

Rodent Automated Training (RAT) Cage

BME 401 Final Report

Ellie Degen, Sara McCutcheon, and Zoe Orenstein

Table of Contents

Project Description	2
Review of Problem.....	2
Project Scope.....	3
Prototype Analysis.....	3
Design Specifications.....	4
Verification Results.....	6
1. Screens.....	6
2. Capacitive Touch Sensor.....	6
3. Solenoid Valve.....	6
4. Lights.....	9
5. Speaker.....	9
6. Cage Design.....	9
Validation Results.....	10
1. Screens.....	10
2. Capacitive Touch Sensor.....	10
3. Solenoid Valve.....	11
4. Lights.....	11
5. Speaker.....	12
6. Cage Design.....	12
DesignSafe Analysis.....	15
Conclusion.....	16
Appendix.....	18
A. Cage.....	18
Drawings.....	18
Overhead Cage View.....	32
Assembly.....	32
B. Parts List.....	33
C. Valve Cradle Drawing.....	37
D. Wiring Diagram.....	38
E. Circuit Diagram.....	39
F. Code.....	40
G. Manual.....	51
References.....	54

Project Description

Review of Problem

A wide variety of biomedical research relies on animal models; these models reveal causal relationships between biological mechanisms and behavioral symptoms applicable to human pathology (1) . Animal-based studies can predict drug effects, develop alternative therapeutics for the treatment of human diseases, hone in on higher order cognitive and learning phenomena, and inform decision across a variety of other biological areas (2) .

While a powerful tool, animal models are extremely difficult to duplicate – both between laboratories and within labs themselves. In fact, studies considering the reproducibility of rodent behavioral studies show that the experimenter who handles the rodent introduces variability that cannot be eliminated by physiological methods or by standardizing the genetic backgrounds of the rodents. This human influence most often occurs when the handler transports the rodent between the home cage and training area. The transportation diminishes the validity of experiments, as the day-to-day behavior of a rodent changes with its level of contact with the handler on that day. The handler's gender can also influence how the rodent acts during their training sessions, further contributing to uncontrolled variability (3) . Handling of the animals between training sessions can induce stress in the rodents and impact their cognitive performance during training (1) .

In addition to combatting variability arising from direct human/rodent contact, researchers have to spend hours coming into lab and training the rodents individually. This wastes time they could be spending doing analysis, experimental design, writing grants, and gathering data. Additionally, certain cage environment conditions such as lights and sound are altered during certain studies, so altering one cage's environment for a study concurrently impacts all other cages and experiments. The Rodent Automated Training (RAT) Device address these issues,

eliminating the handler-induced and cage environment variability and the time wasted by researchers training rodents by entirely automating the training process in the home cage.

Project Scope

Client: Keith Hengen, a professor of biology at Washington University in St. Louis and Principal Investigator in the Hengen Lab. Sam Brunwasser, PhD. candidate in the Hengen Lab.

Need: Dr. Hengen's lab needs to more efficiently gather informative neuronal network data by decreasing the amount of time and manual labor required for training rodents under differing experimental conditions and decrease the variability in data that arises from human-rodent interaction.

Scope: To address the Hengen lab's need, we will design a rodent cage that facilitates experimental individualization and automation of data collection; incorporate automated light and sound control, automated water delivery, allow for potential visual experiments and rodent response indication, have an opening for a monitoring camera, and will also fit onto the shelves housing the lab's current rodent cages. A software package will enable control of the rodent's sound, light, watering, and visual stimuli, and allow for exportation of this data in usable data types. This specialized cage and controlling software will be functional and available to Dr. Hengen by April 29, 2019, after the BME day presentation.

Prototype Analysis

This prototype must meet certain specifications based on the need from the client and the scope of the project. The best design solution was determined in the Progress Report via a Pugh chart and then updated as the project changed to accommodate the client's needs. Verification and validation testing were used to confirm that the prototype works in the way it is meant to. Risk was evaluated using DesignSafe. The results from these tests will be discussed below.

Design Specifications

The initial cage design was determined with a Pugh chart in the Progress report, and an in depth analysis of that design can be found there. However, due to the change from a rat to a mouse in the client's study, the cage design had to be changed in order to accommodate the anatomy of the mouse and the desired field of view for the study. The final cage design can be found in Appendix A, along with individual renderings of the each wall with dimensions.

Instructions on how to put the cage together can be found in Appendix A.

The cage has a unique back wall to accommodate two spouts, which are each individually connected to a corresponding solenoid valve. Each spout is connected to a capacitive sensor, which, when activated, will trigger the opening one of two solenoid valves, allowing water to flow through the valve and be available at that spout.

Behind the back wall and the spouts is a monitor which can display two visual stimuli for the mouse to choose between. One of these stimuli will be considered the "correct" stimulus, and water will be made available at the corresponding spout if the mouse chooses that stimulus. If the mouse chooses the "incorrect" stimulus, no water will be available, and there will be an additional timeout of the system.

When the mouse chooses the correct stimulus, a tone will play from a speaker, reinforcing the behavior, and linking the sound to the water reward. Additionally, the speaker will indicate the beginning of training by playing a tone when the screens light up for the first time.

The cage houses two mice, with the mouse under observation in the larger area with the unique two-spout wall and the monitors. The smaller area holds a mouse not under observation. The wall separating the two cages has holes in the bottom so that the mice can hear, see, and smell each other, as per the animal regulations followed by the Hengen Lab.

Table 1 lists the design components of the RAT Cage.

Table 1: Design components and a brief description of each. A full parts list with purchasing links is in Appendix B. Drawings with dimensions for CAD pieces are in Appendices A and C.

Design Component	Description
Cage	12" x 7" x 20" Cage that houses two mice. One is being trained in a visual discrimination task. The other is there to provide company for the first mouse in accordance with the Hengen lab animal regulations.
Solenoid valve (2)	Valves that can open and close to allow water through to connected spouts.
Arduino Mega	Arduino that controls the capacitive sensor and the opening/closing of the solenoid valves.
Arduino UNO	Arduino that controls the tone played when the mouse correctly chooses the water and when training has begun for the day.
Raspberry Pi	Controls the monitor and all of the arduinos. Has the central code that turns on each arduino during relevant times during training.
Micro SD card	Holds the Raspberry Pi operating system Raspian. Is plugged into the Raspberry Pi
Monitor	Displays the visual stimuli for the mouse.
Tubing	For the water to go from the reservoir through the solenoid valves and into the spout.
Tubing connectors (4)	Connects the solenoid valve to the tubing without leakage of water.
Spout (2)	Connects to the water and the capacitive sensor. When the mouse licks a spout, it will trigger water delivery to the spout.
Speaker (2)	Plays the tone when triggered by the arduino.
Capacitive sensor	Senses changes in capacitance, connected to each spout.
Wall power supply	Supplies power to the driving motor.
Driving motor (2)	Supplies power to the solenoid valve and connects to the arduino to signal the solenoid to open and close when needed.
Bread board	Used to wire the driving motors, set up a ground and power port. Soldered.
Valve Cradle	Holds the valves in place on top of the cage.

Verification Results

1. Screens

The verification plan for the screens was initially to test that two independent screens could display different images at the same time from the same controller. When the design changed to one large monitor rather than two small screens, the plan was modified to test that two random differing images could appear on the left and right of the large monitor at the same time. Once this was achieved, the group moved on to the second step in the screen validation. In the second step of the screen verification, the process of displaying two random images from a set of images on the left and right side of the screen was repeated at five minute intervals to ensure that images from the set could be displayed at specified times.

The group met both verification goals by using a test set of images and displaying them next to each other on a laptop and on the monitor that will be used in the system.

2. Capacitive Touch Sensor

The verification plan for the capacitive touch sensor was to connect the sensor to a spout and have a group member touch it and observe the response from the system. The group tested this by having one member monitoring the computer readout of the capacitance at the sensor, and one group member touching the spout. When the group member was touching the spout, the computer readout indicated a significant change in capacitance. This was tested on four different days in four different locations. Each test had the group member touching the spout ten times. In all the tests, the readout indicated a significant change in capacitance. Thus, the touch sensor was verified to be working.

3. Solenoid Valve

The verification plan for the solenoid valve was as follows. First, the valve should open and water should flow through when 12V are applied to the system. This was verified by setting

up an Arduino circuit and applying 12V to the solenoid when it had the proper tube connectors in place and a reservoir of water. When the 12V were applied, water flowed through from the reservoir to a graduated cylinder below, verifying that water could flow through the device controlled by Arduino code and circuit.

Second, the valve should open for a period of time as to let a specific amount of water through each time. According to Long et al. this amount of water should be 10-12 μ L for water deprivation studies. However, that amount of water is difficult to deliver when gravity is the driving force moving the water. In Long et al., they were delivering this amount of water by hand with a syringe, which is not possible in this setup. The client was more concerned with the automation of the project, so the amount of water delivered per opening of the solenoid valve was changed from the verification and validation report from 10-12 μ L to 120-130 μ L.

In order to test the valve to see how much water went through the valve each time it opened, the group started with opening the valve for 0.5 seconds 25 times, and measuring the amount of water at the end in a 10mL graduated cylinder. After three cycles of the valve opening for 0.5 seconds, the group stopped the test because 6 mL of water had already come out. The next trial was run at 0.1 seconds of the valve opening, but the same amount of water came out in three cycles. The group determined that 0.5 seconds was the quickest the valve could open and close.

In order to decrease the amount of water flowing through the valve while keeping the time at the minimum of 0.5 seconds being open, the top of the Lead-Free Brass Hose Barb



Figure 1: Both solenoid valves side by side in the valve cradle. The valve on the right has the tinfoil covering with a 1mm hole in it (hole not shown)

Adaptor was covered in tinfoil, and a small hole was poked through with a pair of tweezers (diameter = 2mm) . Running the simulation again with this setup, after 25 cycles, 10 mL of water had come through, meaning that for each cycle of opening and closing the valve, 400 μ L was delivered. This was repeated twice more, and the average was taken for all three trials.

This process was repeated, with the hole in the tinfoil decreasing to further decrease the amount of water delivered with each cycle. The amount of water from each diameter is shown in Table 2.

Table 2: Diameter of the hole in the tinfoil above the solenoid valve with the average amount of water delivered per valve opening.

Diameter (mm)	Number of trials	Water delivered per opening (mL)
2	3	0.403
1.75	4	0.332
1.5	3	0.257
1.25	3	0.189
1	3	0.131
0.75	4	Inconclusive, possibly 0.05
0.5	4	Inconclusive, possibly 0.03

In all of the trials except the last 8 (the ones with 0.75 and 0.5 mm diameters), every time the valve opened, water would drop into the graduated cylinder. During the trials with the 0.75mm and 0.5mm diameter holes, multiple cycles of opening and closing would occur without water being delivered to the graduated cylinder. In the 0.75 mm trial, every 3-4 cycles, about 0.13mL of water would drop into the graduated cylinder. In the 0.5mm trial, every 4-5 cycles about 0.13 mL of water would drop into the graduated cylinder. These numbers were generally consistent across all 8 trials. From these data, it was determined that with this current setup, the

1mm diameter opening with the average delivery of 0.131 mL of water per open/close cycle would be used in the final setup.

