Project 2 Report
Team 21

Andrew Berkeland, Andrew (AJ) Eubanks, Steven Johnson, Eric Mitchell, Alvi Topuzi

10/11/2024

Executive Summary

This project had many subgoals. For Project 2 specifically, we had to use image processing techniques to detect changes in a background image. We implemented a webcam as an optical sensor. Using background subtraction, image erosion, and various detection techniques, our program identifies the colors and shapes of objects on a sheet of paper. We also used mathematical operations to calculate the angle between the positive horizontal axis and each region's centroid. The positive horizontal axis starts at the image center and projects in the direction of increasing pixel column number. In addition, we addressed some of the parts of the design process for our final project. We developed engineering requirements from customer requirements. Also, we brainstormed many final project ideas and concepts. Using a weighted decision matrix, we decided on the best concept for our final design. We created a non-functional mock-up of our final design. Lastly, our team developed a website to document our project's progress. This project taught us about different computer image processing techniques and website design fundamentals. We also learned how to convert customer needs into descriptive engineering requirements. Our most important lesson is that we need a better light source for our camera setup. We faced many issues with our image washing out, and we believe constant lighting will lessen the effect of other lights in the room.

Camera Subsystem Design - Engineering Requirements

Customer Requirement	Engineering Requirement
Camera-based	Uses a Logitech Brio 100 camera as the primary shape and color detection sensor
Uses Matlab	We will use Matlab for all image detection and processing purposes related to software. Matlab will display the output of the processing and detection algorithms.
Detects blue, red, dark green, and yellow stickers	The color detection algorithm will use the HSV color scheme to differentiate the colors at the centroids of each identified region
Detects the shape of the colored objects	The shape detection algorithm will use area to find regions of interest. From there, it will have circularity thresholds to differentiate circles, squares, and triangles.
Identify the placement of up to nine colored stickers on the board	The algorithm will identify and plot up to nine centroids of the detected objects on the board.
Capable of performing background subtraction (no stickers initially on the sheet)	The algorithm will incorporate the background subtraction method to compute the image foreground. It will also use other methods to identify and modify regions of interest.

Hardware Design

Figure 1: Hardware Demo Design

Our temporary setup, constructed from cardboard, fishing line, and tape, is a camera mount. Figure 1 shows our setup for Project 2. The camera is at a height of 17 inches and plugged into a laptop behind the cardboard wall. At this height, the camera can capture images of the entire game board. The design required a wall to alleviate lighting issues. It also requires a stable surface for the camera. In our final project, we plan to use LED light strips to address the lighting issues we faced. Also, we plan to 3D-print the majority of our design which will make the camera mount more stable.

The camera is an appropriate sensor for this design because it captures images with enough quality to determine if ¾" shapes are present on the backdrop. Since our final project's game pieces will be larger than this, the camera will also be appropriate for it as well. Our test setup evaluates the camera's resolution, clarity, and integration with MATLAB detection software. With a camera resolution of 640x480, our setup captures one square inch in about 3248 pixels. The calculations used to find the number of pixels per square inch are shown below.

$$\text{Width: } \frac{640\, pixels}{11\, inches} \approx 58\, pixels/inch$$
$$\text{Height: } \frac{480\, pixels}{8.5\, inches} \approx 56\, pixels/inch$$

$$(1\, inch\ *\ 56\, pixels/inch)\ * (1\, inch\ *\ 58\, pixels/inch)$$
$$=\ 1\, inch^2 *\ 3248\, pixels/inch^2 \approx 3248\, pixels$$

Software Design

The flowchart in Figure 2 illustrates the image processing workflow implemented in MATLAB to detect shapes and colors on our gameboard. The process begins by capturing a background image, followed by capturing a new image with stickers placed on the board. MATLAB then performs background subtraction to isolate the stickers from the background, computes a binary image, and applies an edge detection process (e.g. erosion) to refine the regions of interest. These regions are analyzed based on their size, and those with a region area within the desired range (less than 100,000 and greater than 300 pixels) are selected for further processing.

