ECE 4950 Project 2

Team Megafist
Group 13
Valerie Moran, Brandon Cuttino, Ben Liggett, Sarah Anderson, and Rocky Aho
October 19, 2020

Executive Summary:

The overall aim of this project was to gain an understanding of many of the initial, progressive, and successive, but important processes of an engineering project such as generating creative solutions to an engineering problem, qualifying customer requirements to engineering requirements, quantifying those qualifications, developing mock-ups of and proof of concepts, and finally the finances and even the documentation behind that project. To begin this project, we first set about brainstorming the requirements for the design based on the criteria established by our target customers. We then translated those customer requirements to weighted, quantified engineering requirements that we would then use to rank three different brainstormed and illustrated designs. The design with the highest score would be chosen as the design for further development. Once that final design was determined, we set about designing a physical mock-up of it as well as looking at the likely expenses of developing the product. We then looked to see if the idea of detecting colors with the theorized materials was even possible, and by creating a program, we were successfully able to do so. We also created a website during all this successive development to be able to document everything we managed to achieve during this project. In the end, we gained a better understanding of what it takes to qualify what a customer wants, to quantify the customer's requirements for developing the best solution, and many of the processes that go into developing the product further, including other processes such as the financials and documentation of a project.

Engineering Requirements:

Table 1. Engineering Requirements

Customer Requirements	Engineering Requirements	Verification Test for each Requirement
Inexpensive	Cost of production ($)	Under $50.
Quiet	Minimal decibel noise output.	Measured sound below 45dB.
Robust	Components connected and secured to one another. Sturdy frame and structure.	Frame won't break under 3lb of stress repeatedly.
	Device built for variations in marker placements.	Device produces correct results independent of marker locations.
Fast	Max time for the system to complete its task less than 1 minute	Device completes the task in under 1 minute regardless of marker placements.
Efficient	The device doesn't overestimate or underestimate the turning radius. Smooth motion.	Percent error measured below 1% for turning radius. Motors turn at a constant rate with consistently accurate results.
	The vision system doesn't use repetitive or unnecessary code.	Clean and compact code. Solves task in under 20 seconds
Safe	No risk of electrical shock from components or loose wires.	Wires are hidden and securely placed. No bare metal from electrical components. Sturdy
User friendly	Complete user interface made to be simple and sufficient. All tedious sections automated.	Ready to use in under 5 minutes. Instructions supplied. Completely autonomous except for setting marker locations.
Closed loop system to move washers to predetermined location	To produce a subsystem that doesn't require manual inputs	No manual input when the device runs, Plug and Play functionality.
Defined rest position and notification system	Output a rest position of the gameboard and a notification system	LED lights up to signify task completion. Returns to rest position set in the closed loop system.

Design Solutions:

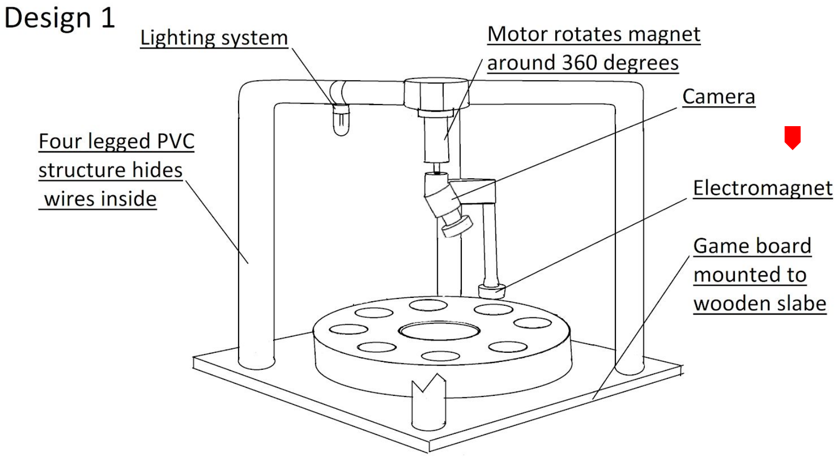
Design 1
Lighting system
Motor rotates magnet around 360 degrees
Camera
Four legged PVC structure hides wires inside
Electromagnet
Game board mounted to wooden slabe

Figure 1. Design Solution 1 Diagram

Design 1, as shown in **Figure 1**, has a stationary wooden base with the game board on top of it. A four-legged support system is mounted on top of a wooden slab. At each corner of the slab, PVC pipe is attached to build the four-legged support system. Each of the vertical posts connects to an elbow connector and branches inwards to connect to a middle point. Attached to one of the overhanging beams is a lighting system, which will be focused on the gameboard. This will ensure that the camera has constant lighting during execution. The intersection of the four beams holds the camera, which is focused on the game board, and a stepper motor. The stepper motor is connected to an elbow connector that hovers the electromagnet above the washers. This motor will rotate around the board and stop when above the correct washer. The main features of this design are a stationary base, a rotating arm for the electromagnet, and a fixed lighting system.