4. Lights

The verification plan for the lights was to set up an Arduino circuit and code that would turn on and off a string of LED lights at 2 second intervals. The group set up the circuit and wrote the code, connected the LED lights and verified that at two second intervals, the LED lights flashed on and off. Additionally the group verified that the lights could be turned on for specified time intervals. The group tested and verified 2, 5, and 10 second long intervals for the Arduino to turn the lights on.

5. Speaker

The verification plan for the speakers was to set up an Arduino circuit and code that would play a 1,000 Hz tone for 1 second. The group set up the circuit and wrote the code, next hooked it up to the speakers, and verified that a 1,000 Hz tone played for 1 second.

6. Cage Design

The verification plan for the cage was as follows. First, the group had to verify that the dimensions of the proposed cage would fit in the available space. To do that, the base of the cage was sketched to scale on three sheets of 8.5x11 inch paper and laid out on the racks in the Hengen lab where the rodent cages are. The base drawing fit well within the area available, so the cage was laser cut and constructed.

Second, the group had to ensure that a rodent living in the cage could not get out. The plan was to put in bedding, food, and water and allow the rodent to live in the space for 24 hours

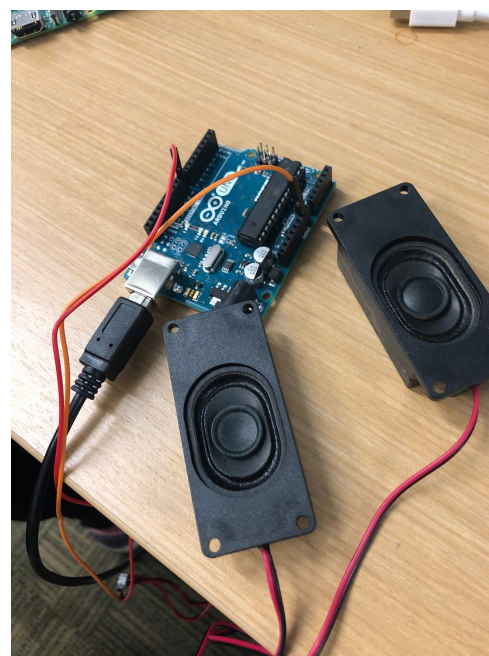


Figure 2: Speaker/Arduino setup

under surveillance to make sure they could not access the antechamber. However, when the two small screens were changed to one larger monitor, the antechamber was scrapped. Since there was no more antechamber, the group just had to verify that the rodent (in this case a mouse) was unable to get out of the cage. Since the only change to the design from the cages already in the Hengen lab was the addition of a second hole for a second water spout, it was determined by the client that there was no need to do this verification as the mice already live in cages with one spout hole and are unable to get out.

Validation Results

1. Screens

The plan to validate the screen was to have the raspberry pi send an experimental image to the screen and ensure that the system is capable of recognizing which image is “correct”, or the desired image for the mouse to select. The raspberry pi was able to project two images, provided by the client, at a specified time and print out which image was the “correct” image. By confirming both of these aspects of the screens, they could be validated as functioning parts of this overall system.

2. Capacitive Touch Sensor

To validate that the capacitive touch sensor was working in the system, the group set up a mock test, where one spout was deemed “correct” and the other was deemed “incorrect”. Next, a group member touched the “incorrect spout”. The system registered the lick in a print readout on the computer screen, and a timeout of 40 seconds was initiated. During this timeout, no licks were recorded, even though they were registered by the sensor. After the 40 second timeout, a group member touched the “correct” spout. The system registered the lick in a print readout on the computer and the group heard the corresponding solenoid valve click open. 20 seconds of timeout followed to allow the mouse to drink the water that would be available.

Next the group disconnected the Arduino controller from the computer and connected it to the Raspberry Pi. The same protocol was initiated, with one spout being designated as “correct” and the other as “incorrect”. A group member touched the incorrect spout and a 40 second timeout was initiated. During the timeout, when either of the spouts were touched, nothing happened. After 40 seconds, the “correct” spout was touched, and the corresponding valve clicked open and a 20 second timeout was initiated. Note that when the “incorrect” spout was touched, the corresponding valve remained closed.

3. Solenoid Valve

To validate that the solenoid valves were working in the system, first, the group set up a test to ensure that each spout was only connected to one valve. To determine this, the Arduino Mega was set up with two solenoid valves and the capacitive sensor. Simple code was written so that the signal from sensor 5 on the MPR 121 capacitive sensor would correspond to the valve at digital pin 10 on the Arduino Mega and sensor 6 to digital pin 8. Next, a group member touched the spout attached to sensor 5, and the group heard the valve attached to pin 10 open (a loud click occurs when the valve is opened or closed). Next, the same group member touched the other spout, connected to sensor 6, and heard the valve attached to pin 8 open.

Next, the group integrated this Arduino Mega to the rest of the system. While validating the capacitive sensor, the group was also able to validate that the spout opened when the “correct” spout was touched and did not open when the “incorrect” spout was touched. The account of this validation can be seen under the Capacitive Sensor heading.

4. Lights

The group did not validate or integrate the LED lights into the system because the client decided he did not want the additional lights stimulus in his initial experiment. Thus, the lights

were not integrated. However, the code and the equipment are set up, so when the client would like to integrate the lights, he can do so with ease.

5. Speaker

The speaker was validated by testing both that a sound can be made at the designated testing time and that a sound can be made when the correct water spout is chosen. First the code was run so that the noise would sound at 5 specific times, each 1 minute apart. The speaker was integrated correctly into the system so that it successfully passed this test. Next, to mimic the mouse licking the spout, a group member touched the “correct” spout and the speaker made a distinct noise as positive reinforcement. By the speaker demonstrating accurate timing and integration into the system’s knowledge of the “correct” spout, the speakers were validated as properly integrated.

6. Cage Design

The cage is designed to allow for easy integration of all of the design parts while ensuring that the training protocol can be completed. The two key steps to validate this design were first to assemble all of the components and second to ensure that the mouse can properly view the images from the cage. When the parts

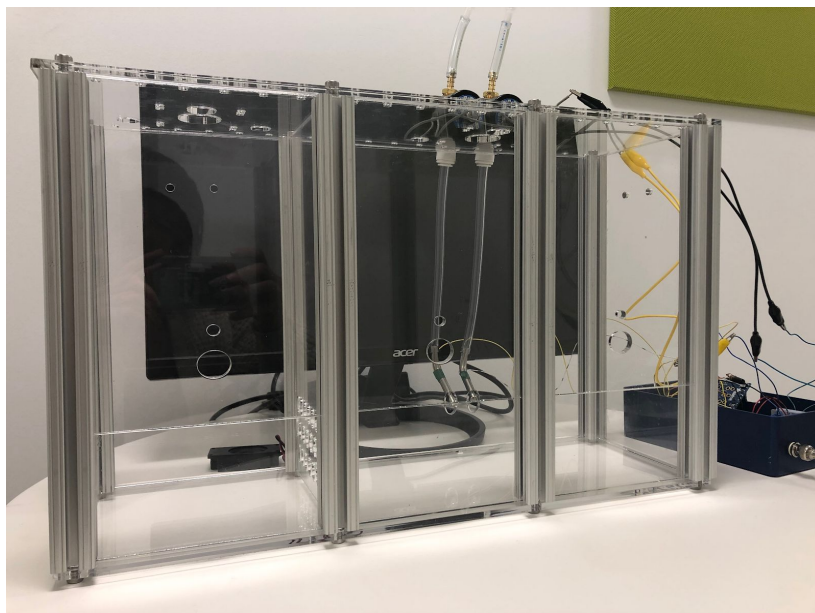


Figure 3: Overall Cage Setup

are assembled, the cage can effectively project an image to the mouse, measure the change in capacitance at the spouts, and provide water through a motorized valve. The water reservoirs are hung above the cage, out of view of Figure 3, and the water can easily flow down to the valves valves which are elevated on the top of the cage

using a 3D printed cradle. The cradles stabilize the valves while keeping them out of view of the mouse to prevent interference with the visual stimuli. Similarly, the circuit box is placed to the side of the cage, allowing the wires to easily reach the water spouts without blocking the mouse's vision. When assembled, the cage also has a designated space to house a second mouse which allows this cage to function as both the testing and housing facilities for the mouse. This was essential because lab protocol dictates mandatory dual-housing.

After validating that the cage could be effectively assembled for the the experimental protocol, it was essential for the group validate that the mouse has a clear view of the visual stimulus with minimal obstruction.

To meet the needs of the client, the visual stimulus should occupy the mouse's entire binocular field of view and extend into their monocular view on both sides. As indicated in the

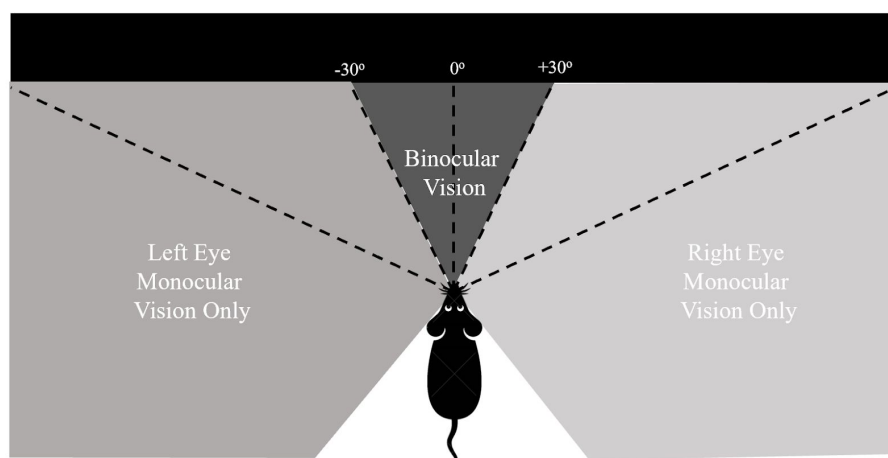


Figure 4: Mouse Field of View

figure below, a mouse's binocular field of view is exclusively the 60 degrees immediately in front of them. Their remaining field of view is monocular with respect to the eye on that side.

The center of the screen is located exactly 20 cm in front of the water spouts. The left and right edges of the screen are therefore each 53° from the center of the mouse's vision, providing a significant image in the monocular field of view. A screen closer to the mouse would allow for the image to occupy a greater portion of the mouse's visual field however this would

sacrifice the mouse's ability to clearly focus on the images. In similar visual discrimination tasks, a mouse's has been head-fixed and presented with images on a screen. In those cases, the screen was located 14 to 18 cm away from the mouse (4,5). In visual tasks with free mobilization however, images have been displayed as far as 105 cm away from the mouse (6). Since the protocol associated with this design does not require head fixation but does promote stable head position through consistent water distribution, 20 cm is a valid distance for the screen to accommodate both the needs for focus and monocular view.

Lastly, the assembled cage has metals support beams and it was essential for these beams not to interfere with the mouse's view of the screen. The beams in the testing cage are twice as far apart as they are in a typical housing cage, allowing for the vision of the screen to be completely uninterrupted by the beams. Since the beams are located approximately 90 degrees from the center of the mouse's vision and the edges of the screen are 53 degrees from center, the cage design can be validated as permitting a clear view of the screen.

DesignSafe Analysis

When considering the safety of this design and going through the DesignSafe Analysis, the group considered the two main users of this system - the client, Sam Brunwasser running the experiment (the technician) and the rodent living in the cage (the patient). The main concerns in this system are all the electrical components, and ways wires could get frayed, or how there could be problems with grounding. These concerns were addressed during the process of building and designing the prototype and have been outlined in the analysis in Figure 5.

