Software Formalization

Year: 2023 Semester: Spring Team: 3 Project: ”Rigged” Card Shuffler

Creation Date:Feb 17, 2023 Last Modified: Feb 17, 2023

Author: Zachary Williams Email: will2051@purdue.edu

Assignment Evaluation:

| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| --- | --- | --- | --- | --- |
| **Assignment-Specific Items** | | | | |
| **Third Party Software** |  | x2 |  |  |
| **Description of Components** |  | X3 |  |  |
| **Testing Plan** |  | x3 |  |  |
| **Software Component Diagram** |  | x4 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

*Relevant overall comments about the paper will be included here*

1.0 Utilization of Third Party Software

| Name | License | Description | Use |
| --- | --- | --- | --- |
| Scikit-image | BSD License | Collection of algorithms meant for image processing. [1] | This is being used to have access to an efficient implementation of the sobel operator. |
| numpy | Numpy | Library of well optimized tensor data representation and linear algebra functionality. [2] | This is being used to store image data as ndarrays for use in the computer vision algorithm. |
| picamera | BSD License | Library for interfacing with the picamera and storing the data in python. [3] | This is being used to capture images from the picamera to be used in the computer vision algorithm. |

2.0 Description of Software Components

The software components for the project are as follows: Basic I/O, Raspberry Pi Shuffling Algo, Computer Vision, and Embedded Shuffling Algo.

The Basic I/O component is responsible for handling I/O inputs from the push-buttons, displaying menus and shuffling status on the LCD screen, and communicating with the Raspberry Pi. External interrupts on a set of GPIO pins will detect the push-button inputs. SPI will be used to communicate with the LCD screen. The software for writing to and initializing the LCD screen was taken from a past project in ECE 362. UART will be used to communicate between the Raspberry pi and microcontroller. This component will be developed entirely by our team barring only the software libraries given through the ECE 362 labs.

The Raspberry Pi Shuffling Algorithm component is responsible for receiving the input from the microcontroller and determining the shuffling order that will be present in the final deck based on the user inputs. This component will take a set of inputs from the microcontroller {game, number of players, winner, Card Orders [rank,suit,position]} all of these will have the ability to have default values as often times some of these will not be necessary to determine the output deck order. This component will be developed entirely by our team except for the use of the random and serial python libraries for randomness and UART communication respectively in this component.

The Computer Vision component is responsible for taking pictures of the cards and determining the rank and suit of the card using computer vision algorithms. This component takes images of the cards using the picamera library and stores the image data in a numpy array. The computer vision uses edge detection and bounding boxes to compare the new images to known images that we have taken in the card holder. This component will be developed largely by our team using libraries for an implementation of the sobel operator and numpy for image data.

The Embedded Shuffling Algorithm component is responsible for controlling the motors using the microcontroller based on inputs from the raspberry pi. The Basic I/O component will pass UART messages from the raspberry pi to this component. Based on those messages, the microcontroller will determine the number of steps that need to be taken based on a lookup table and will tell the stepper motor n-steps in the direction necessary. This component will also control the DC motors and stepper motor that will kick the cards out of the card holder during the shuffling process. The DC motors will be controlled using GPIO outputs and the stepper motors will be controlled using timers in OPM. This component will be developed entirely by our team, and we do not plan on using any third party software for it.

3.0 Testing Plan

Importance: 1

The Basic I/O component will be tested piecewise using testing functions accessible through the UI. The LCD Board testing function will display several different types of screens ranging from basic menus to screens that display the shuffling progress. The button input testing will allow the user to manipulate the location of items based on the buttons pressed. The UART testing will send basic messages to the Raspberry pi and wait for responses and display this on the LCD screen. This is the most important component to test since accessing other crucial tests are dependent on the LCD screen and button inputs functioning correctly. The menu system is also critical for the device to function properly.

Importance: 3

The Raspberry Pi Shuffling Algorithm component will be tested using card position fixture files to imitate the UART messages from the microcontroller. The testing will take these fixture files and construct an output deck order that will be tested for repeat cards or positions, if specified card locations were met, and will be printed out to allow the user to determine if win conditions were met. This testing will be able to be done independent of the microcontroller. This component is the same importance as the computer vision component to test as while these are crucial for accurate rigging of the deck, the product could still function as a completely random shuffler without these components.

Importance: 3

The Computer Vision component will be tested using a train-test split on card images. Currently, the software has been tested using highly controlled images and thus has not accurately been tested. The computer vision will be tested further by performing the train-test split on card images directly from our product. This will allow for more accurate training because of the more representative training data and will hopefully allow for a more accurate overall algorithm. This component is the same importance as the raspberry pi shuffling because although this is crucial for a rigged shuffle, this would not be necessary for a purely random shuffle.