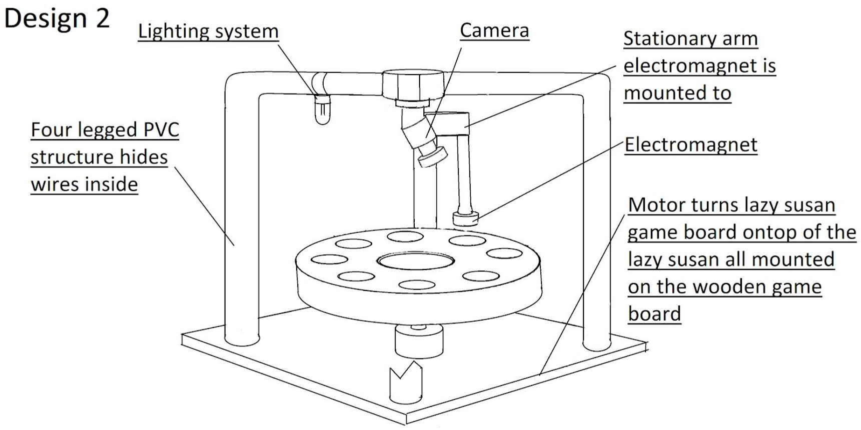
Design 2
Lighting system
Camera
Stationary arm electromagnet is mounted to
Four legged PVC structure hides wires inside
Electromagnet
Motor turns lazy susan game board ontop of the lazy susan all mounted on the wooden game board

Figure 2. Design Solution 2 Diagram

Design 2, as shown in **Figure 2**, has a similar support system and structure as Design 1. It starts with a wooden slab base with four vertical PVC posts at each corner. These four posts connect to elbow connectors and branch inward to connect at a center point. On top of the wooden slab, there is a Lazy Susan, which allows the game board to rotate. The game board will be mounted on top of the Lazy Susan. In the center of the Lazy Susan, is a stepper motor that will rotate the game board to an ideal position. Connected to the center point formed by the PVC pipe is a camera, which will be focused on the game board to obtain an accurate reading of the washer locations. This beam will also hold a long stationary arm that suspends the electromagnet above the washer location. A lighting system is attached on the opposite side of the beam. This will ensure that the camera has constant lighting during execution. The main features of this design are a lazy susan to create a rotating base, a stationary arm to suspend the electromagnet, and a fixed lighting system.

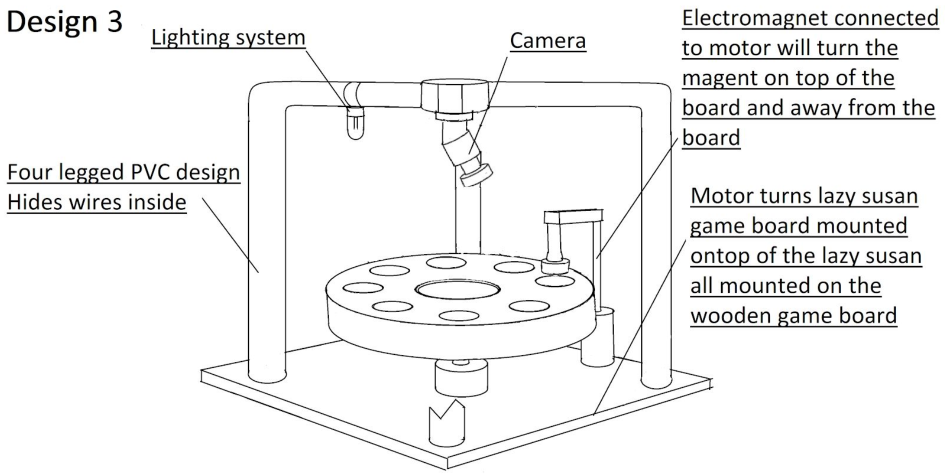
Design 3
Lighting system
Camera
Electromagnet connected to motor will turn the magent on top of the board and away from the board
Four legged PVC design Hides wires inside
Motor turns lazy susan game board mounted ontop of the lazy susan all mounted on the wooden game board