Item	User	Task	Hazard	Cause/Failure Mode	Severity	Exposure	Probability	Risk Level	Method of Risk Reduction/Mitigation	Severity	Exposure	Probability	Risk Level	Person Responsible	Date	Comments
Cage	Technician	Train rodent	energized equipment	Wires fraying due to general wear and tear, wires disconnecting from ground due to careless movement of cage	Serious	Occasional	Possible	Medium	Put the bulk of the wiring in a box, use connectors for the wires that have to go outside the box and connect to the cage. That way, if someone moves the cage, the wires will simply unplug rather than be pulled or yanked out.	Serious	Occasional	Unlikely	Low	Elle Degen	4/5/19	Professor Klaesner assisted with this issue. Note: There is no option for "subject" which in our case is a mouse, so "patient" was used. The risk of a power surge is very low, and there are surge protectors on all the outlets in the lab, so no mitigation needed. Additionally, the risk of harm from a shorting capacitor is only serious if the capacitor has a large capacitance, which the MFR 121 does not.
Spot	Patient	Lick Spot.	electrocution due to capacitive sensor	Power surge causing steep voltage increase, or capacitor shorting.	Serious	Frequent	Unlikely	Low		Serious	Frequent	Unlikely	Low			
Valves	Technician	Train rodent, cleaning cage	valves fall from cage and break	Technician bumping into cage, movement of cage during cleaning	Slight	None	Practicable	High	Design and 3D print cradle to hold the solenoid valves securely in place	Slight	None	Unlikely	Low	Zoe Orenstein	4/13/19	Professor Klaesner assisted with this issue. Technicians monitor the water level changes for all the rodents in the lab twice a day-- if a rodent is not drinking, it will be noticed within the day. Additionally, solenoid valves are very stable when they are in a working circuit. The issues that could cause failure of the solenoid valve have been addressed already, and those that are out of the control of the user are unlikely to occur in this situation.
Cage	Patient	Train rodent	electric shock	Lack of grounding (earthing or neutral) - the cage is on a wire rack	Serious	Frequent	Probable	High	Connect a ground wire from the circuit to one of the metal support	Serious	Frequent	Unlikely	Low	Elle Degen	4/5/19	
Valves	Patient	Drinking water	Valve breaks	Power failure, residual coil power, short circuit	Serious	Frequent	Negligible	Low		Serious	Frequent	Negligible	Low			

Figure 5: DesignSafe analysis chart

Conclusion

After thorough testing and many iterations of the product design, a cohesive automated cage has been designed that meets the demands of the client. Throughout the process of designing of the cage, many discussions were held between the group and the client that resulted in massive alterations to the design specifications and even the core concept of the cage. Through these gradual changes, the client's vision of the product has significantly changed therefore one of the greatest successes of the cage is its inherent modularity. This cage is designed specifically to serve our client with the potential of aiding the greater neuroscience research community. Each of these researchers will be looking to perform studies that have slightly different demands and our cage can satisfy a range of experimental protocols.

Looking ahead, the client is planning a variety of studies that will use the cage which are enhanced by the modular aspects. The aspects that can be most easily adjusted are the number of spouts, the number of visual stimuli, light reinforcement, and sound reinforcement. The code is currently written to accommodate two spouts for a forced-decision experiment however, activating only one of these spouts can easily convert the cage to a go/no-go experiment. Similarly, the program allows for two images to be displayed on the screen simultaneously but this can easily be changed to only project a single image at a time. The reinforcement through the lights and sound have been prepared however there is a large library of stimuli that can be accessed through these components allowing for stimulation more complex than reinforcement. The potential uses for this cage are drastically increased by the modularity of the components which is critical in in the ever-changing research environment.

While the current design satisfies the demands of the client, there are changes that can be made to make the system function more accurately and efficiently. Some of the changes that could be made to the water delivery system include a smaller valve, a smaller water spout,

and a motorized water pump. When the water delivery was originally designed, it was intended to be used by a rat however during the design process, the animal was changed to a mouse. A mouse drinks less water than a rat therefore it would be ideal for less water to be released with each opening of the valve. A smaller valve and smaller spout would ensure a better result for mouse experiments. The motorized water pump would allow for improved visibility of the screens by allowing the water tubing to be located below field of view of the mouse. The current tubing requires gravity to help water flow from the reservoir to the spout, causing the tubing to fall directly in front of the mouse and obstruct part of their view of the screen. If this water was pumped from below, then the tubing would not pass in front of the mouse and their view of the screen would be improved.

Another significant change that could improve the design of the RAT cage would be the integration of a curved monitor in place of our flat monitor. This would project the image both in front of the mouse and on their sides, allowing the image to encompass a greater portion of their field of view. By showing visual stimulus on the sides of the rat, there is potential to stimulate different parts of their brain and expand the scope of the experiment.

Lastly, the user interface could be improved to become more user-friendly. Fortunately the client is familiar with Python, enabling him to easily work with the program that we have designed. If this cage is eventually expanded to other labs, not all researchers may be familiar with Python so it would be beneficial to make the user-interface more accessible to those not familiar with coding.

Although there is potential to share this design with more labs, there is not potential to obtain Intellectual Property rights to the design. Similar concepts have been discussed in scientific publications (4,5), disqualifying it from being considered IP.

