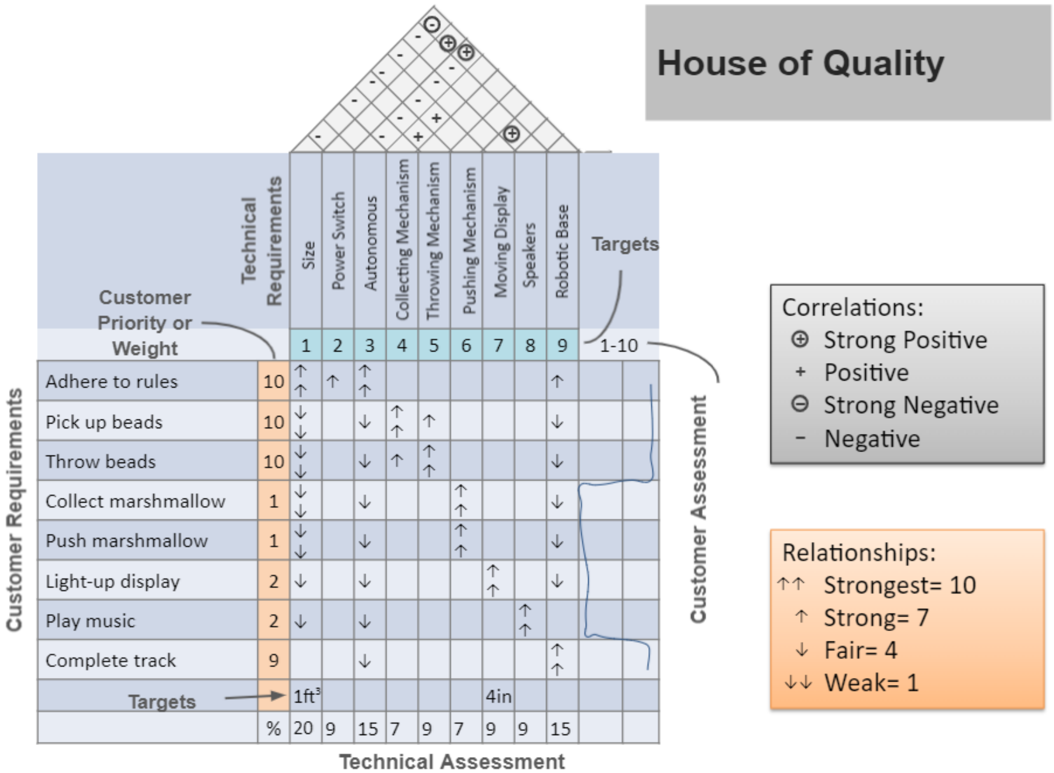
Senior Design Group 301
Concept Selection
10/26/21

House of Quality



The House of Quality chart gave us a visual and better understanding as to what technical requirements are worth the most to us. The Customer Requirement weights were decided in accordance to the amount of points awarded for completing the respective tasks. Additionally, the rules have to be followed, so they are worth a full 10 out of 10 weight. The correlations were decided based on how negatively or positively the requirements affect each other. Similarly, the relationships were decided based on how negatively or positively the customer requirements (needs) affect the technical requirements. The final percentages (Technical Assessment), were decided based on an equation we created. We took the positive arrows minus the negative arrows for each category. Then we subtracted this number from ten. This gave us a total of 93% when adding all the weights. To increase clarity on what we think is more important on the design, we bumped the size requirement to 20% (since it is absolutely required), bumped the robotic base to 15% (to match the autonomy), and increased the throwing mechanism to 9% (to distinguish the importance over the collecting of beads and the marshmallow mechanism). From these results, we believe that the level of importance of each requirement is as follows:

Size > Autonomy > Robotic Base > Power Switch > Throwing Mechanism > Moving Display > Speakers > Collecting Mechanism > Pushing Mechanism.

One thing to note with our final values, is that in order to launch our beads, we need to collect them. So, the effective importance for mechanisms related to beads is 16%. Since we do not have competitors, our customer assessment is a visual representation of the need's priorities.

AHP Tables

Brain	Weight	Arduino	Raspberry Pi	MSP430	MSP432	FPLD/FPGA
Accuracy	0.64	2	4	2	3	5
Ease of Implementation	0.26	3	5	4	4	1
Cost	0.1	4	4	5	3	1
Score		2.46	4.26	2.82	3.26	3.56

The Raspberry Pi is the best brain choice because of its ease of use and its accuracy, which was based on it's RAM, frequency, and pin count.