Figure 3. Design Solution 3 Diagram

Design 3, as shown in **Figure 3**, is a combination of both the previous designs. The design starts with a wooden slab base with four vertical PVC posts at each corner. These four posts connect to elbow connectors and branch inward to connect at a center point. On top of the wooden slab, there is a Lazy Susan, which allows the game board to rotate. The game board will be mounted on top of the Lazy Susan. In the center of the Lazy Susan, is a stepper motor that will rotate the game board to an ideal position. Connected to the center point formed by the PVC pipe is a camera, which will be focused on the game board to obtain an accurate reading of the washer locations. Attached to the wooden base will be a vertical beam connected to a stepper motor. This motor is attached to the electromagnet. The stepper motor will pivot the electromagnet a maximum of 90 degrees. By swinging the magnet 90 degrees, the electromagnet won't block the camera from seeing the game board. A lighting system is attached to one side of the horizontal structure. This will ensure that the camera has constant lighting during execution. The main features of this design are the two stepper motors and the fixed lighting system. The first motor will rotate the base into the correct position, and the second stepper motor will swing the electromagnet away from the game board for an accurate reading or towards the game board, to pick up the washers.

Concept Evaluation:

Table 2. Complex Decision Matrix

Evaluation	Weight factor	Design 1	Design 2	Design 3
Durability	2	1	3	2
Cost	2	3	2	2
Functionality	3	1	1	3
Speed	1	1	2	2
Safety	3	1	3	2
Reliability	2	1	2	3
Ease of use	1	1	3	2
Total Score	■	**18**	**31**	**33**

For our weighting factors, we used a range from 1-3, with 3 being the most important. We selected safety and functionality to be the two most important. This was because we need our design to be safe and effective at completing the tasks it was designed for. Next, a weight factor of 2 was assigned to cost, durability, and reliability. We need to make sure our design is not too expensive, rarely will break, and produces constant results. Lastly, we selected a weight factor of 1 for the speed and ease of use metrics. Our product doesn't need to be so fast that it breaks on its own, and the design doesn't need to have a sophisticated user interface because the human input will be at a minimum. Therefore, our product doesn't need to prioritize speed or ease of use.

Our final design choice was Design 3. It outranks the other two designs as shown in **Table 2**. For functionality, this design is superior. The pivoting electromagnet won't cover any markers, which allows our camera subsystem to acquire accurate images for the locations of said markers. Design 3 will also have good speed as it will rotate the base using a Lazy Susan while the electromagnet pivots onto the board. It will be durable, safe, and reliable due to the rigid structure, hidden wiring components, and the dual-motor system for accurately picking up the washers. Each of the designs features a user interface, so the hardest part of the design to use will be the setup. The main reason for selecting Design 3 was that it would produce the best results. As shown in Design 1 and Design 2, the camera subsystem will be blocked by the electromagnet. This will result in poor image recognition of where the game pieces are located. However, Design 3 will acquire the most accurate images of the game board due to the pivoting electromagnet.

Engineering Requirements for Camera Subsystem:

Table 3: Engineering Requirements for Camera System

Customer Requirement	**Engineering Requirement**
Accurate/Reliable	99% accurate in displaying color and location of each marker.
	Able to detect multiple colored markers and locations on a gameboard. Specifically, red, green, yellow, and blue markers in eight possible locations.
Cheap	Under $15
Fast Algorithm	Shorter than 20 seconds to complete the color identification and location of all markers
Safe	No potential hazard while using

Document Hardware:

For this portion of the project, the main components used were a USB 2.0 camera, the gameboard printed on paper, a lamp, a dry erase board, and 12 colored markers. The design is shown in **Figure 4**. There are 3 markers of each color green, blue, red, and yellow. The sheet of paper as the background will only be used for testing and will not be present in the final product. During setup, a lamp was tested at multiple locations to get the least amount of glare from the paper. Shining the lamp directly onto the paper created so much noise in the photograph that the software could not subtract the two images. The lamp was then placed next to the gameboard with the camera directly overhead. This lighting was sufficient for the camera and the image processing. The camera has an adjustable lens that can be used to focus the camera for a cleaner image. This functionality was used to get rid of some of the noisy background. Once the background picture was satisfactory, the markers were placed onto the gameboard in a circle. The light fixture needed to be constant for both the initial setup and the marker set up, and by doing so, the discrepancy between the two was minimized. The height of the camera was set at 12 3/4 inches. This height was determined iteratively so the camera would be in focus and get a clean shot of the game board within the frame. If the camera was too close to the game board the focus would become blurry or the edges of the game board would get cut off from the picture. Having a portion of the gameboard cut from the image was not a problem as long as it was consistent with both images. But the camera being blurry would cause issues with background subtraction. Being out of focus would cause the edges of the black outlined game board to not be subtracted from the image, and because of this, finding the centroid of the markers did not function properly.