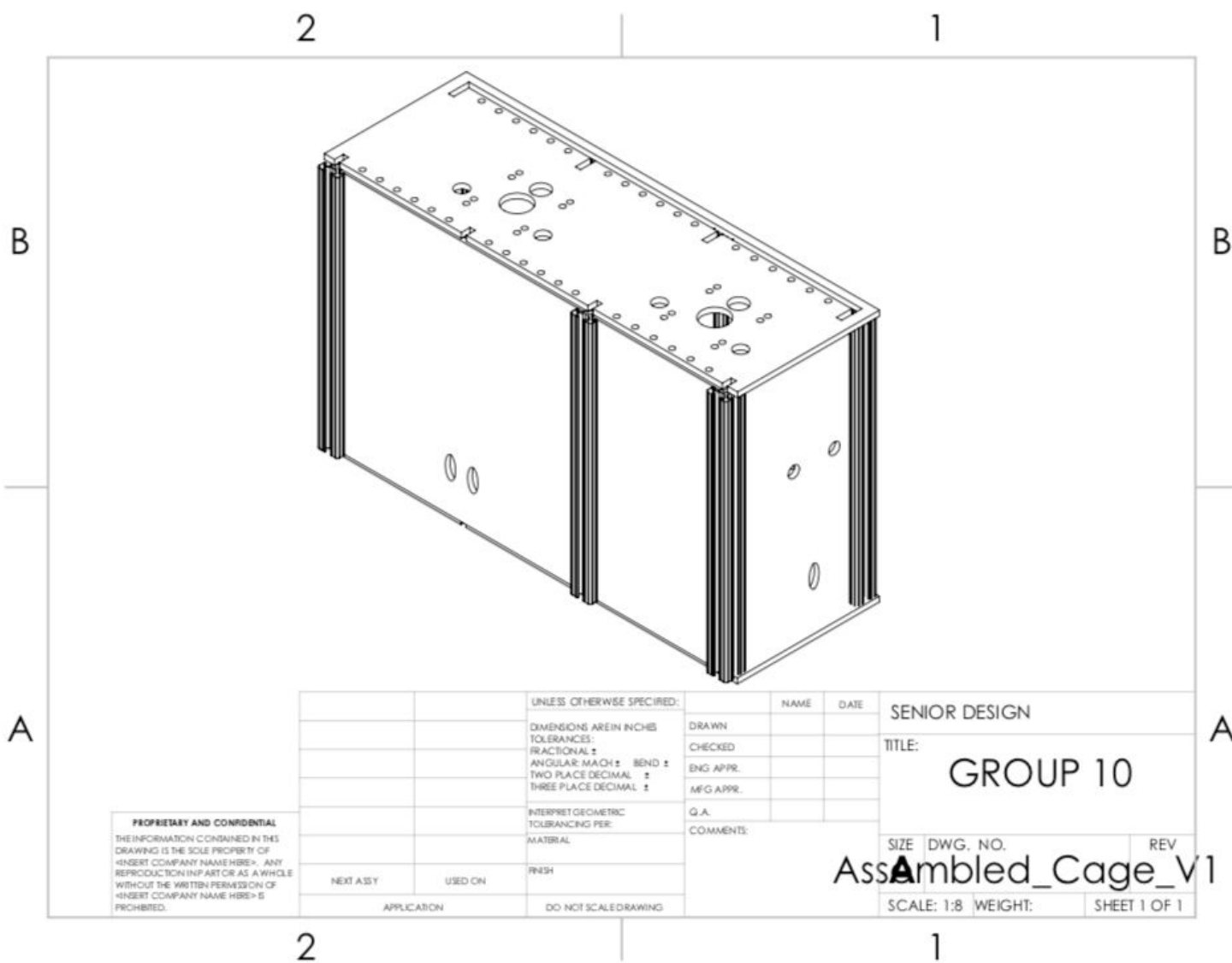
Appendix

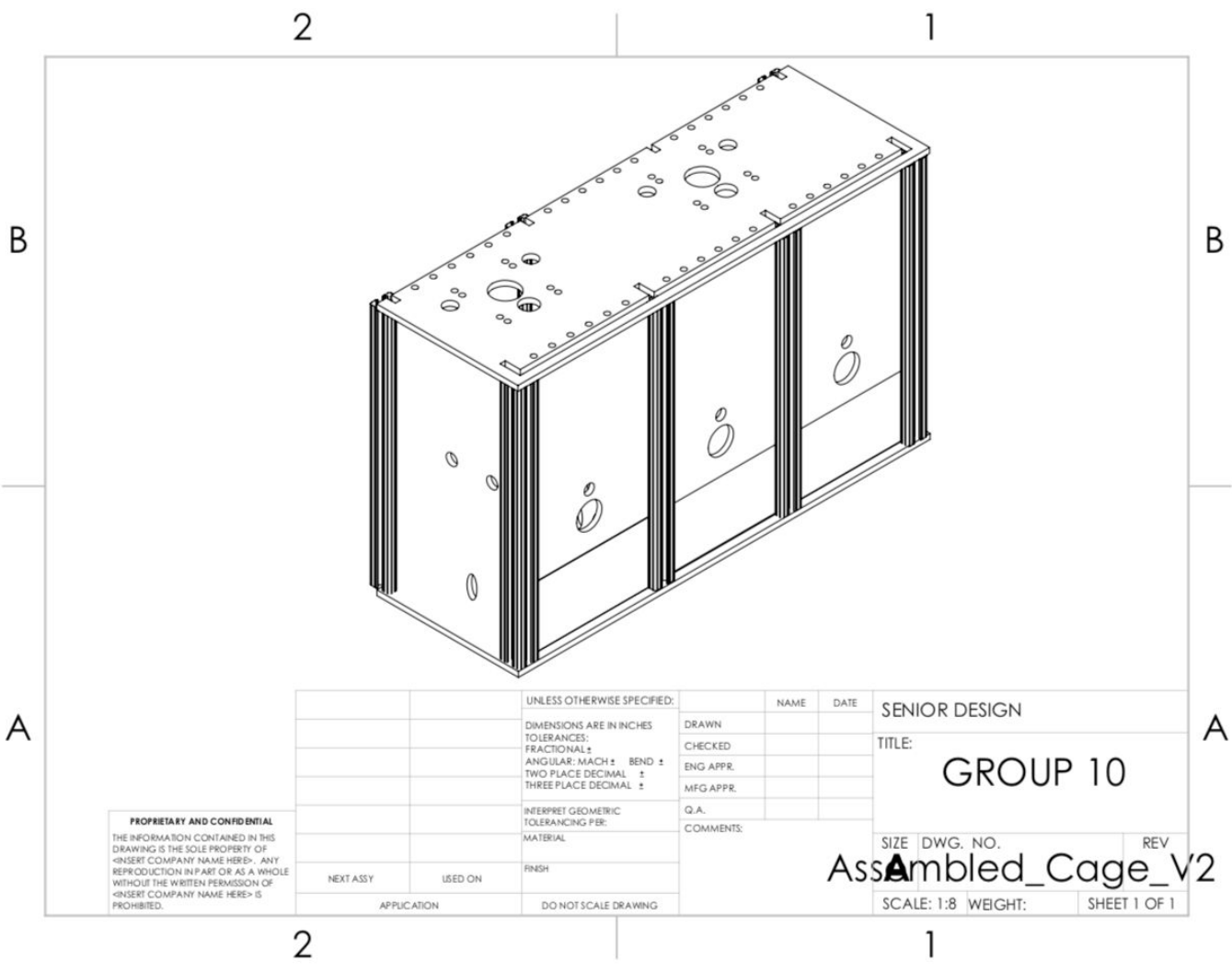
A. Cage

Drawings

The walls of the cage were laser cut from sheets of extruded acrylic in the Hengen Lab. They were assembled together with 10 series 80/20 support beams and hex bolts. As specified on each drawing, all dimensions are in inches.

This design includes one of every drawing shown in this appendix aside from the assemblies and Cage Base Open 2 and Cage Top Open 2, which are both in the cage twice





PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

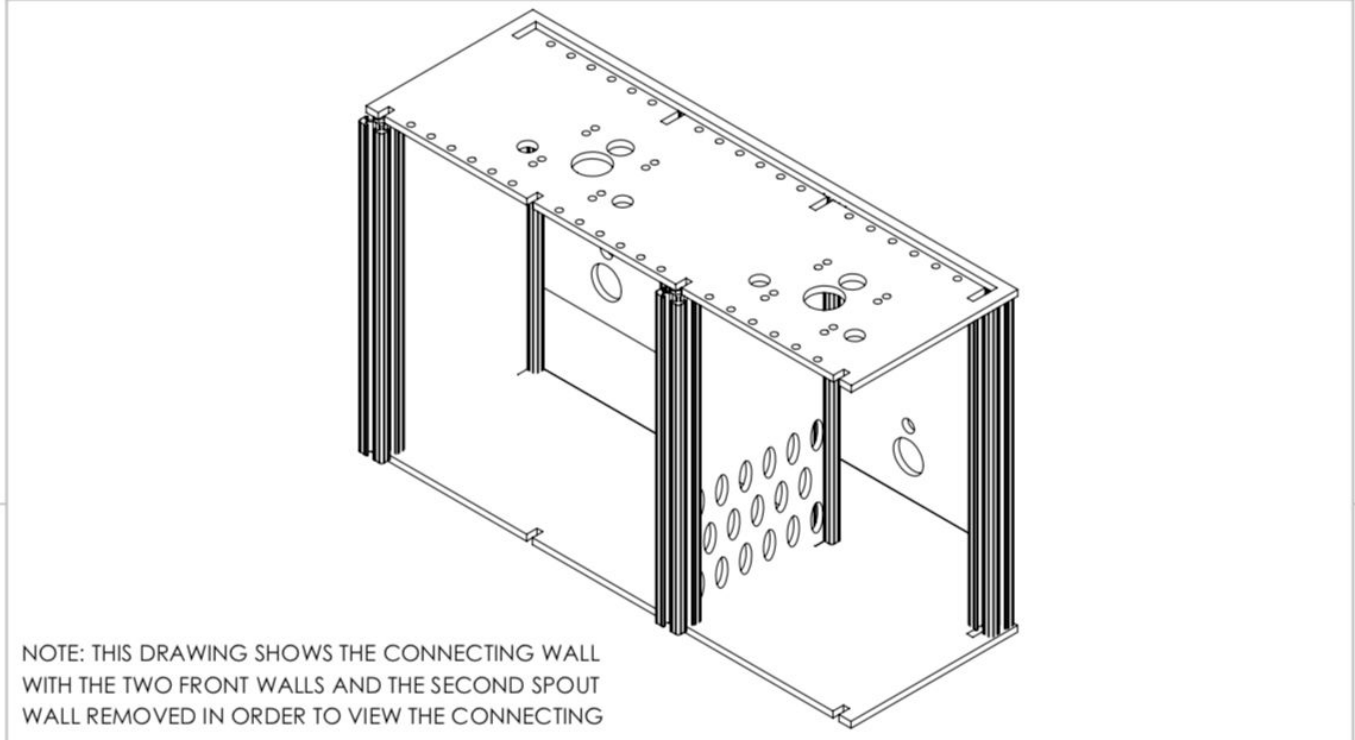
		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN
		DIMENSIONS ARE IN INCHES	DRAWN			TITLE:
		TOLERANCES:	CHECKED			GROUP 10
		FRACTIONAL ±	ENG APPR.			
		ANGULAR: MACH ± BEND ±	MFG APPR.			
		TWO PLACE DECIMAL ±	Q.A.			
		THREE PLACE DECIMAL ±	COMMENTS:			
		INTERPRET GEOMETRIC TOLERANCING PER:				SIZE DWG. NO. REV
		MATERIAL				AssAsembled_Cage_V2
		FINISH				SCALE: 1:8 WEIGHT: SHEET 1 OF 1
NEXT ASSY	USED ON	DO NOT SCALE DRAWING				
	APPLICATION					

2

1

B

B



NOTE: THIS DRAWING SHOWS THE CONNECTING WALL WITH THE TWO FRONT WALLS AND THE SECOND SPOUT WALL REMOVED IN ORDER TO VIEW THE CONNECTING

A

A

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

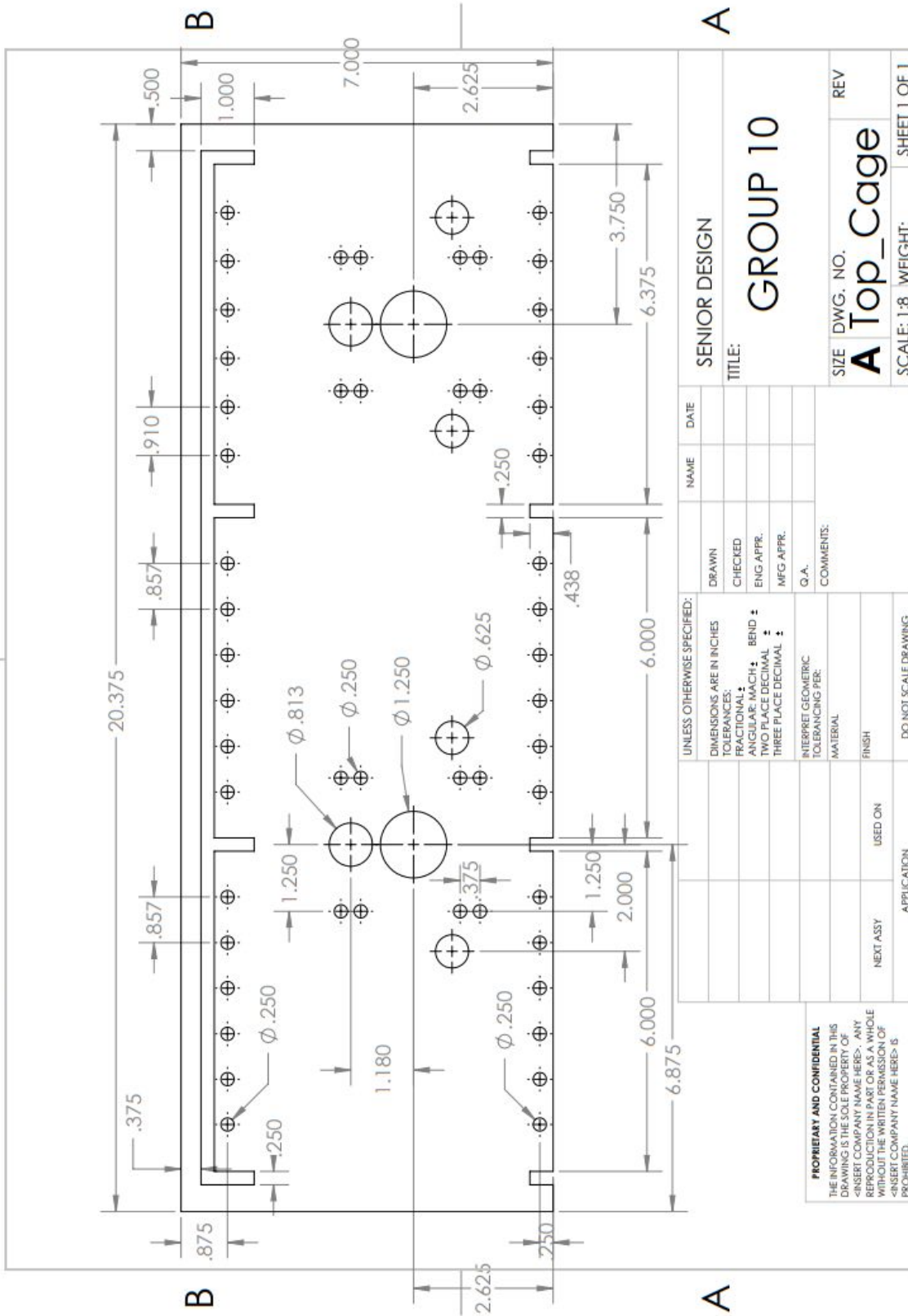
		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN	
		DIMENSIONS ARE IN INCHES		DRAWN		TITLE:	
		TOLERANCES:		CHECKED		GROUP 10	
		FRACTIONAL: ±		ENG APPR.		SIZE DWG. NO. REV	
		ANGULAR: MACH: ± BEND: ±		MFG APPR.		Partial_Cage_View	
		TWO PLACE DECIMAL ±		Q.A.		SCALE: 1:8 WEIGHT: SHEET 1 OF 1	
		THREE PLACE DECIMAL ±		COMMENTS:			
		INTERPRET GEOMETRIC TOLERANCING PER:					
		MATERIAL					
		FINISH					
NEXT ASSY	USED ON						
APPLICATION		DO NOT SCALE DRAWING					