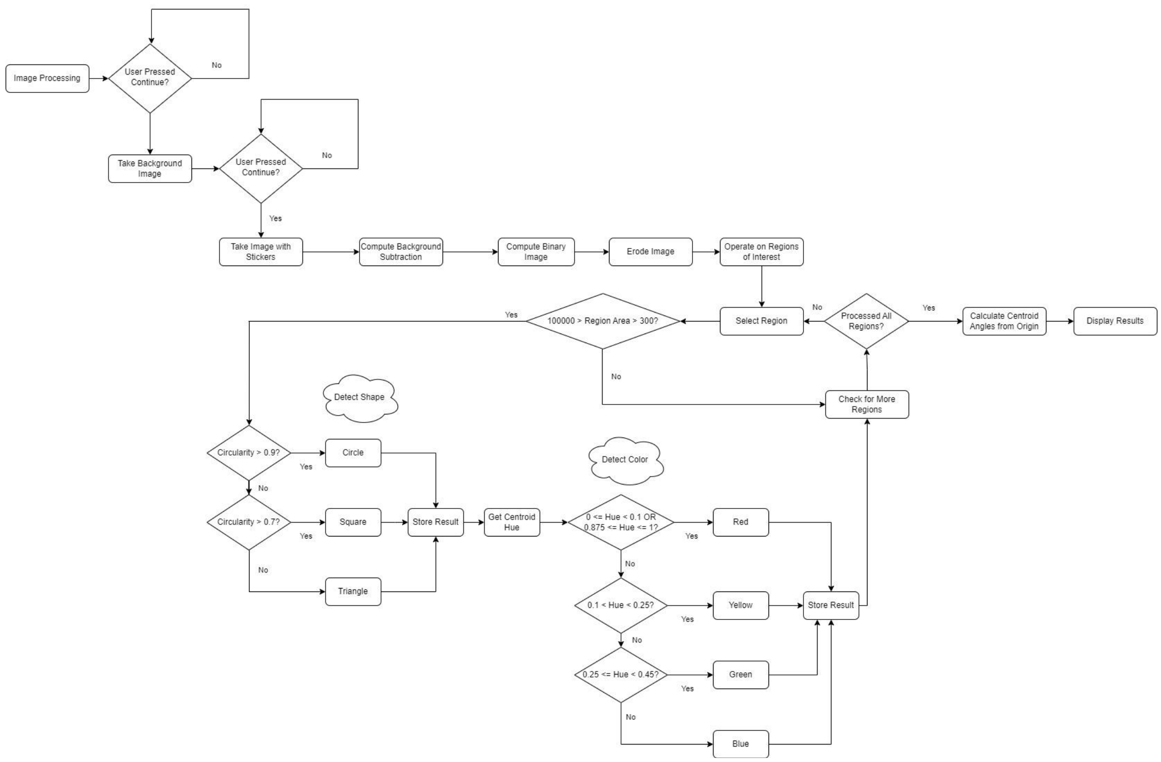

Figure 2: Software Flowcharts

Once regionprops() identifies regions of interest, our Matlab program uses shape detection algorithms to categorize the stickers. When called on a binary image, regionprops() calculates the circularity of each region. The process we chose to determine a region's shape is as follows: if circularity is 0.9 or above, the program classifies the region's shape as a circle. If the circularity is 0.7 or higher, the algorithm classifies the region as a square; otherwise, the region is categorized as a triangle. Additionally, the program calculates the hue value from the central pixel of each region. Once the hue is determined, another sequence of statements classifies the region's color. The hue range determines whether the sticker is red, yellow, dark green, or blue. These results, including shape and color, are then stored and displayed for the user. In addition, the region's angle from the center of the image is displayed. The program loops through all regions until the entire image is processed, after which Matlab plots the results.

Engineering Requirements for the Entire System

Customer Requirement	Engineering Requirement	Test to be conducted
Autonomous	The system must be able to play a turn without user instruction. The last command the machine should receive before making its move is when the user takes their turn.	If the robot can not move after the user presses a button for the desired location, this test is a failure
Inexpensive	The system should cost less than $200 for parts and assembly.	Write a parts inventory and total the costs. If the total cost is less than $200, the test passes.
Quiet	The system should generate less than 80 dB (max normal conversation volume) of noise while running.	Measure the loudness with a decibel meter. If the loudness is greater than 80dB, the test fails.
Robust	The frame should be resilient when impacting a table from a 3” drop or impacting a hand with a force of 500 N (lowest adult punching force)	Drop the enclosure and hit the enclosure's sides with a light punch. If the frame breaks or the machine fails to work, this test fails.
Fast	The system should be able to take its turn within 10-15 seconds	Set a stopwatch when the user’s turn ends. This test fails if the robot’s turn takes more than 15 seconds.
Efficient	The robot should be able to take a turn with ten or fewer total movements from any of the moving system components.	Count the number of component moves for the system. If the robot makes more than ten physical movements, the test fails.
Safe	The moving system components should be in an enclosure that is not open to the outside.	If all moving parts are enclosed, the test passes.
User-Friendly	The average user who knows the rules of Connect 4 can understand how to use the GUI.	Sample 10 people from campus randomly. The test passes if 80% of them can play the game based on the on-screen instructions.
At least one closed-loop controlled motor with smooth motion	A closed-loop motor with PID control must be a component. The PID-controlled motor allows for the motor to travel	Use a motor. If there is a visible jitter in the mechanism, the test fails.