The gameboard used originally was printed on a normal sheet of printing paper. The problem that arose from this was that once the sheet was moved around it would have small creases on it. These creases would capture light and show up in the final image as a shape when it was a blemish on the paper. To fix this issue the gameboard was printed on a thicker white sheet of printer paper. This game board was put on top of several other clean blank sheets of paper. This was done because the marker board that the game board sat on top of was brown and would mess with the picture slightly. Four strips of tape were applied to the 4 corners of the gameboard to make the position the exact same between pictures. The colored “washers” were cut from the sheets of paper into circles. The main reason for cutting them out was because the final design will have a fixed game board and shifting washers, so this will better emulate the final product. Another reason this was done was because the game boards with the colors already printed on them would have to be placed exactly on top of the first game board. This was difficult to accomplish and did not yield good results. The last reason for cutting out the markers was that the colors could be shifted easily to give the program a different look to test on. This allowed for more testing than the 2 tests that were provided with the lab. The web camera that was provided is an appropriate sensor for this project for several reasons. The first reason is the web-camera takes the picture at a resolution of 480x640, which is 8.9 x 6.7 inches. This is a sufficiently large photo for the task that we want to accomplish. The camera's ability to shift its lens focus makes it a good sensor because the camera can be set to different heights if needed. This makes the camera a versatile sensor.

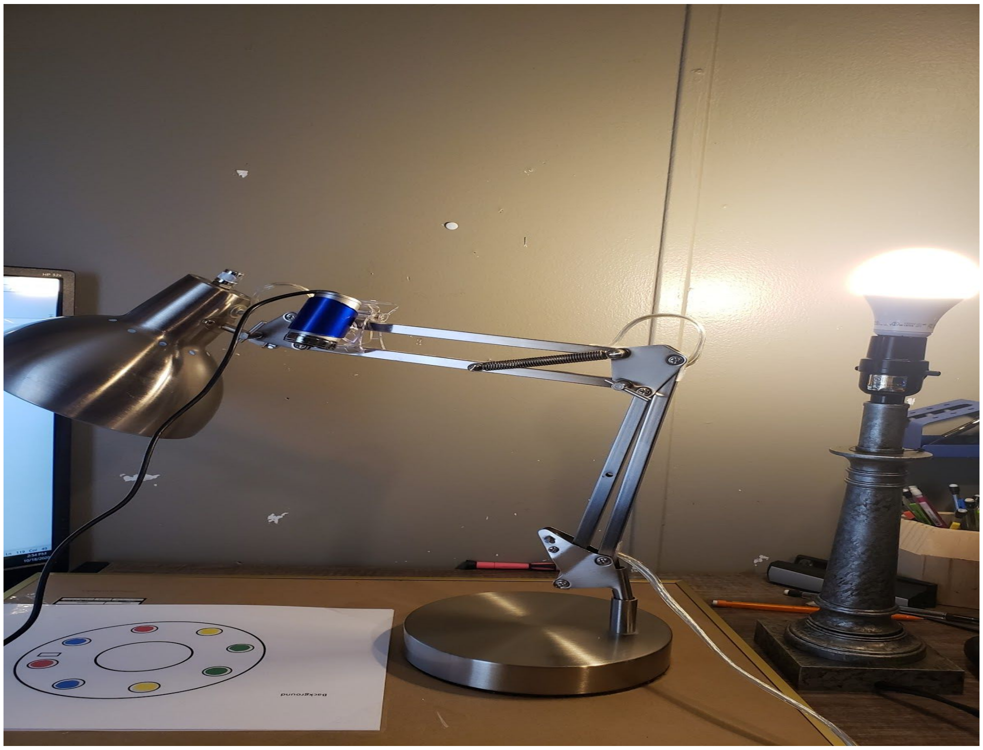

Figure 4. Hardware Design

Document Software:

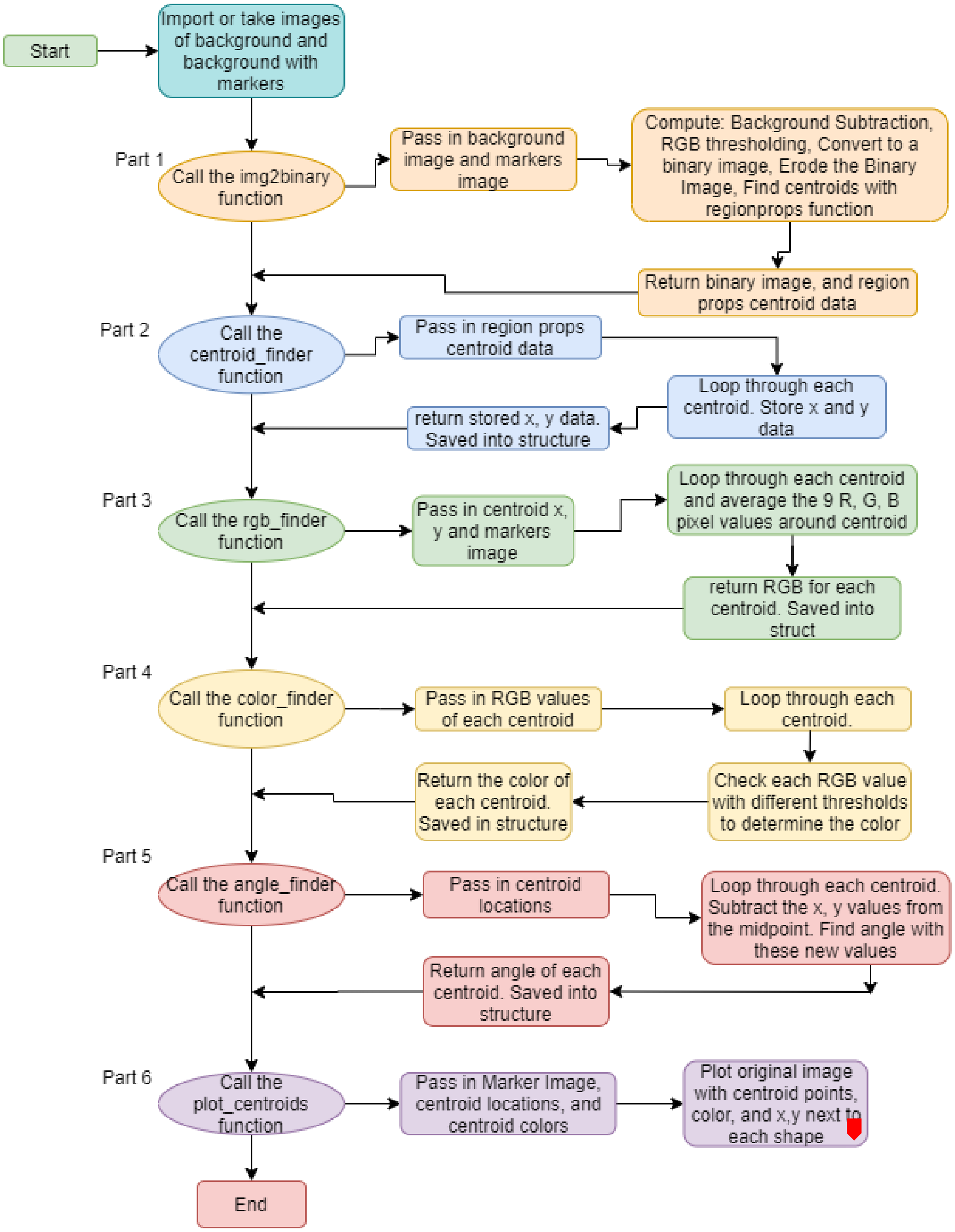
Start
Import or take images of background and background with markers
Part 1
Call the img2binary function
Pass in background image and markers image
Compute: Background Subtraction, RGB thresholding, Convert to a binary image, Erode the Binary Image, Find centroids with regionprops function
Return binary image, and region props centroid data
Part 2
Call the centroid_finder function
Pass in region props centroid data
Loop through each centroid. Store x and y data
return stored x, y data. Saved into structure
Part 3
Call the rgb_finder function
Pass in centroid x, y and markers image
Loop through each centroid and average the 9 R, G, B pixel values around centroid
return RGB for each centroid. Saved into struct
Part 4
Call the color_finder function
Pass in RGB values of each centroid
Loop through each centroid.
Check each RGB value with different thresholds to determine the color
Return the color of each centroid. Saved in structure
Part 5
Call the angle_finder function
Pass in centroid locations
Loop through each centroid. Subtract the x, y values from the midpoint. Find angle with these new values
Return angle of each centroid. Saved into structure
Part 6
Call the plot_centroids function
Pass in Marker Image, centroid locations, and centroid colors
Plot original image with centroid points, color, and x,y next to each shape
End

