system whose main goal is to be able to play a game of chess with a human opponent. This section establishes, clearly and concisely, the "look and feel" of the product, what each potential end-user should expect the product do and/or not do. Each requirement specified in this section is associated with a specific customer need that will be satisfied. These requirements will be continually updated and changed as needed based of hardware and software requirements. The main functionality behind each requirement, however, will not be changed.

3.1 Assess The State of the Game

3.1.1 Description

The robot shall be able to asses the game through the use of a camera using machine vision and machine learning. The machine will get information about the location of all the current pieces, and from this information it will generate the best possible move (dependant on the difficulty setting) given the circumstances.

3.1.2 Source

CSE Senior Design project specifications

3.1.3 Constraints

Constraints would include chess pieces not being moved in accordance to the rules, or set on the edge of an area.

3.1.4 Priority

Critical

3.2 Perform a Move

3.2.1 Description

Once the state of a game is assessed and the best move (for that difficulty) is chosen the GUI shall reflect that move.

3.2.2 Source

CSE Senior Design project specifications

3.2.3 Constraints

None

3.2.4 Priority

Critical

3.3 Difficulty Settings

3.3.1 Description

The game shall have three different difficulty levels that our machine will assess when playing. This difficulty level will be set by using a switch with three different states. Each state will represent an easy, medium, and hard difficulty. The difficulties shall be represented in the moves that the GUI makes, each difficulty having its own percentage of winning. This difficulty shall be able to be switched between during the game, which will effect the systems current move.

3.3.2 Source

CSE Senior Design project specifications

3.3.3 Constraints

Constraints will include the amount of difficulty we can reduce for the easy level based on the chess algorithm, and dealing with the switch being changed during the assessment process.

3.3.4 Standards

Stockfish difficulty levels standard

3.3.5 Priority

High

3.4 Error Handling

3.4.1 Description

If the human opponent makes a move that is illegal or moves a piece that is not centered in a square, a speaker from the computer shall emit a warning message saying either "That is an illegal move, please move again" or "I am not sure of your move, please move again". It will then wait for its opponent to move again before it reassess the board. If everything is fine it will continue normal operation

3.4.2 Source

CSE Senior Design project specifications

3.4.3 Constraints

Constraints will include dealing with the human doing other things like taking pieces off the board for no reason.

3.4.4 Priority

Critical

3.5 Dealing with Win/Loss

3.5.1 Description

Once the state of a game is assessed and it is determined that the computer opponent is in checkmate, the speaker shall emit "You win. Good Game". If the state of the game is assessed and the next move the robot will deliver checkmate to its opponent, the speaker will instead emit after the move is made by the system and say "Checkmate. Good Game."

3.5.2 Source

CSE Senior Design project specifications

3.5.3 Constraints

Constraints will include the algorithms ability to determine win and loss.

3.5.4 Standards

World Chess Federation (FIDE) Standard

3.5.5 Priority

Critical

3.6 Determine the Current Turn

3.6.1 Description

The system shall be able to determine when it is its turn to make a move. A button shall be used to determine turn order. Once it is pressed, the system will then assess the board and make its move. If nothing has been moved from the previous state then the system will emit a message from the speaker.

3.6.2 Source

CSE Senior Design project specifications

3.6.3 Constraints

Constraints will include the ability to store past states to check against the current.

3.6.4 Priority

Critical

3.7 Put In Check

3.7.1 Description

Once the board is assessed and the system makes a move that will put its opponent in check its speaker will emit "Check" to let its opponent know that he or she is in check.

3.7.2 Source

CSE Senior Design project specifications

3.7.3 Constraints

Constraints will include determining check moves.

3.7.4 Priority

Moderate

3.8 Dealing with Friendly Promotion

3.8.1 Description

Once one of the system's pawns get to the end of the board, we shall have a system for promotion. Once the move is made, the system will have already (in the assessment phase) figured out which piece would be best to retrieve. It will then emit from the speakers "Please Promote my Pawn to PIECE", but PIECE will be replaced by the chosen piece. On the next turn of the system, it will assess the board and if the correct piece was not chosen then it will emit "You have not promoted my pawn to PIECE please do so and press the button again". Once it is determined that it has been correctly promoted the game will continue normally.

3.8.2 Source

CSE Senior Design project specifications

3.8.3 Constraints

Constraints will include what pieces are available to be promoted.