Coding Language	Weight	Python	C	C#	C++	Scratch	RaspberryOS	Java	JavaScript	VHDL
Accuracy	0.64	8	9	6	9	3	7	5	5	4
Ease of Implementation	0.26	8	9	2	8	1	8	5	6	2
Cost	0.1	0	0	0	0	0	0	0	0	0
Score		7.2	8.1	4.36	7.84	2.18	6.56	4.5	4.76	3.08

C is the best coding language to use because of its direct interface with hardware and because of the team's familiarity with the language.

Wheels	Weight	Omni	Standard	Ball	Mecanum	Track
Accuracy	0.64	5	3	5	5	1
Ease of Implementation	0.26	2	5	1	2	4
Cost	0.1	1	5	1	1	1
Score		3.82	3.72	3.56	3.82	1.78

Although the omni wheels scored the highest, we settled on standard wheels due to their simple construction and movement method.

Navigation	Weight	Track center line	Track Barriers	Model Predictive	A*
Accuracy	0.64	5	5	4	6
Ease of Implementation	0.26	6	5	4	3
Cost	0.1	6	5	4	3
Score		5.36	5	4	4.92

Line tracking was selected for maintaining orientation on the track because of the simple programming required.

Motors	Weight	Tiny Pager	Stepper	Linear	Brushed	Brushless	Spur Gear
Accuracy	0.64	2	6	1	4	5	3
Ease of Implementation	0.26	1	5	1	6	6	5
Cost	0.1	6	3	1	5	2	4
Score		2.14	5.44	1	4.62	4.96	3.62

Although the stepper motor scored the highest, we decided on the brushless motors because of their high performance and low power draw.

Power Supply	Weight	Lead	Lithium	LiPo	NiMH
Accuracy	0.64	2	3	1	4
Ease of Implementation	0.26	1	2	3	4
Cost	0.1	2	1	3	4
Score		1.74	2.54	1.72	4

An NiMH battery will be employed because of their abundance, ease of charging, and high efficiency.

Sensors	Weight	LiDAR	Sonar	Camera	Radar	Light	Ultrasonic	Transmissive IR	Reflective IR	Gyroscope
Accuracy	0.64	9	1	7	1	1	5	8	6	3
Ease of Implementation	0.26	4	1	3	1	5	1	9	9	2
Cost	0.1	1	8	2	3	7	9	5	9	4
Score		6.9	1.7	5.46	1.2	2.64	4.36	7.96	7.08	2.84

Transmissive IR will be used for barrier detection, but we will also be using cameras to identify the bead receptacles such as the nets and red solo cups.

Displays	Weight	iPod Touch	Multimedia D.	Android Tablet	Android Phone	LCD	Circuit Speaker	Accordion Arm	Folding Joint
Accuracy	0.64	5	8	7	6	6	8	8	6
Ease of Implementation	0.26	4	8	4	4	6	7	8	8
Cost	0.1	7	5	6	1	8	8	6	7
Score		4.94	7.7	6.12	4.98	6.2	7.74	7.8	6.62

A multimedia device will be attached to a motor of some sort that will move the device rotationally 4+ inches.

Bead Collector	Weight	Fan	Vacuum	Arm	Claw	Fish Hook	Grappling Hook	Slide
Accuracy	0.64	6	7	9	8	6	7	3
Ease of Implementation	0.26	2	4	5	6	8	7	9
Cost	0.1	7	2	1	3	8	6	9
Score		5.06	5.72	7.16	6.98	6.72	6.9	5.16

We will be using a robotic arm to collect the beads from the trees due to the high level of accuracy.

Bead Storage	Weight	Retractable Grate	Tub	Tub w/ Door
Accuracy	0.64	2	1	3
Ease of Implementation	0.26	1	3	2
Cost	0.1	2	3	2
Score		1.74	1.72	2.64

With our proposed design, we decided bead storage will not be necessary, and the beads will just be held by the arm until they are thrown.

Bead Launcher	Weight	Vacuum	Flywheel	Flywheel/Belt	Slingshot	Arm
Accuracy	0.64	1	3	4	2	5
Ease of Implementation	0.26	5	4	3	3	4
Cost	0.1	4	3	3	5	2
Score		2.34	3.26	3.64	2.56	4.44

Although the arm scored the highest for its launching capabilities, we decided on using a flywheel mechanism to eliminate the need for a storage area.