2

1

2

1



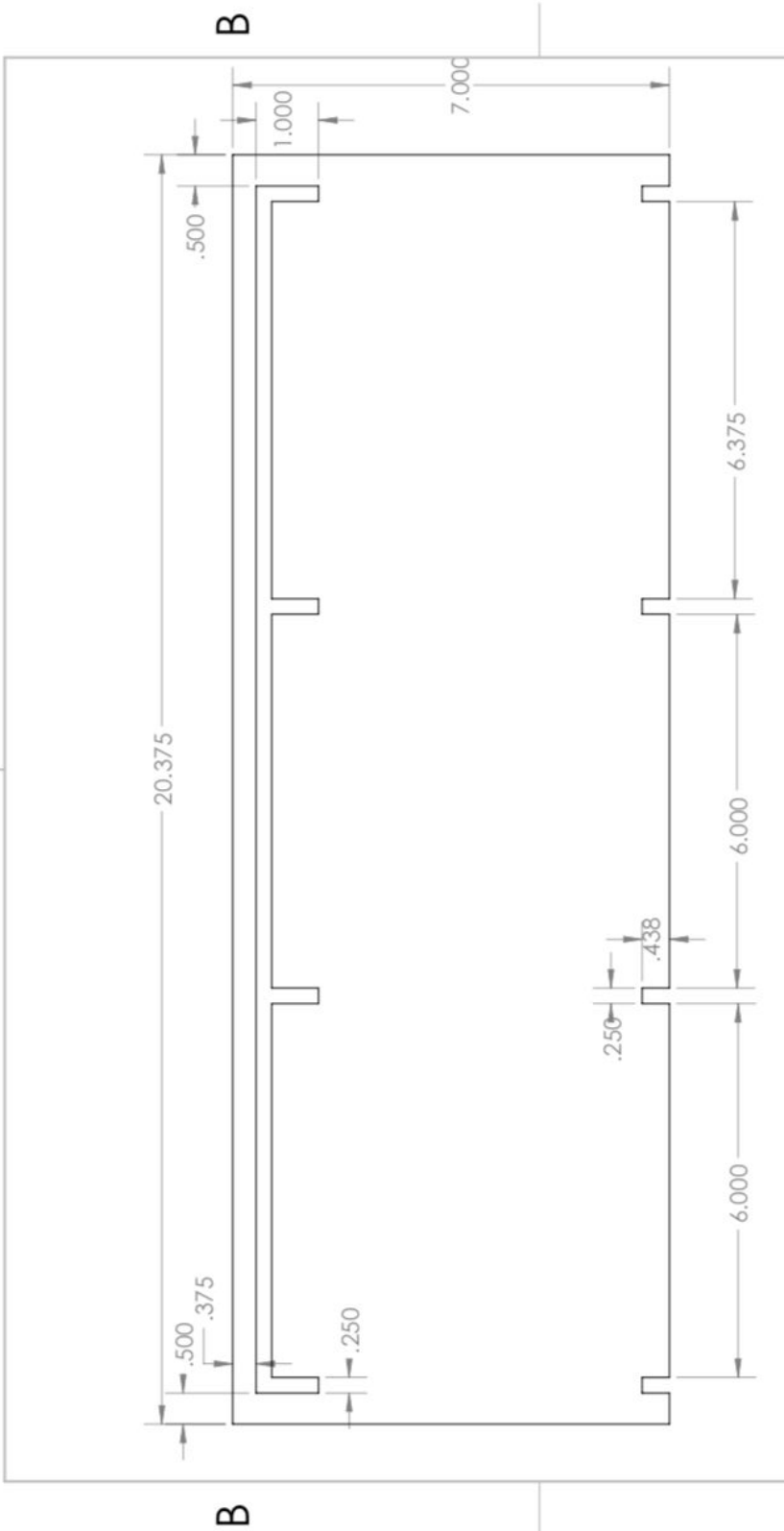
2

1

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL: ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±		DRAWN	NAME	DATE
		CHECKED		
		ENG APPR.		
		MFG APPR.		
		Q.A.		
		COMMENTS:		
INTERPRET GEOMETRIC TOLERANCING PER:				
MATERIAL:				
FINISH:				
NEXT ASSY		USED ON		
APPLICATION		DO NOT SCALE DRAWING		
<p>PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.</p>				
<p>SENIOR DESIGN</p> <p>TITLE: GROUP 10</p>				
SIZE	DWG. NO.	REV		
A	Top_Cage			
SCALE: 1:8	WEIGHT:	SHEET 1 OF 1		

1

2



A

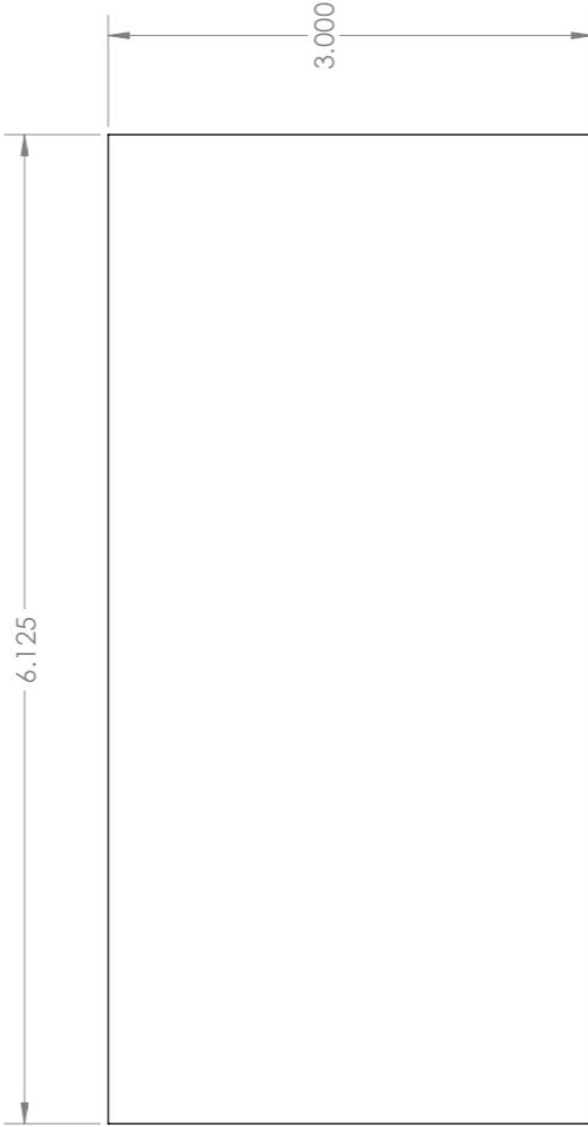
A

UNLESS OTHERWISE SPECIFIED:		DRAWN		NAME		DATE		SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		CHECKED						TITLE:	
TOLERANCES:		ENG APPR.						GROUP 10	
FRACTIONAL:		MFG APPR.						SIZE DWG. NO.	
ANGULAR: MACH :		G.A.						A	
BEND :		COMMENTS:						REV	
TWO PLACE DECIMAL :								A	
THREE PLACE DECIMAL :								Cage_Bottom	
INTERPRET GEOMETRIC TOLERANCING PER:								SCALE: 1:8	
MATERIAL								WEIGHT:	
FINISH								SHEET 1 OF 1	
NEXT ASSY		USED ON		APPLICATION		DO NOT SCALE DRAWING		1	
APPLICATION								2	

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

1

2



B

B

A

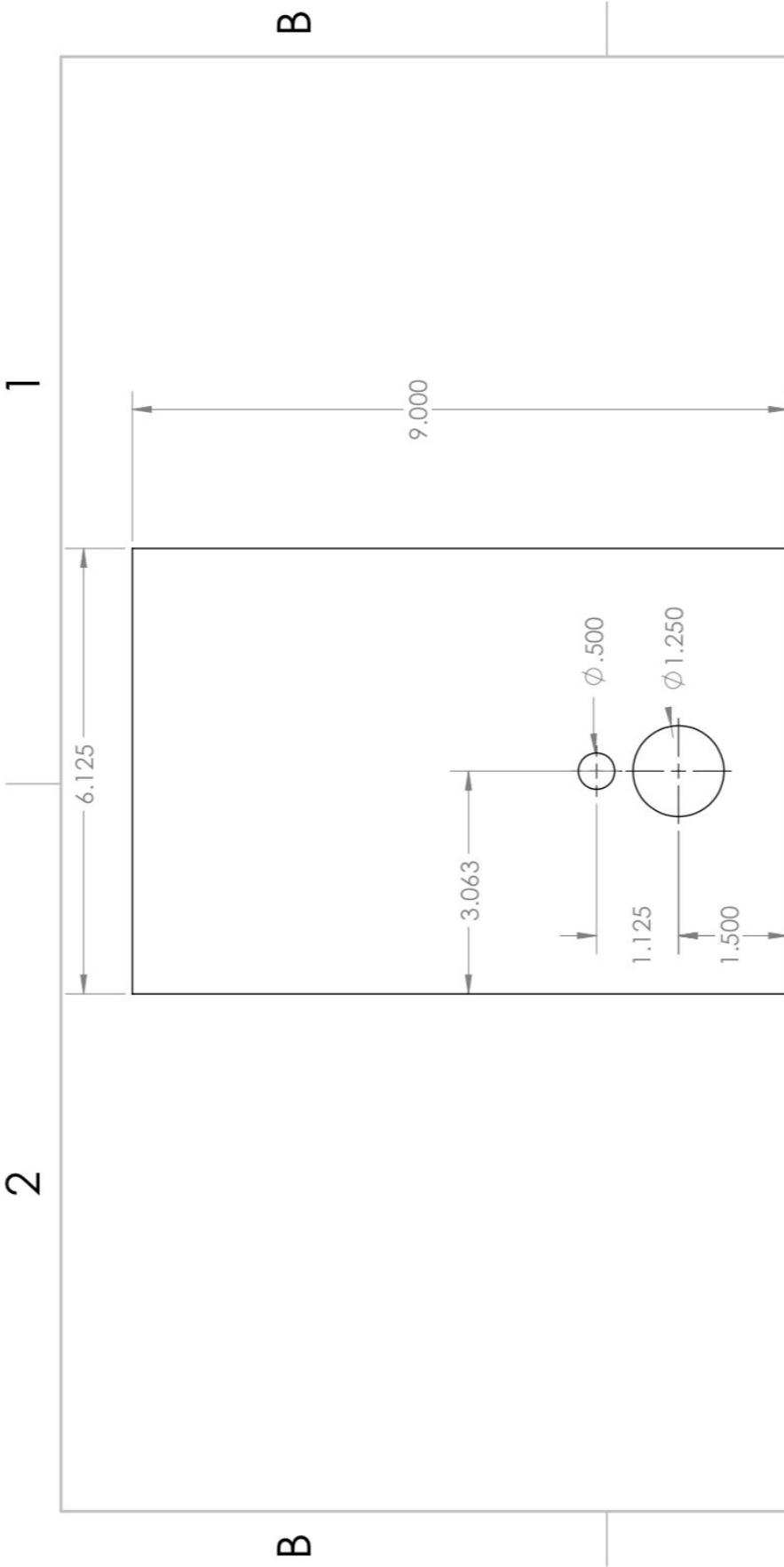
A

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		DRAWN		TITLE:	
TOLERANCES:		CHECKED		GROUP 10	
FRACTIONAL: ±		ENG APPR.		SIZE	DWG. NO.
ANGULAR: MACH: ± BEND: ±		MFG APPR.		Cage_Base_Open_1	
TWO PLACE DECIMAL ±		Q.A.		SCALE: 1:2	WEIGHT:
THREE PLACE DECIMAL ±		COMMENTS:		SHEET 1 OF 1	
INTERPRET GEOMETRIC TOLERANCING PER:		REV			
MATERIAL		Cage_Base_Open_1			
FINISH		NEXT ASSY			
DO NOT SCALE DRAWING		USED ON			
APPLICATION		APPLICATION			

1

2

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.