3.8.4 Standards

World Chess Federation (FIDE) Standard

3.8.5 Priority

Critical

3.9 Dealing with Enemy Promotion

3.9.1 Description

Once one of the enemy's pawns gets to the end of the board, the system shall recognize this. Once recognized the speaker will emit "Please promote your pawn and press the button once finished". Then, after the button is pressed the game will continue as intended. It is up to the human opponent to promote his own pieces.

3.9.2 Source

CSE Senior Design project specifications

3.9.3 Constraints

Constraints will include what pieces are available to be promoted.

3.9.4 Standards

World Chess Federation (FIDE) Standard

3.9.5 Priority

Critical

3.10 Dealing with Captures

3.10.1 Description

Whenever the game is assessed and it is appropriate for the system to make a capture it shall first, pick up the enemies piece and put it in a specified area that will be determined to be to the right of the game board. Then, the system will move its piece to the corresponding square.

3.10.2 Source

CSE Senior Design project specifications

3.10.3 Constraints

Constraints will include recognized the capture.

3.10.4 Standards

World Chess Federation (FIDE) Standard

3.10.5 Priority

Critical

3.11 Testing Mode

3.11.1 Description

To test the game we will have a state available that will be accessible by putting the difficulty to hard, and holding the button down for 5 seconds. Once the mode has been entered the speakers will emit "test mode". This mode will allow users to test the machines interaction with various board configurations without having to worry about making the correct moves. In this mode the system shall simply assess the current board situation once the button is pressed, and make the best move from what is given. No previous states will be saved. The system will be able to get out of test mode by setting the difficulty

to easy and holding the button down for another five seconds. The speaker will then emit "Exiting Test Mode".

3.11.2 Source

CSE Senior Design project specifications

3.11.3 Constraints

N/A

3.11.4 Priority

Moderate

4 Packaging Requirements

This is the requirements describing how the product will be packaged and delivered to the customers.The product needs to be easy and safe to move. The components that need to be moved include a chessboard with chess pieces, the camera with the tripod, and any laptop with the software installed. The product will be delivered to schools for presentation so they will be safely delivered is important.

4.1 The Chess Board and Chess Pieces shall be delivered in a box

4.1.1 Description

The chess board and chess pieces will be delivered in a box to maintain ease of delivery while keeping the product safe.

4.1.2 Source

Team Discussion

4.1.3 Constraints

The size of the box will effect how the product will be delivered. You want an appropriate size box to ensure the safety of the product while remaining relatively low weight.

4.1.4 Priority

Low

4.2 The camera and tripod shall be delivered in a box

4.2.1 Description

The camera and tripod will be delivered in a box with some padding to maintain ease of delivery while keeping the product safe.

4.2.2 Source

Team Discussion

4.2.3 Constraints

The size of the box will affect how the product will be delivered. You want an appropriate size box to ensure the safety but with this more expensive equipment you are more focused on safety than weight so there will be some padding.

4.2.4 Priority

Medium

5 Performance Requirements

5.1 The system shall have a maximum boot up time of 10 seconds

5.1.1 Description

The system will take at most 10 seconds to boot up. The boot up time will depend on the operating computer and the computer vision software.

5.1.2 Source

Team Discussion

5.1.3 Constraints

The computer used and computer vision software must be installed prior.

5.1.4 Priority

Low

5.2 The system shall have a maximum shutdown time of 10 seconds

5.2.1 Description

The system will be capable of shutting down in less than N seconds. This includes the shutting down of the operating system as well as the computer vision software.

5.2.2 Source

Team Discussion

5.2.3 Constraints

The computer used and computer vision software must be installed prior.

5.2.4 Priority

Low

5.3 The system shall have a maximum restart time of 15 seconds

5.3.1 Description

The system will restart in less than N seconds. The system will need to shutdown and reboot the computer used and computer vision software.

5.3.2 Source

Team Discussion

5.3.3 Constraints

The computer used and computer vision software must be installed prior.

5.3.4 Priority

Low

5.4 The system shall recognize the game state of the board using computer vision in at least 2 seconds

5.4.1 Description

The system will be able to recognize the game state of the board using computer vision in at least 2 seconds. It is important that the computer vision quickly recognizes the game state of the board because

it will be doing this constantly so it will know what move to make. We will make work faster by training the system using images of the chess board.

5.4.2 Source

Team Discussion

5.4.3 Constraints

Computer vision software will need to be installed.