Marshmallow Mech.	Weight	Angled Fan Blade	Triangle Bumper	Moving walls Lat.	Moving walls Lon.	Rotating Baskets	Accordian Arm	None	Retractable motorized collector w/ sensor
Accuracy	0.64	4	2	6	6	4	7	1	8
Ease of Implementation	0.26	6	7	2	2	6	3	8	1
Cost	0.1	6	4	2	2	6	6	8	1
Score		4.72	3.5	4.56	4.56	4.72	5.86	3.52	5.48

To push the marshmallows from the roadway, we will be using an arm that extends out linearly, with a rotatable cup that fits around the marshmallow to push it along.

Marshmallow storage	Weight	Tube	Reel	Bin	Net
Accuracy	0.64	1	4	2	4
Ease of Implementation	0.26	1	2	1	4
Cost	0.1	3	1	4	4
Score		1.2	3.18	1.94	4

To store the marshmallows, we found a net that simply drags it along is the best choice due to its simplicity and ease of implementation. However, we have decided to use a more static cup design that emulates a net.

Base	Weight	Plywood	Aluminum Sheet	Aluminum Bead	Plastic	Metal Corner Bead
Accuracy	0.64	3	5	4	1	4
Ease of Implementation	0.26	5	5	5	5	5
Cost	0.1	4	3	5	4	5
Score		3.62	4.8	4.36	2.34	4.36

The aluminum sheet was chosen for its sturdiness, light weight, and cost.

Attachment	Weight	Glue	Screws	Velcro	Command	Nails	Bolts
Accuracy	0.64	1	5	2	2	5	5
Ease of Implementation	0.26	5	5	5	5	5	5
Cost	0.1	5	4	5	5	4	4
Score		2.44	4.9	3.08	3.08	4.9	4.9

The screws, nails and bolts were chosen due to their simplicity when it comes to implementing, as well as their strength and cost, however, we are also likely to use glue, velcro, and command strips in our build, depending on the component placement.

Wiring	Weight	Copper	Silicone Cables	Polyurethane	Polyvinyl Chloride	Solid	Stranded
Accuracy	0.64	5	6	6	4	5	4
Ease of Implementation	0.26	4	5	6	4	4	5
Cost	0.1	6	2	5	2	1	2
Score		4.84	5.34	5.9	3.8	4.34	4.06

Polyurethane cabling will be used because of their low cost, and we already have access to an abundance of them.

Decor	Weight	FAMU/FSU Colors	COE Colors	LED Lights	Mascots
Accuracy	0.64	4	4	1	4
Ease of Implementation	0.26	3	4	1	3
Cost	0.1	4	4	1	4
Score		3.74	4	1	3.74

The COE colors were chosen to represent both institutions, while keeping a consistent color scheme. It would also be easier to implement than LED Lights or the Mascots since those would take space on the robot.

Connections	Weight	Solder	Breadboard
Accuracy	0.64	2	1
Ease of Implementation	0.26	1	2
Cost	0.1	1	2
Score		1.64	1.36

Soldering was chosen for its ability to create a secure attachment between the components and the devices. However, a breadboard will be implemented into our final design as well, as it allows for certain components to be hot swappable, in case there is a faulty component.

The following table was used to determine the importance of cost, ease of implementation, and accuracy. In the AHP tables, the weights are translated directly and can be seen as percent importance. Each concept was then rated against each other from 1 to the number of concepts in the category. The higher the ranking, the better the concept. The score is the summation of the concept rank times the weight of the category. The highest scores gave us the theoretical best concept.