Figure 5. Software Flowchart

The MATLAB code begins by defining an empty structure called gameState. After, two images are taken or imported. The first image is a background image, and the second image is the background image with the markers overlaid on top. These two images are passed through into a user-defined function called img2binary. As shown in **Part 1** of **Figure 5**, the four main steps within this function are background subtraction, RGB thresholding, converting the image to a binary image, eroding this binary image, and determining the centroids of the binary image. First, the size of both images is checked to make sure that they have the same dimensions. Next, the image containing the markers is subtracted from the background image, and saved into a new background subtraction variable. Then, this new image is looped through each pixel to determine if it is greater than a set threshold. The set thresholds for red, green, and blue are 50, 50, and 50. If any of these RGB values are greater than the threshold, then the pixel will be set to an arbitrary color, which was selected as green. After looping through each pixel, this new image is then converted to a binary image using the built-in function im2bw. This binary image is then eroded to remove noise from the image. Next, the built-in function regionprops is called on the eroded binary image, and the resulting values are stored in a new variable called STATS. After calling the regionprops function, both the STATS variable and eroded binary image are returned to the function call.

After calling the img2binary function, the centroid finder function is called. As shown in **Part 2** of **Figure 5**, the variable STATS is passed into the centroid finder function. The centroid values for each of the shapes are contained within the STATS matrix. A variable called items is used to store the size of the STATS matrix. A FOR loop is used to loop through the size of the STATS variable. Inside this loop, each of the centroid values is stored into 2 temporary variables. Once the values are separated from the extra information contained in STATS, they are concatenated into a single matrix. Lastly, the concatenated matrix is returned into the function call and assigned to the field rgb_centroid_values in the gameState structure.

After calling the centroid finder function, the RGB finder function is called, as shown in **Part 3** of **Figure 5**. The values passed into the RGB finder function are the centroid locations from the gameState structure and the picture of the gameboard that contains the markers. To determine the duration of the FOR loop used later, the size function is used on the centroid locations variable. Next, a temporary variable is initialized, which will store the RGB values of each centroid. Then, a FOR loop is used to run through each centroid location, and temporary variables for x and y are set to the rounded value of the centroid's location. Then, three additional values are initialized to zero. These values will be used to store the sum of the three different color values red, green, and blue. Within the same loop, a double FOR loop is set up to go through 9 different pixel values, with the centroid being the middle pixel. By running through 9 different pixels, the color found will be an average of multiple pixels instead of one pixel, which will decrease errors when determining the color. The first loop will run through 3 values in the x-direction, and the second nested loop will run through 3 values in the y-direction. These two nested loops will run through 9 total pixels for each centroid. The RGB values for each pixel are separated into red, green, and blue variables, and turned into 16-bit integers. Then, each red, green, and blue value is summed with the values associated with the same color, and after looping through each of the 9 pixels, the temporary values are then taken and divided by 9. This is done so that the average of the 9 pixels around the centroid is accurately represented. By taking the average of the 9 pixels, it eliminates the possibility of having a single pixel be confused for a different color. Once this operation is completed the 3 color values are concatenated together and assigned a variable. This variable is then returned to the gameState structure and stored as the rgb centroid values field.

After calling the RGB finder function, the color finder function is called, as shown in **Part 4** of **Figure 5**. The RGB centroid values variable is passed into the color finder function. This variable is set up to represent the three colors in each column. The first column is set up to represent red, the second column is set up to represent green, and the third column is set up to represent blue. Therefore, the centroid values have 3 columns and an adjustable number of rows depending on the number of markers placed on the gameboard. A FOR loop is used to determine the color of each of the markers. This loop is set to run to the length of the number of rows in the centroid values matrix. Inside the loop, an if statement is set up to determine if the marker is yellow, red, blue, or green. To determine if the color is yellow, the if statement checks if the first column's value is higher than 180, and if the second column is greater than 180. If both values are greater, then the color is determined to be yellow. If the color is not yellow then the loop checks if the second column is less than 150, or if the first column is greater than 180. If both of the values are then the color is red. Next, the loop checks if the first column is less than 150, and if the third column is greater than 150. If this is true then the color is blue. If all 3 of these statements are false, then the color is determined as green. This variable is then returned to the gameState structure and stored as the centroid colors to be used later in the program.