UNLESS OTHERWISE SPECIFIED:		DRAWN		NAME		DATE		SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		CHECKED						TITLE:	
TOLERANCES:		ENG APPR.						GROUP 10	
FRACTIONAL ±		MFG APPR.						SIZE DWG. NO. REV	
ANGULAR: MACH ± BEND ±		Q.A.						Cage_Top_Open_1	
TWO PLACE DECIMAL ±		COMMENTS:						SCALE: 1:2 WEIGHT: SHEET 1 OF 1	
THREE PLACE DECIMAL ±		INTERPRET GEOMETRIC TOLERANCING PER:							
		MATERIAL							
		FINISH							
		USED ON							
		APPLICATION							
		DO NOT SCALE DRAWING							

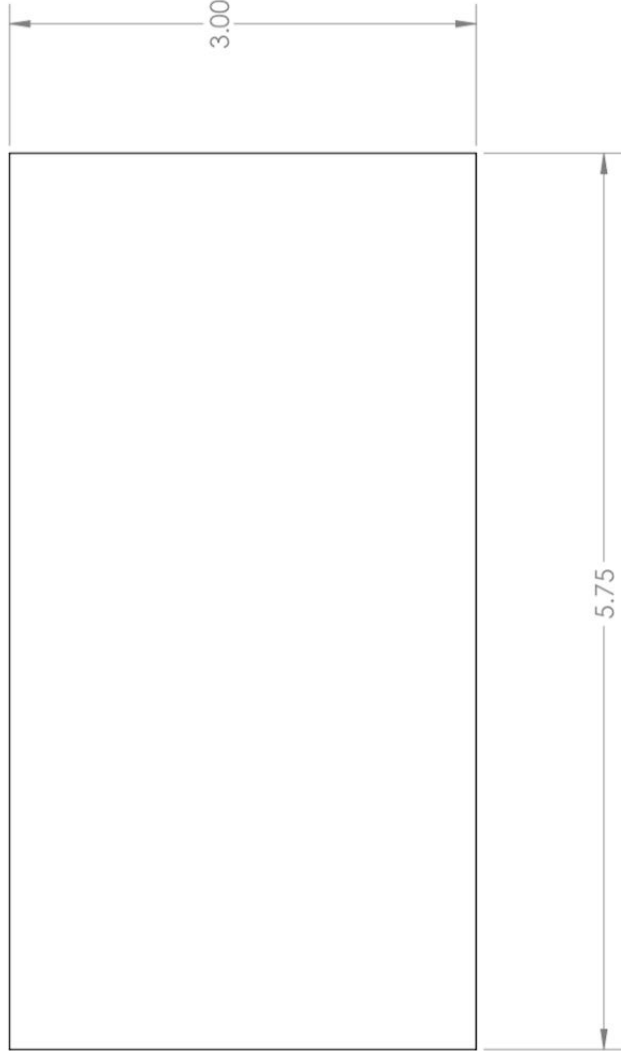
PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

1

2

B

B



A

A

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		DRAWN		TITLE:	
TOLERANCES:		CHECKED		GROUP 10	
FRACTIONAL: ±		ENG APPR.		SIZE DWG. NO.	
ANGULAR: MACH ± BEND ±		MFG APPR.		Cage_Base_Open_2	
TWO PLACE DECIMAL ±		Q.A.		REV	
THREE PLACE DECIMAL ±		COMMENTS:		SCALE: 1:1 WEIGHT: SHEET 1 OF 1	
INTERPRET GEOMETRIC TOLERANCING PER:					
MATERIAL					
FINISH					
DO NOT SCALE DRAWING					
NEXT ASSY	USED ON				
APPLICATION					

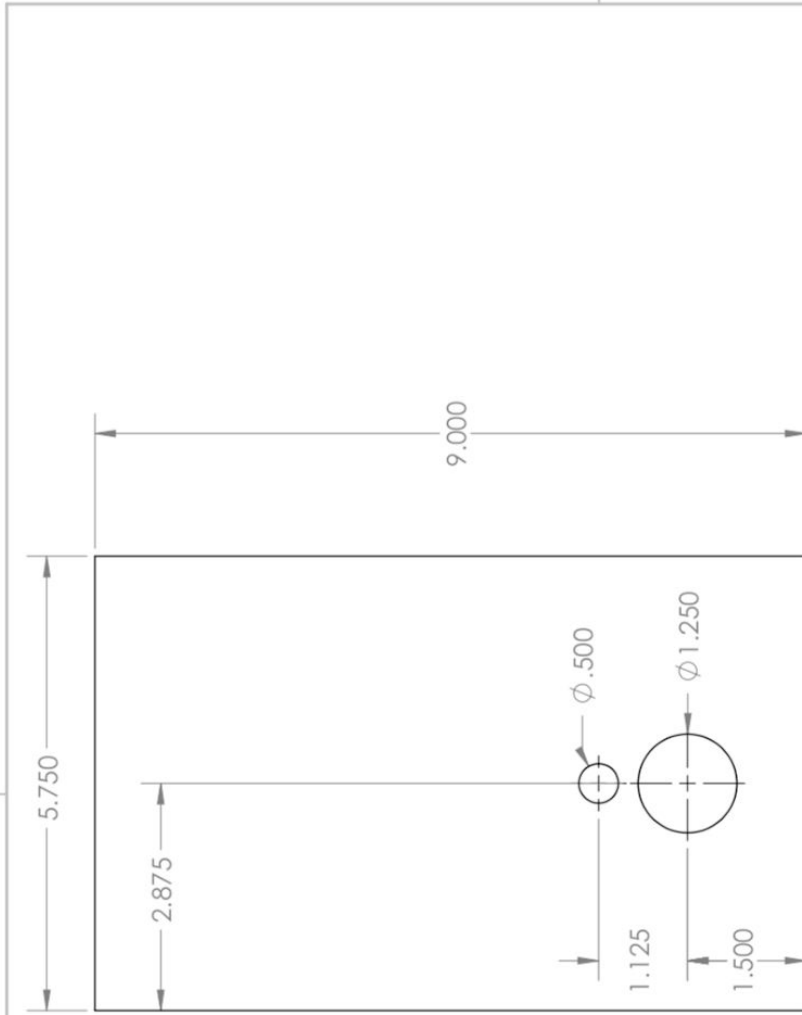
PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

1

2

2

1



B

B

A

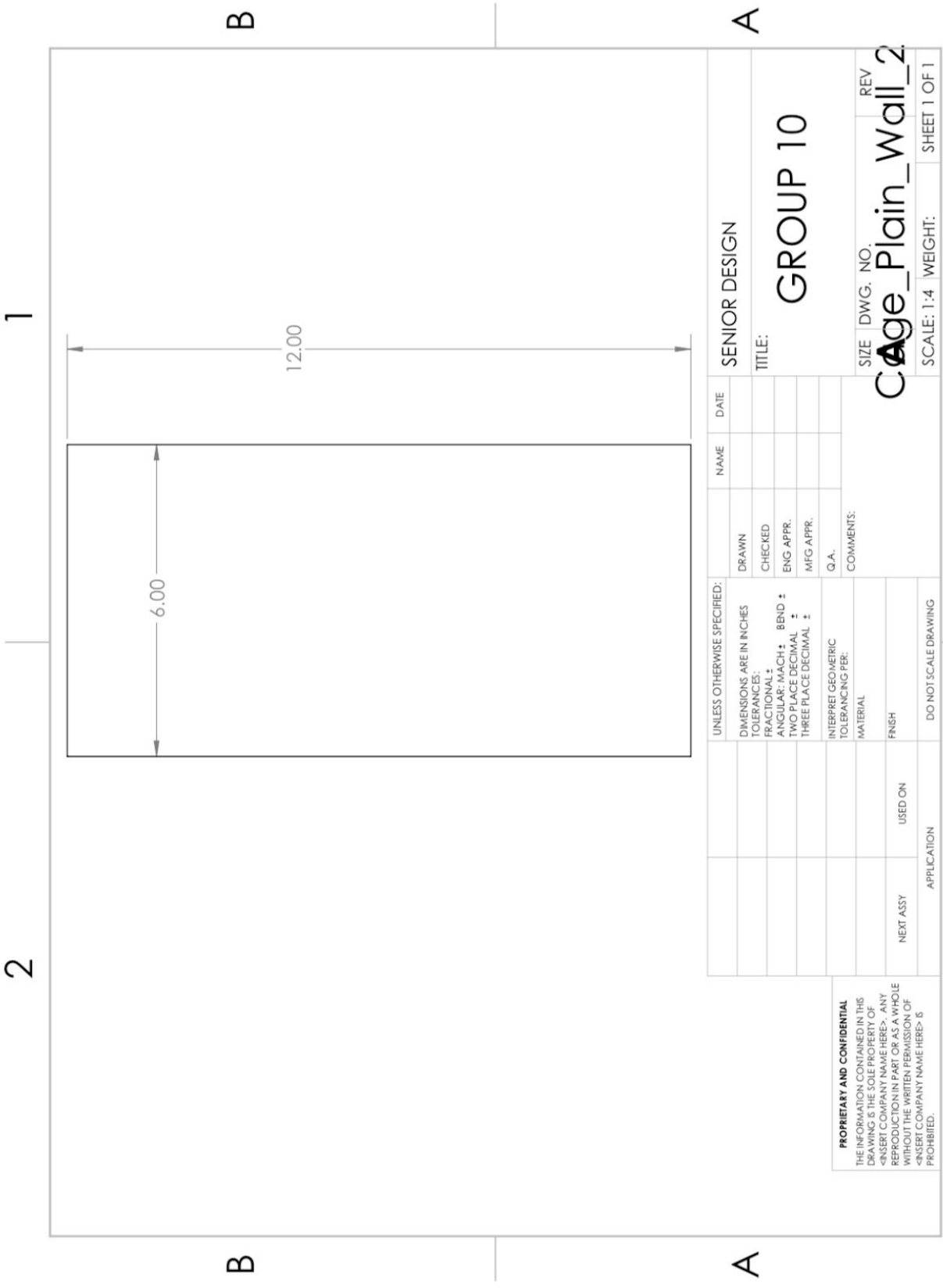
A

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS
 DRAWING IS THE SOLE PROPERTY OF
 <INSERT COMPANY NAME HERE>. ANY
 REPRODUCTION IN PART OR AS A WHOLE
 WITHOUT THE WRITTEN PERMISSION OF
 <INSERT COMPANY NAME HERE> IS
 PROHIBITED.