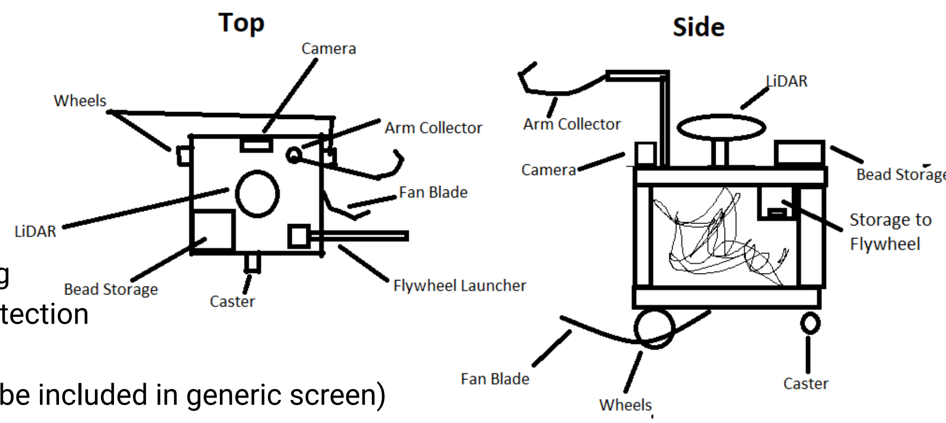
	Cost	Ease of Implementation	Accuracy	Geometric Mean	Weights
Cost	1	0.3333333333	0.2	0.405480133	0.104729434
Ease of Implementation	3	1	0.33333333	0.9999999967	0.2582849937
Accuracy	5	3	1	2.466212074	0.6369855723
				Results sorted:	Weights
Scale of importance:	1 = Equal	3 = Moderate	5 = Strong	Accuracy	0.64
				Ease of Implementation	0.26
				Cost	0.1

Pugh Selection Matrix

The following prototype options are from the original concept generation.

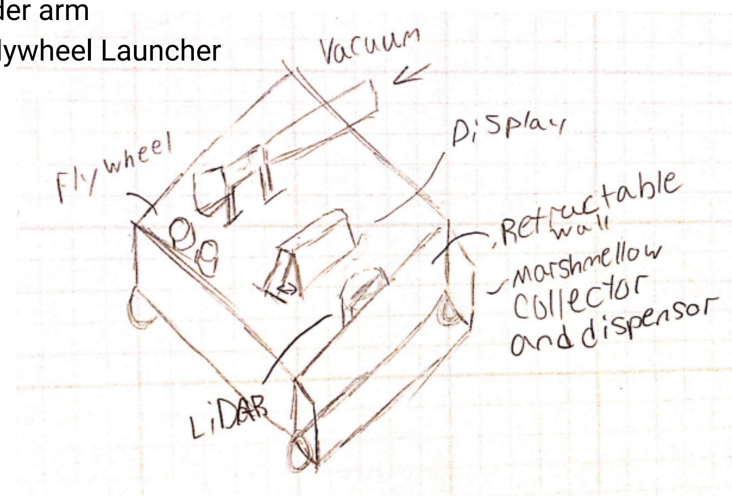
1:

- Raspberry Pi 4
- C
- 2 Standard Wheels, 1 Caster
- Center Line Tracking
- Barrier Tracking
- Brushless Motor
- LiPo Batteries
- LiDAR for Barriers
- Reflective IR for Line Tracking
- Camera for Net/Tree/Cup Detection
- Generic Screen
- Simple Circuit Speaker (may be included in generic screen)
- Folding Joint for Multimedia Display
- Fish Hook for Bead Collection
- Container to Store Beads Connected to Launching Mechanism
- Flywheel Launcher for Beads
- Fan to Collect Marshmallow
- No Storage - Drag it Along
- Aluminum Base
- Bolts for Attaching Base and Wheels, etc.
- Polyurethane Connection Cables
- COE Colors and Text
- Soldered Connections to Components
- Breadboard to Connect Components to Base



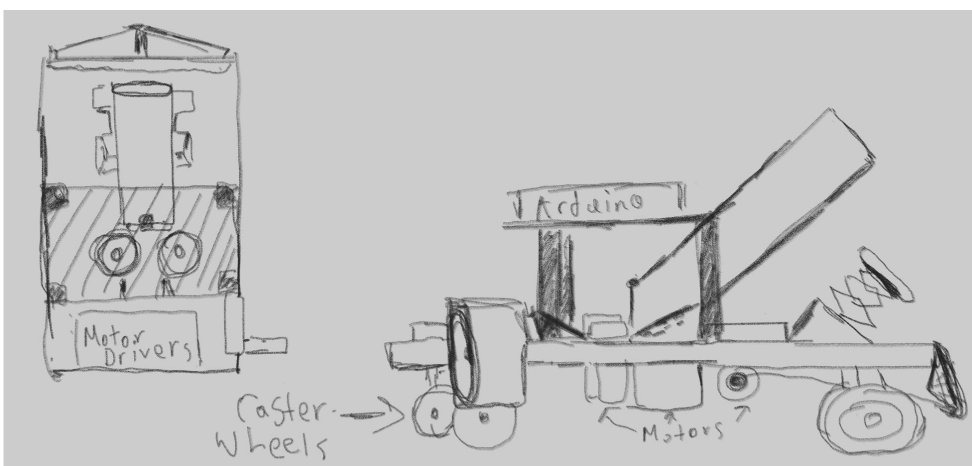
2:

- Raspberry Pi 4
- Python
- Standard wheels with caster
- Line Tracking and Barrier Tracking
- Linear Motor
- Lithium battery
- LiDAR
- Multimedia Display with an accordion extender arm
- Vacuum Arm with a retractable grate and a flywheel Launcher
- Moving extensions of walls Longitudinal
- Aluminum Sheet
- Screws and bolts
- Copper Wiring
- COE Colors
- Solder



3:

- Brain - Arduino Mega 2560
- Coding Language - C++
- Wheels - Standard(x2) and Caster(x1) for simple programming
- Linear Movement/Steering - Line and Barrier Tracking for tracking of the environment and the robots position
- Motors - Brushless for more efficient operation of the wheels
- Power Supply - Lithium
- Sensors - LiDAR , Camera and IR for environmental scanning and data collection
- Display and speaker - Android Phone and Robotic accordion extender arm
- Bead Collector - Vacuum Collector
- Bead Storage - Vacuum Collector container with door and grate
- Bead Launcher - Flywheel Launcher
- Marshmallow Mechanism - Triangular Bumper pushing marshmallow aside and Accordion arm extender with caster wheel to push marshmallow into the opening on the track
- Physical Base - Aluminum Sheet for lighter chassis
- Attachment Methods - Hot Glue and screws and bolts
- Wiring - Polyurethane Cables
- Decorations - COE Colors and Goose for school spirit
- Connections - Breadboard for ease of swapping out parts and soldering for parts that need to be soldered



Pugh Chart 1

Criteria	Weight	Option 1	Option 2	Option 3	Option 4 (Baseline)
Accuracy	0.64	-1	0	1	-
Ease of implementation	0.26	1	0	-1	-
Cost	0.1	1	0	-1	-
Score		-0.28	0	0.28	-
Continue?		No	Yes	Combine	Combine

Pugh Chart 2

Criteria	Weight	Option 2	Combination	AHP
Accuracy	0.64	-1	1	-
Ease of implementation	0.26	-1	1	-
Cost	0.1	0	1	-
Score		-0.9	1	-
Continue?		No	Yes	-

The Pugh chart gave insight to the whole design of the robot and how the components worked together in each category. We thoroughly analyzed and compared each component to the AHP results and tallied up which designs had the most desirable components for each category to fill in the Pugh charts. Based on the first chart we decided to not use option 1 as it had the worst ranking due to the lack of accuracy. We decided to combine option 3 and its high accuracy with our AHP design, which used the top components in each AHP category. This combination method led us to our final design which is discussed below.

Final Selection

- Brain - Raspberry pi
- Coding Language - C
- *Wheels - Standard(x2) and Caster(x1) for simple programming
- *Linear Movement/Steering - Line and Barrier Tracking for tracking of the environment and the robots position
- *Motors - Brushless for more efficient operation of the wheels
- Power Supply - NiMH
- Sensors - camera, transmissive IR for environmental scanning and data collection
- Display and speaker - Multimedia device and rotating arm
- Bead Collector - Arm
- Bead Storage - Arm
- Bead Launcher - Flywheel
- Marshmallow Mechanism - Accordion arm w cup
- *Physical Base - Aluminum Sheet for lighter chassis
- Attachment Methods - screws and bolts
- Wiring - Polyurethane Cables
- Decorations - COE Colors for school spirit
- Connections - Breadboard for ease of swapping out parts and soldering for parts that need to be soldered

The final design is based on a combination between our third high fidelity concept and the best concepts found from our AHP tables. We also made minor tweaks based on the team's familiarity with certain components and concepts. The robot will use the Raspberry Pi while using C and it will be based on a three-wheeled system: two powered and one passive caster. It will employ both line and barrier tracking and will use a camera and transmissive IR. Brushless motors will be used for Stanmovement. A NiMH RC-car battery will be used for power delivery. An arm will be used in conjunction with a flywheel launcher to collect and throw the beads. An accordion with a small cup will be used to move and push the marshmallow. An aluminum sheet will be used as the physical base along with screws and bolts. Polyurethane cables will be used to connect the battery and boards together. The robot will be decorated in blue to represent the College of Engineering's spirit. This final design appears to give us the best shot at victory at the competition.