5.4.4 Priority

High

5.5 The system shall perform a chess move in at least 30 seconds

5.5.1 Description

The system will perform a chess move in at least 30 seconds. The time required to the make the move will depend on the difficulty set for the algorithm as well as the time it takes for computer vision to evaluate the board state.

5.5.2 Source

Team Discussion

5.5.3 Constraints

The difficulty level will effect how long it will take to make the correct move. Computer vision will also need to be installed to read the board state.

5.5.4 Priority

High

6 Safety Requirements

Below are the safety requirement we must adhere to while working on READY GO. Requirements will consists of Lockout/Tagout for equipment items and wiring compliance. It will also include the National Electric Code wiring compliance and the RIA robotic manipulator safety standards.

6.1 Laboratory equipment lockout/tagout (LOTO) procedures

6.1.1 Description

Any fabrication equipment provided used in the development of the project shall be used in accordance with OSHA standard LOTO procedures. Locks and tags are installed on all equipment items that present use hazards, and ONLY the course instructor or designated teaching assistants may remove a lock. All locks will be immediately replaced once the equipment is no longer in use.

6.1.2 Source

CSE Senior Design laboratory policy

6.1.3 Constraints

Equipment usage, due to lock removal policies, will be limited to availability of the course instructor and designed teaching assistants.

6.1.4 Standards

Occupational Safety and Health Standards 1910.147 - The control of hazardous energy (lockout/tagout).

6.1.5 Priority

Critical

6.2 National Electric Code (NEC) wiring compliance

6.2.1 Description

Any electrical wiring must be completed in compliance with all requirements specified in the National Electric Code. This includes wire runs, insulation, grounding, enclosures, over-current protection, and all other specifications.

6.2.2 Source

CSE Senior Design laboratory policy

6.2.3 Constraints

High voltage power sources, as defined in NFPA 70, will be avoided as much as possible in order to minimize potential hazards.

6.2.4 Standards

NFPA 70

6.2.5 Priority

Critical

7 Maintenance & Support Requirements

After the delivery of the product, the CheckMates team will provide maintenance and support for the product. This includes any updates in firmware and software that comes with the READY GO system. It will also include the privacy for the source code using Git.

7.1 Firware and Software Updates

7.1.1 Description

Updates will be patched through the use of the internet, where new updates will be available for download via the user. Updates will include new features/bug fixes.

7.1.2 Source

CheckMates

7.1.3 Constraints

Users will be able to download if they have internet connection

7.1.4 Priority

Moderate

7.2 Source Control using Git

7.2.1 Description

Source Code will remain open as the program utilizes a lot of open source code in creation. GitHub will be open to anyone.

7.2.2 Source

CheckMates

7.2.3 Constraints

Users will be able to download if they have internet connection.

7.2.4 Priority

Moderate

8 Other Requirements

In this section, other requirements determined by our team in the design process are listed. This are product features identified by the team in discussions about the design including the devil mode which allows the system preform the flipping-board operation and the programming language that will be used in developing the system.

8.1 Devil Mode

8.1.1 Description

Once the mode has been entered, the system will be able to detect whenever its evaluation function predicting more than 80 percent of the loss, and then perform the flipping-board operation. This mode will be accessible by putting the difficulty to hard, and holding the button down for 10 seconds.

8.1.2 Source

Team discussion

8.1.3 Constraints

The robot shall perform this operation gently with respect

8.1.4 Priority

Future

8.2 Programming Language

8.2.1 Description

Most of the open-source libraries we plan on using are in python3. Source code shall be written in python3 to march the library.

8.2.2 Source

Team discussion

8.2.3 Constraints

Must be adapted to the robot arm

8.2.4 Priority

Critical

9 Future Items

In this section, features and requirements that deemed out of scope/budget for the project are listed. These are product features identified by the team in discussions about the design including the devil mode which allows the system to preform the flipping-board operation.

9.1 Devil Mode

9.1.1 Description

Once the mode has been entered, the system will be able to detect whenever its evaluation function predicting more than 80 percent of the loss, and then perform the flipping-board operation. This mode will be accessible by putting the difficulty to hard, and holding the button down for 10 seconds.

9.1.2 Source

Team discussion

9.1.3 Constraints

The robot shall perform this operation gently with respect

9.1.4 Standards

The robot shall be able to make the decision and perform the operation within 10 second

9.1.5 Priority

Future