After calling the color finder function, the angle finder function is called, as shown in **Part 5** of **Figure 5**. The only variable passed into the function is the field from gameState structure containing the centroid locations. Next, an empty array, which will store the angles, is initialized, and the number of centroids is determined using the size function. Then, for each centroid, two different displacement values are calculated. The first is the displacement value for x, which is the x value of the centroid subtracted from 320. The next displacement value is for y, where the y value of the centroid is subtracted from 240. By subtracting the midpoint of the image by each centroid, a corresponding angle can be calculated. After calculating both displacement values, the built-in atan2d function is used. This function takes in both values and will output a corresponding angle in degrees. The calculated angle is then stored in a vector. After looping through each centroid, the function will then return a vector that contains an angle for each centroid. This vector is then stored as the angle field in the gameState structure.

After calling the angle finder function, the centroid plotting function is called, as shown in **Part 6** of **Figure 5**. Three parameters are passed into this function. The first parameter is the original image containing the markers. The second parameter is the field from the gameState structure containing the centroid coordinates. The last parameter passed is the field from the gameState structure containing the colors found for each marker. Inside the function, the original image is plotted. Then, a loop that runs for each centroid will plot the centroid, plot the rounded x and y values for the coordinates of each centroid, and the color of each marker. The text values will be plotted next to each marker. After the image is displayed, the centroid plotting function ends and returns nothing. After this function returns, the code ends.

Megafist-TEK Engineering
Estimated Financial Scenario

Start-up Costs

Personnel	5 Engineers @ $55K/yr + President @ $75K/yr + Admin. Asst. @ $25K/yr = $375,000
Fringe Benefit (FB)	A fringe benefit is a form of pay for the performance of services. For example, you provide an employee with a fringe benefit when you allow the employee to use a business vehicle to commute to and from work. Assume Fringe Benefit Package @ 36% (incl. employee's SS tax, vacation, holidays, medical, retirement (401K), dental, life insurance, relocation, unemployment insurances, etc): (5 x $55,000 + $75,000 + $25,000) x 0.36 = $ 135,000 *Note: Federal Insurance Contributions Act (FICA) tax (Social Security and Medicare) is imposed by the federal government on both employees and employers. The entire FICA percentage of 15.3%* ● *Employee's pay 6.2% for SS and 1.45% for the Medicare (this is not included in your cost)* ● *The employer is liable for 6.2% Social Security and 1.45% Medicare taxes=7.65%*
Building	Initially rent a suite of offices with 2 engineers/office (12' x 14'), an office/conference room for President (12' x 20'), and a reception/office area of 16' x 20'. (3 cubicles) x (12' x 14'/cubicle) + President office of (12’x 20’) + Reception/office area of (16' x 20') = 1064 sq ft Use nominal figure for office space in industrial park sectors of` Clemson area, $9.50/sq ft/mo. Then the lease rate for office space will be $0.79/sq ft/mo x 1064 sq ft = $841/mo. = $10,087/yr.
Furniture	Rental of a desk, chair, credenza set will run about $60/mo. Need 7 sets for a total monthly expenditure of $420/mo = $5,040/yr The remaining equipment, furniture and software expenses are estimated to be about 7 computers @ $1500/computer $10,500 7 sets of general software @ $1000/set $7,000 Specialized software $18,000 Copier, printer $4,000 Table and chairs for conference room $3,888 7 telephones @ $35/ea $245 Total $48,673
Phone and Internet	According to Bell South, the cost of a combined voice/data line, is $70.00/mo for operation. For 7 telephones the total cost will be $5,880/year. Assume that long distance calls add another 40% to this to get a total estimated annual phone cost of $8,232.

Travel	Another cost item which will be important is travel. There will have to be continual contact with potential clients, attendance at selected technical conferences and workshops, and visits to plants or other locations where potential clients might be. Assume (modestly) that this will that the cost per local trip is $200 and the cost per out-of-state trip is $3,000 there will be 2 of each trip each month $6,400/mo for the first year, or an annual total of $76,800.
Interest	Capital (i.e. money) is needed to fund these initial purchases as well as to underwrite operating expenses until a revenue stream is established by selling engineering services to customers. Assume that through personal contacts a credit line of $800,000 has been established. This is to be repaid over the period of a year with 11 equal payments starting 1 month after the loan date. The negotiated interest rate is 5% per year. The monthly payment M is calculated from = $74,726 Where P is the principal amount ($800,000), I is the interest rate (5%), and q is the number of payments to be made (11). From this, Debt Service = Total interest paid in year = 11 x M - P = $21,983.