	from one point to the next in a continuous motion without making visually noticeable andabrupt stops.	
Detect coins off the board and move them to the desired location on the board determined by the user and a GUI	Use a webcam to detect where available coins are on the side of the board The piece moves to the location selected by the user on a screen. The screen displays all available moves.	The test passes if the robot can grab and place pieces without interfering with placed pieces.
Clears the gameboard upon completion and places pieces back in their appropriate locations	The robot can move all pieces back to a color-coordinated rack on the side of the game board after a winner is detected and the user initiates a board reset.	The test passes if the system can reset the board to its original state with all pieces in the correct locations.
Displays that the game is complete and declares the winner (if any)	The GUI will display “Game Over” and the winner	The GUI detects when the game is over at all appropriate times and displays the winner.
16” x 16” board, with each space being 2” x 2”	The board spaces will be square with 2”x2” dimensions per square and a 16” by 16” board.	The board is made as described.
Game pieces of 1.5” in diameter	The game pieces will be of two colors and circular with 1.5” diameters	We will construct circular game pieces with 1.5” diameters.

Design Solutions

We are treating the z-axis as vertical movement for all three design solutions. Our board is horizontal, and we will treat the x-axis as left-to-right movement and the y-axis as forward and backward movement, with respect to the camera lens. The camera will be mounted above the board, looking down on it. Our three design solutions are shown in Figures 3, 4 and 5.

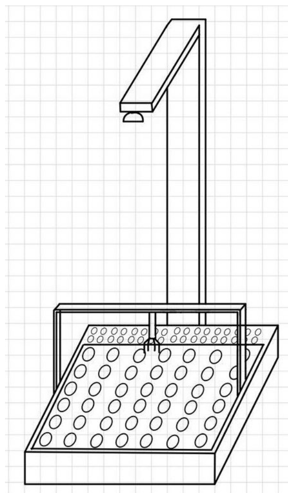

Figure 3: Design Solution #1

Design #1: Manipulator claw with x- and y-axis traversal. Under the board, there will be two sets of rails for movement across the board. These will move the claw along the x- and y-axes. The claw will open, grab a piece, move to the selected spot, and drop the piece in place.

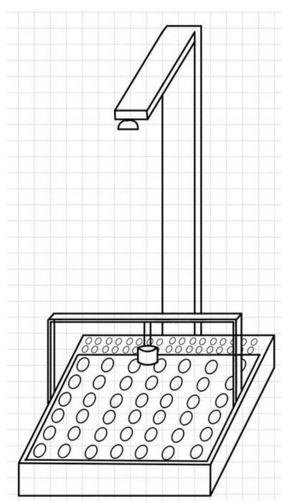

Figure 4: Design Solution #2

Design #2: This design is the same as Design #1, except the claw is replaced with an electromagnet. We will attach a ferromagnetic object to each of the game pieces. The electromagnet will be used to move game pieces around the board.

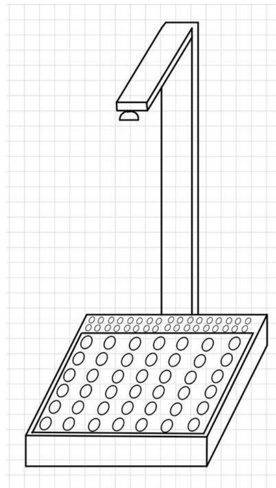

Figure 5: Design Solution #3

Design #3: This design is similar to Design #2. In this design, the electromagnet will stay under the board. The electromagnet will drag the game pieces across the board into the places selected by the user and the computer.

Concept Evaluation

		Design Choice 1	Design Choice 2	Design Choice 3
Autonomous	5	5	5	5
Inexpensive	3	3	5	3
Quiet	3	3	5	5
Robust	3	5	5	5
Fast	5	5	5	5
Safe	5	5	5	5
User-Friendly	3	3	5	5
Easy to Create	1	3	5	3
Lightweight	1	3	5	5
Aesthetic	1	3	3	5
Total Score		**126**	**148**	**142**

Our chosen weighting factors, 1, 3, and 5, represent optional, important, and critical. Our score for each criterion was selected to be 1, 3, and 5 as well. These values represent states of not meeting the criterion, partially meeting it, and completely satisfying it. From this decision matrix, the final design that we will test first is Design 2. This includes an electromagnet that will move the pieces where they need to go from above. While this design is less aesthetically pleasing than having the magnet on the bottom, it will be cheaper since the magnet does not need to be as strong. In addition, the design will be easier to create since we will not be working underneath the board.