Importance: 2

The Embedded Shuffling Algorithm will be tested piecewise using testing functions accessible through the UI. The stepper motors will both have testing functions that spin the motors some number of steps in both directions. The DC motors will have testing functions that will just spin and then stop the motors. The stepper motor for moving the card wheel will have a testing function that will move to several different spots on the card wheel and return back to the original position. The stepper motor and DC motors that are in charge of dispensing cards will have a testing function that will dispense several cards in a row each one at a time. This component is very important to test since the card shuffler will not be able to shuffle the deck accurately without this component.

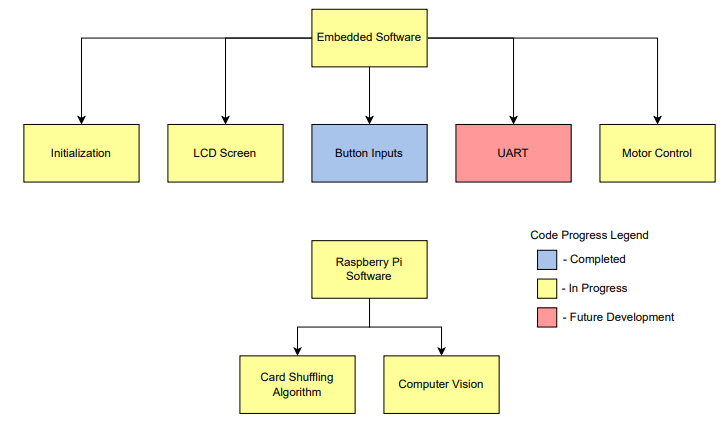
4.0 Sources Cited:

[1] “News¶,” *scikit*. [Online]. Available: https://scikit-image.org/. [Accessed: 17-Feb-2023].

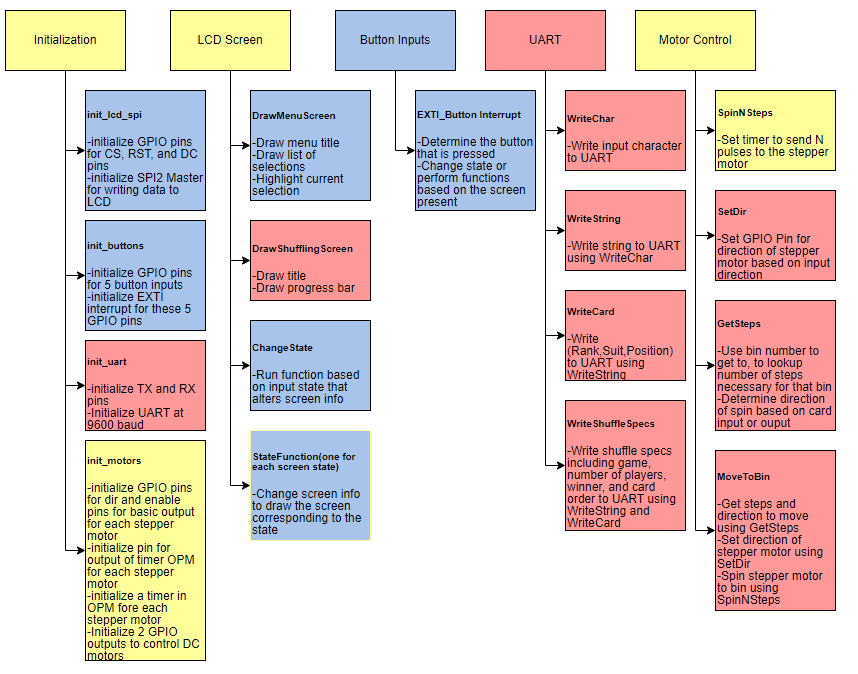
[2] *NumPy*. [Online]. Available: https://numpy.org/. [Accessed: 17-Feb-2023].

[3] “Picamera,” *PyPI*. [Online]. Available: https://pypi.org/project/picamera/. [Accessed: 17-Feb-2023].

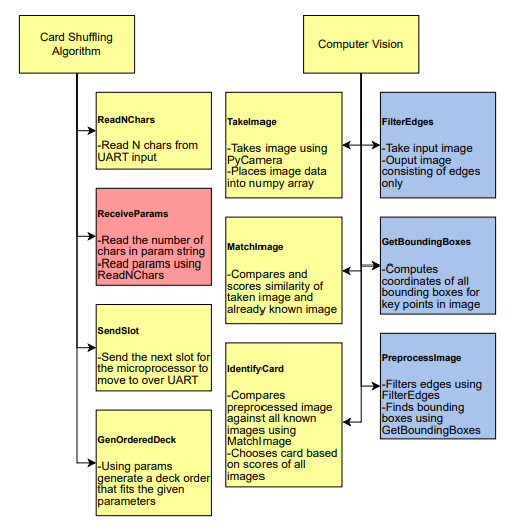
Appendix 1: Software Component Diagram



*Fig 1: Overall Code Structure*

**

*Figure 2: Embedded Code Diagram*

**

*Figure 3: Raspberry Pi Code Diagram*