UNLESS OTHERWISE SPECIFIED:		DRAWN		NAME	DATE	SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		CHECKED				TITLE:	
TOLERANCES:		ENG APPR.				GROUP 10	
FRACTIONAL:		MFG APPR.				SIZE DWG. NO.	
ANGULAR: MACH ±		Q.A.				REV	
BEND ±		COMMENTS:				Cage_Top_Open_2	
TWO PLACE DECIMAL ±		INTERPRET GEOMETRIC				SCALE: 1:2	
THREE PLACE DECIMAL ±		TOLERANCING PER:				WEIGHT:	
		MATERIAL				SHEET 1 OF 1	
		FINISH					
		USED ON					
		APPLICATION					
		DO NOT SCALE DRAWING					

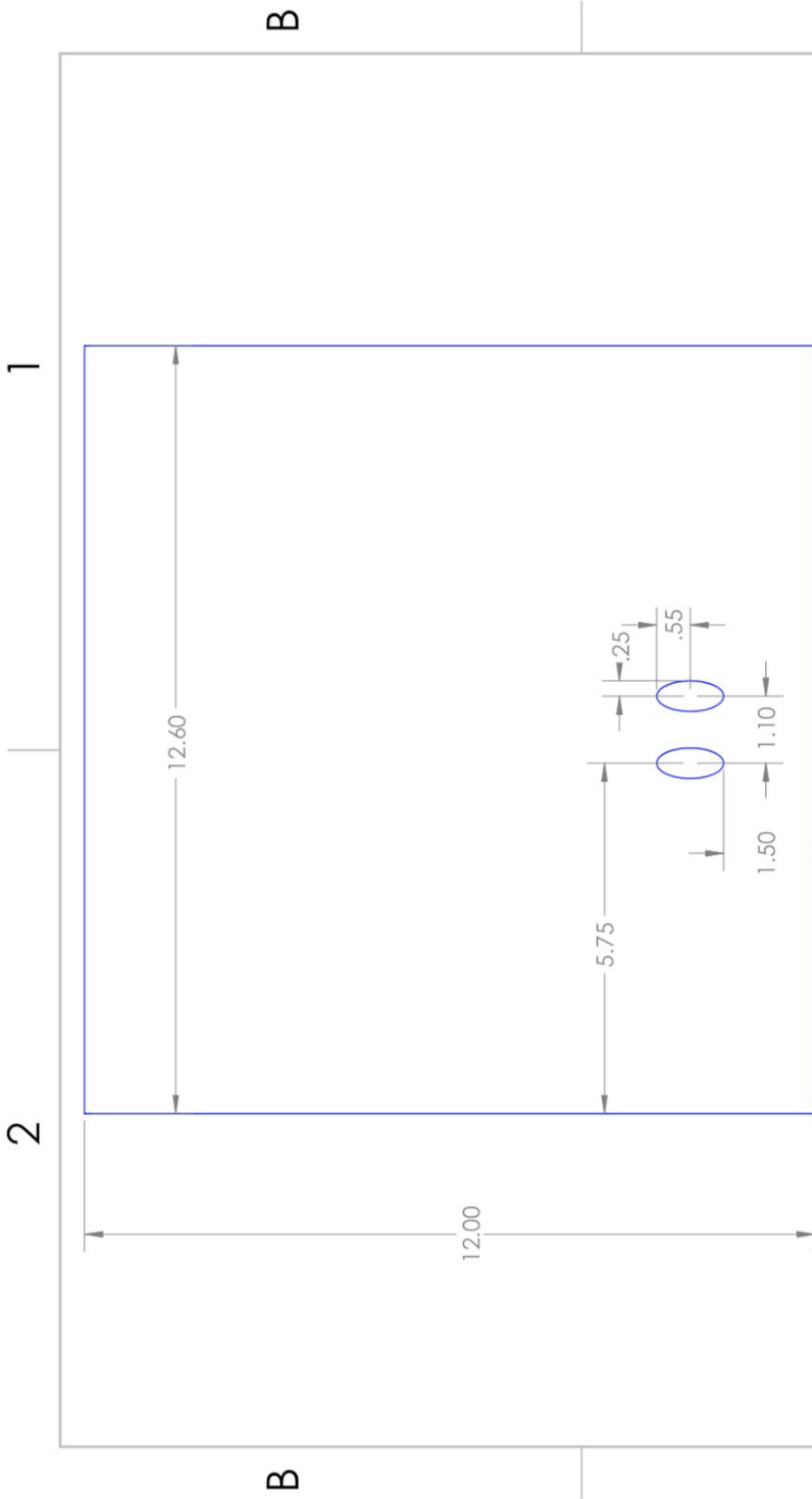
2

1

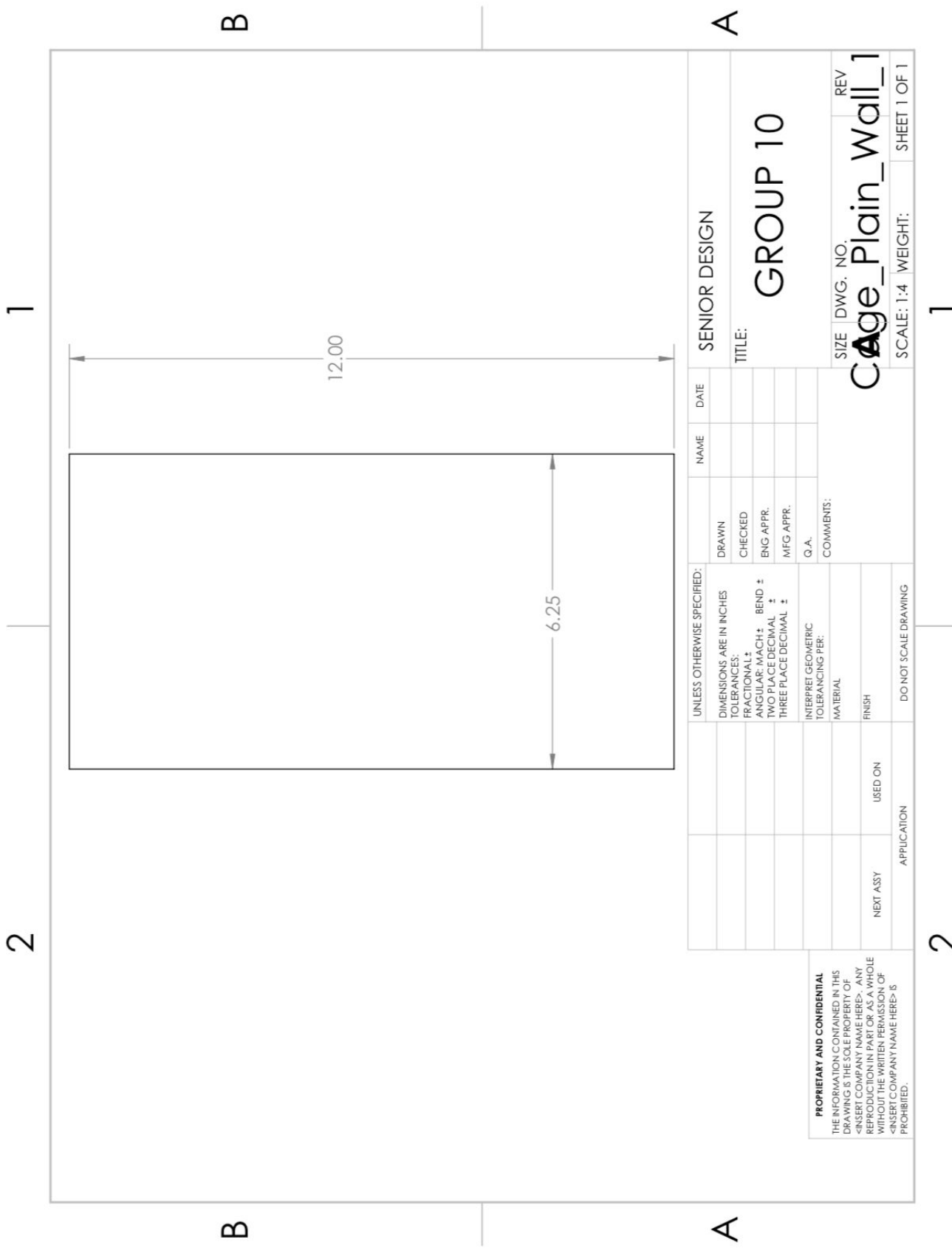


PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		DRAWN		TITLE:	
TOLERANCES:		CHECKED		GROUP 10	
FRACTIONAL:		ENG APPR.		SIZE DWG. NO. REV	
ANGULAR: MACH: BEND:		MFG APPR.		Cage_Plain_Wall_2	
TWO PLACE DECIMAL:		Q.A.		SCALE: 1:4 WEIGHT: SHEET 1 OF 1	
THREE PLACE DECIMAL:		COMMENTS:			
INTERPRET GEOMETRIC TOLERANCING PER:					
MATERIAL:					
FINISH:					
NEXT ASSY	USED ON				
APPLICATION					
DO NOT SCALE DRAWING					

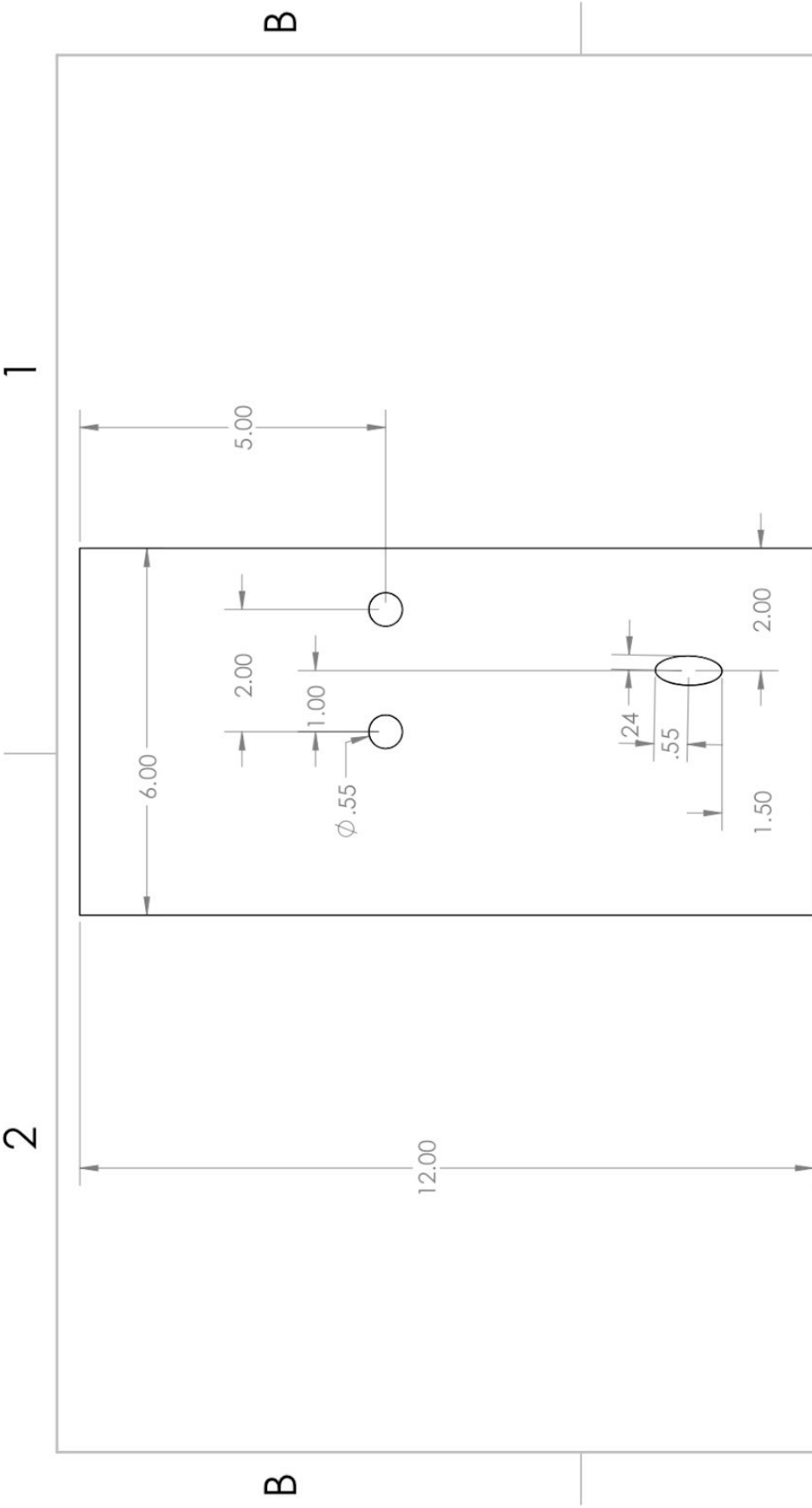


<p>PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.</p>		UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL: ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±		DRAWN CHECKED ENG APPR. MFG APPR.	NAME DATE	SENIOR DESIGN
		INTERPRET GEOMETRIC TOLERANCING PER: MATERIAL FINISH	Q. A. COMMENTS:	TITLE: GROUP 10	SIZE DWG. NO. REV A2_Spout_Wall	
NEXT ASSY USED ON APPLICATION	DO NOT SCALE DRAWING	SCALE: 1:1 WEIGHT:	SHEET 1 OF 1	1	2	1