Cost Estimate

Salaries	$375,000
FB @ 36%	$135,000
Building	$10,087
Furniture	$48,673
Debt service	$21,983
Travel	$76,800
Internet and Phone Service	$8,232
Total Costs	$675,775

Overhead Calculation

Now we will estimate the Overhead (Indirect Technical Expense) we must charge to recover our costs. This cannot be too large, or else we will price ourselves out of business. On the other hand, we must be realistic, or else we will go broke, and therefore out of business.

Assume that the first year, the 5 engineers will be at least 75% "sold", i.e., 75% of their total time can be charged to customers. Then we can bill

5 engineers @ 75% sold (salaries billable to clients)	$206,250
FB @ 36% (FB billable to clients)	$74,250
Total Billable to Clients	$280,500

The remaining salary dollars and FB's must be charged to overhead.
Total Expenses = Total Costs - Total Billable to Clients = $395,275 (Overhead Number)

This implies an Overhead rate of
OH rate = ($395,275/$280,500) x 100% = 140.92%
= (Overhead Number / Total Billable to Clients) x 100%

This implies that every labor dollar (at the "loaded" rate, i.e. with FB's) must be increased by a factor of (salary + (salary * FB/100%)) * ((1+ (OH rate/100%) + (5% profit/100%)) in order to recover the costs of doing business and make a profit (assuming a 5% profit). This is the figure that you will use when estimating the cost of a contract to a customer in a proposal. An overhead rate of 150% means that for each $1.00 of direct labor budgeted for a project; $1.50 needs to be budgeted for overhead costs.

Using the Overhead Number

You estimate that a project will take 1 week (40 hours) of your time, i.e. what does it cost for one week of an engineer's time. How much do you bill your client for this time?

Bill to Client

$$= \left[\frac{\text{1 week work}}{\text{52 weeks per year}} \bullet \left(\frac{\text{salary = \$55K}}{\text{year}} + \frac{\text{FB = 0.36}\square\text{\$55K}}{\text{year}}\right) \bullet \left(1+\frac{\text{overhead rate}}{100\%}+\frac{\text{profit =5\%}}{100\%}\right)\right]$$

= $3,537.46

ECE 4950 Project 2 – Sensor, Financials, and Website Rubric

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

Group Number and Name: Group 13, Team Megafist. Aho, Anderson, Cuttino. Liggett, and Moran

Score	Pts		Performance Indicators
5	5	**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.	g.1
4	5	**Page 1: Title, Group Name, Group Members, and Date** **Executive Summary** (1 concise, well-written paragraph) Provide an overview of this project. Briefly describe what you did and what you learned.	g.1
8 11 10	30	**Overall System Design** **Page 2: Engineering Requirements for the Entire System** (~1 page) Make a three-column table that lists a Customer Requirement in the first column and the resulting Engineering Requirement(s) in the second column. Note that a customer requirement may branch to multiple engineering requirements. The third column should contain the test that will be done on the prototype to verify that the design chosen meets each requirement. **Page 3-4: Design Solutions** (~2 pages) Use a brain-storming session to generate concepts. Document your top three most feasible ideas with sketches and brief descriptions while providing the main features of each concept. **Page 5: Concept Evaluation** (~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. Describe your final design choice, including how the final decision process, which may include more than just the Decision Matrix result (~1 paragraph). The robot mockup will be graded based on the following criteria: 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?	c.2 c.1 c.4
17	20	**Subsystem Design** **Page 6: Engineering Requirements for the Camera subsystem** (~1 page) In the context of just the Camera-as-a-Sensor, make a two column table that contains ta 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	c.2

		generate multiple engineering requirements. For example, the customer will want an “accurate” system, the Engineering Requirement could be 99.5% detection success.	
6	10	**Pages 6-7: Document Hardware** (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?	
4	5	**Pages 7-8: Document Software** (3 pages) Using Flowcharts, state diagrams, data structures etc. describe how the software is implemented. There is no need to include the source code.	
5	5	**Page 10-14: Financial Analysis** Provide a financial analysis that examines turning your group into a start-up company. Use the spreadsheet provided to make calculations and report your results using the MS Word document template and include here.	c.3
		Page 15: Grading Sheet	g.1
20	20	**General Website Format** – ***provide link here:*** sca2.people.clemson.edu 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) All links should be relative to the starting directory so that it can be viewed offline. d) Follow the website guidelines, including accessibility compliance at: http://akapadi.people.clemson.edu/ece4950_websiteDesign.html	g.1

90