We chose the weighting factors for each criterion based on their importance. We ranked autonomous, fast, and safe as being absolutely critical for the design. It must be autonomous. Otherwise, there is no point in creating this machine in the first place. It must be fast because if it is not, the user would rather play with the standard game. Finally, it must be safe so that there is no potential for harm. Next, we ranked inexpensive, quiet, robust, and user-friendly as important things to consider. It is essential for the design to be inexpensive because the cost relies on us to make the system with no profit. It should also be quiet and robust, as these are part of the customer's desires. Finally, it must be user-friendly because we do not want the game design to be complicated and frustrate the user. The user should be able to play the game and know how it works without any questions. Finally, we ranked easy to create, lightweight, and aesthetic as optional criteria. This is because they are not a part of the customer requirements but are things that we desire our design to have. Design Choice #2 ranked the highest overall since it fully satisfies most of these requirements. The only criterion at which it does not excel is aesthetics. This is because it will have to have many components located above the gameboard. Ideally, everything would be hidden, but this would cause other issues that are more important than the look of the game. We believe that even with this design choice, we can manage to make it look as aesthetically pleasing as possible once we start designing to mitigate any losses we plan to encounter based on this design choice.

References

Apoorva Kapadia. Image Processing Lab Demo Code. (2024, Fall). ECE 4950. Clemson, South Carolina: Clemson University.

ECE 4950 Project 2 – Sensor and Website Rubric

Use the guidelines below to complete your report and add at the end of your report.

Group Number (and Name): Team 21

<table>
<tr><th>Score</th><th>Pts</th><th></th><th></th></tr>
<tr><td></td><td>5</td><td>General Format - Professional Looking Document/Preparation (whole document)
a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides).
b) Spelling and grammar are correct
c) Layout of pictures – all figures need numbers and captions and must be referenced in the text
d) Follows the page limitations below.
e) References. Use IEEE reference format.
f) This grading sheet is included as the final page.</td><td>O3-SA1:1</td></tr>
<tr><td></td><td>5</td><td><u>Page 1: Title, Group Name, Group Members, and Date</u>
<u>Executive Summary</u> (1 concise, well-written paragraph)
Provide an overview of this project. Briefly describe what you did and what you learned.</td><td>O3-SA1:1</td></tr>
<tr><td></td><td>20

20

20</td><td><u>Subsystem Design</u>
<u>Page 2: Engineering Requirements for the Camera subsystem</u> (~1 page)
In the context of just the Camera-as-a-Sensor, make a two column table that contains a column for the Customer Requirements (what are the functions of the sensing system?) and the resulting Engineering Requirements. Each row should contain a specific customer requirement and the resulting engineering requirement. One customer requirement may generate multiple engineering requirements. For example, the customer will want an “accurate” system, the Engineering Requirement could be 99.5% detection success.

<u>Pages 2-3: Document Hardware</u> (1 page)
Describe and show images of the equipment used, connection diagrams, calculation of resolution – pixels per square inch/cm on game board etc. Is the camera an appropriate sensor?

<u>Pages 3-4: Document Software</u> (2 pages)
Using Flowcharts, state diagrams, data structures etc. describe how the software is implemented. There is no need to include the source code.</td><td>O1-SA5:2

O2-SA2:1

O2-SA2:1</td></tr>
<tr><td></td><td>30</td><td><u>Overall System Design</u>
<u>Page 5: Engineering Requirements for the Entire System</u> (~1 page)
Make a three-column table that lists the Customer Requirement in the first column and the resulting Engineering Requirement(s) in the second column. Note that a customer requirement might branch to multiple engineering requirements. The third column should list the test that will be done on the prototype to verify that the design chosen meets each requirement.

<u>Pages 6-7: Design Solutions</u> (~2 pages)
Use a brain storming session to generate concepts. Document your top three most feasible ideas with sketches and brief descriptions while explaining the main features of each concept.

<u>Page 8: Concept Evaluation</u> (~1 page)
Use a complex (weighted) Concept Evaluation Matrix to show how the final design was chosen from the three best ideas described previously. Include a description of the weighting factors. Make sure to use at least six of your most important criteria in the matrix and be sure to provide and include the weighting factor (i.e. importance) for each criterion.</td><td>
O1-SA5:2

O2-SA1:4

O2-SA1:4</td></tr>
</table>

		Describe your final design choice (~1 paragraph) The robot mock up will be graded based on the following criterial: a) Does it demonstrate the proposed concept in sufficient detail? b) Is it well thought out? c) Does it look like the final construction using this concept will be robust?	
		Page 9: Grading Sheet	
	20	**General Website Format** – ***provide link here:*** https://ajeubanks.github.io/ ece-4950-team21-website/ Is the website: a) Aesthetically clean b) Complete. Does it include: 1. The Team Description? 2. Reports 1 and 2? 3. Outline of future sections? 4. Proper use of graphics? c) Follow accessibility compliance at: https://siteimprove.com/en-us/accessibility/ada-compliance-website/	**O3-SA1:1**