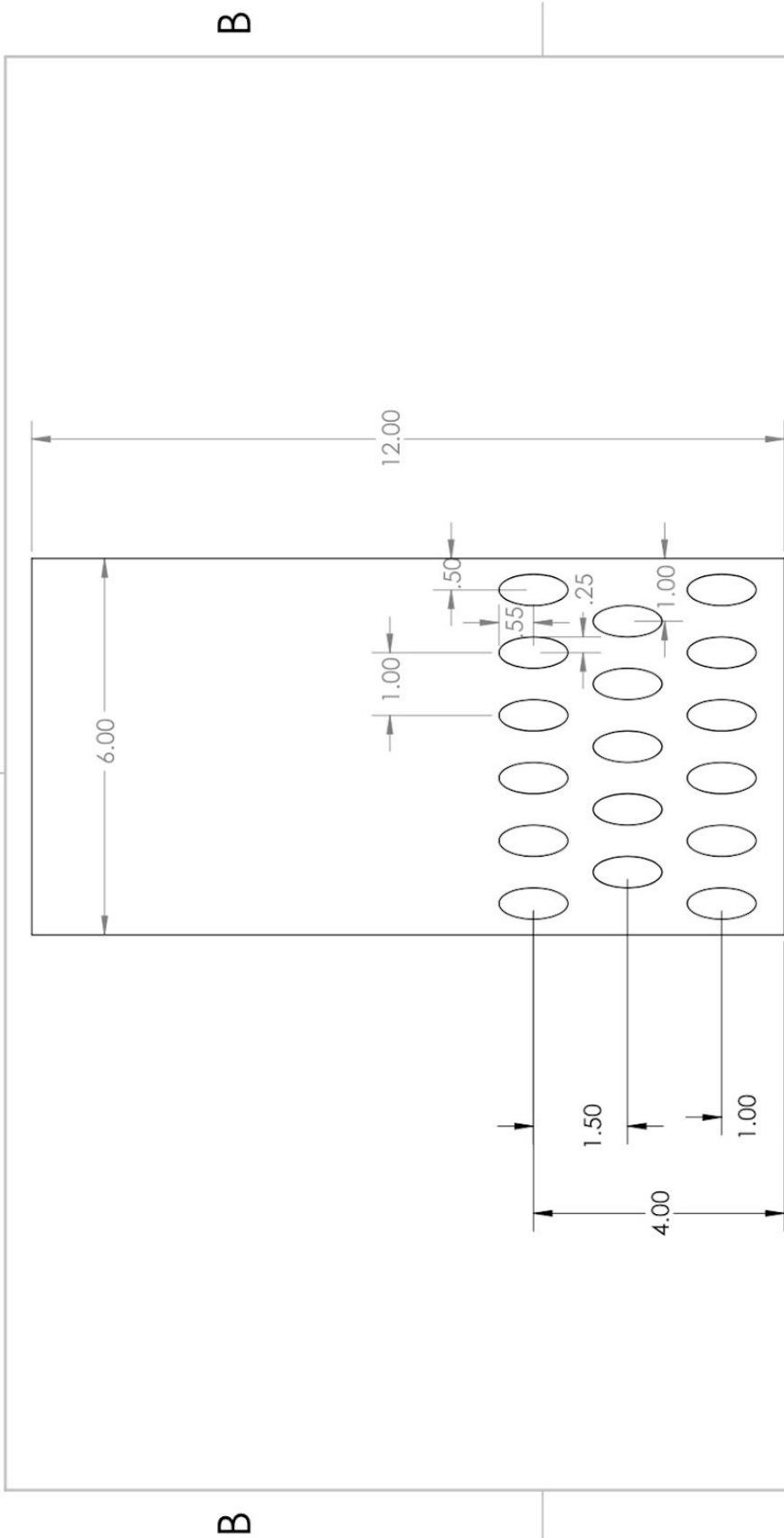
UNLESS OTHERWISE SPECIFIED:		NAME	DATE	SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		DRAWN			
TOLERANCES:		CHECKED			
FRACTIONAL ±		ENG APPR.		TITLE: GROUP 10	
ANGULAR: MACH ± BEND ±		MFG APPR.		SIZE DWG. NO. REV	
TWO PLACE DECIMAL ±		Q.A.		Cage_Plain_Wall_1	
THREE PLACE DECIMAL ±		COMMENTS:		SCALE: 1:4 WEIGHT: SHEET 1 OF 1	
INTERPRET GEOMETRIC TOLERANCING PER:					
MATERIAL					
FINISH					
NEXT ASSY	USED ON				
APPLICATION		DO NOT SCALE DRAWING			

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.



<p>PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.</p>		NEXT ASSY		USED ON		APPLICATION		DO NOT SCALE DRAWING	
		INTERPRET GEOMETRIC TOLERANCING PER:		MATERIAL		FINISH		COMMENTS:	
UNLESS OTHERWISE SPECIFIED:		DRAWN		NAME		DATE		SENIOR DESIGN	
DIMENSIONS ARE IN INCHES		CHECKED		DRAWN		DATE		TITLE:	
TOLERANCES:		ENG APPR.		NAME		DATE		GROUP 10	
FRACTIONAL: MACH ±		MFG APPR.		DRAWN		DATE		SIZE DWG. NO. REV	
ANGULAR: MACH ± BEND ±		Q.A.		CHECKED		DATE		A1_Spout_Wall	
TWO PLACE DECIMAL ±		INTERPRET GEOMETRIC TOLERANCING PER:		ENG APPR.		DATE		SCALE: 1:1 WEIGHT: SHEET 1 OF 1	
THREE PLACE DECIMAL ±		MATERIAL		MFG APPR.		DATE			
		FINISH		Q.A.		DATE			
		DO NOT SCALE DRAWING		COMMENTS:		DATE			

2 1

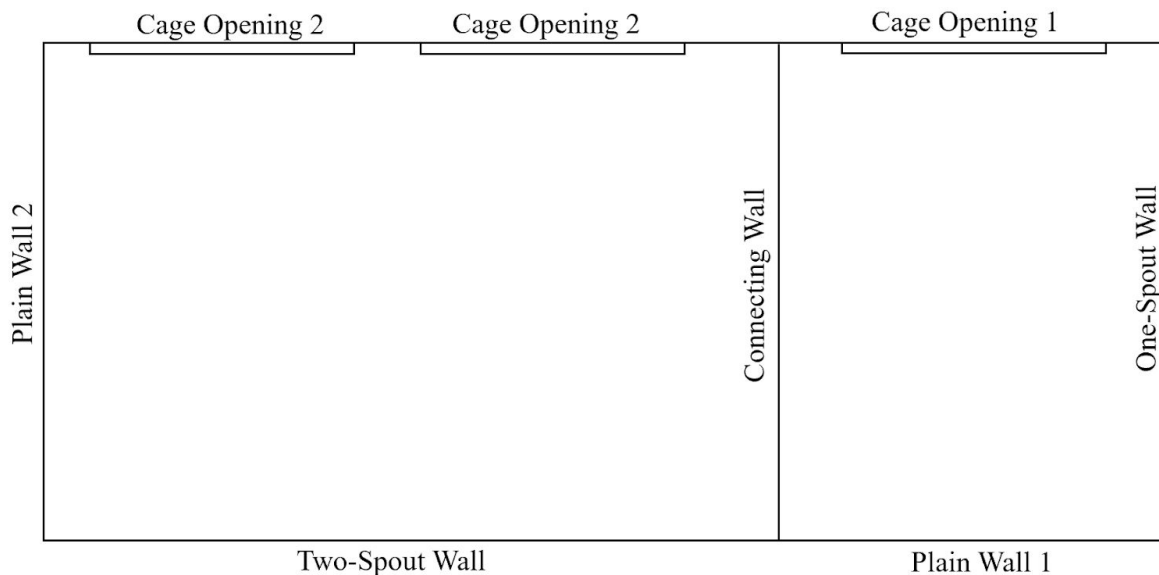


A B

UNLESS OTHERWISE SPECIFIED:		DRAWN		NAME		DATE	
DIMENSIONS ARE IN INCHES		CHECKED		ENGINEER		SENIOR DESIGN	
TOLERANCES:		ENG APPR.		MANAGER		TITLE:	
FRACTIONAL: ±		MFG APPR.		DATE		GROUP 10	
ANGULAR: MACH ±		Q.A.		REV		SIZE DWG. NO.	
BEND ±		COMMENTS:		REV		Connecting_Wall	
TWO PLACE DECIMAL ±		DO NOT SCALE DRAWING		SHEET 1 OF 1		SCALE: 1/4 WEIGHT:	
THREE PLACE DECIMAL ±		APPLICATION		USED ON		1	
INTERPRET GEOMETRIC TOLERANCING PER:		NEXT ASSY		1		1	
MATERIAL		1		1		1	
FINISH		1		1		1	

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF THE COMPANY. IT IS TO BE USED ONLY FOR THE PROJECT AND IN THE MANNER SPECIFIED HEREIN. ANY REPRODUCTION OR TRANSMISSION OF THIS DRAWING IN ANY MANNER WITHOUT THE WRITTEN PERMISSION OF THE COMPANY IS PROHIBITED.

Overhead Cage View



Assembly

To assemble the cage, start with the Cage Bottom. Orient it like in the technical drawing shown above. Next, screw in the 80/20 support beams with the hex bolts to the indented areas in the Cage Bottom. Start with the four corners, then across the top, where the long opening is. Put two beams in each of the indents and secure them with the hex bolts. Next, put in the final beam where the Plain Wall 1 and Connecting Wall intersect. This intersection can be seen in the Overhead Cage View above.

Next, insert the Plain Wall 2 in the orientation seen above. Insert all the walls following the diagram above. When all the walls have been inserted into the 80/20 slotted support beams, put the Top Cage on top. Orient it in the same way as the Cage bottom, with the long opening in the Top Cage lining up with the long opening in the Cage Bottom. Next, screw in the remaining hex bolts to the beams to secure the top to the rest of the cage.

B. Parts list

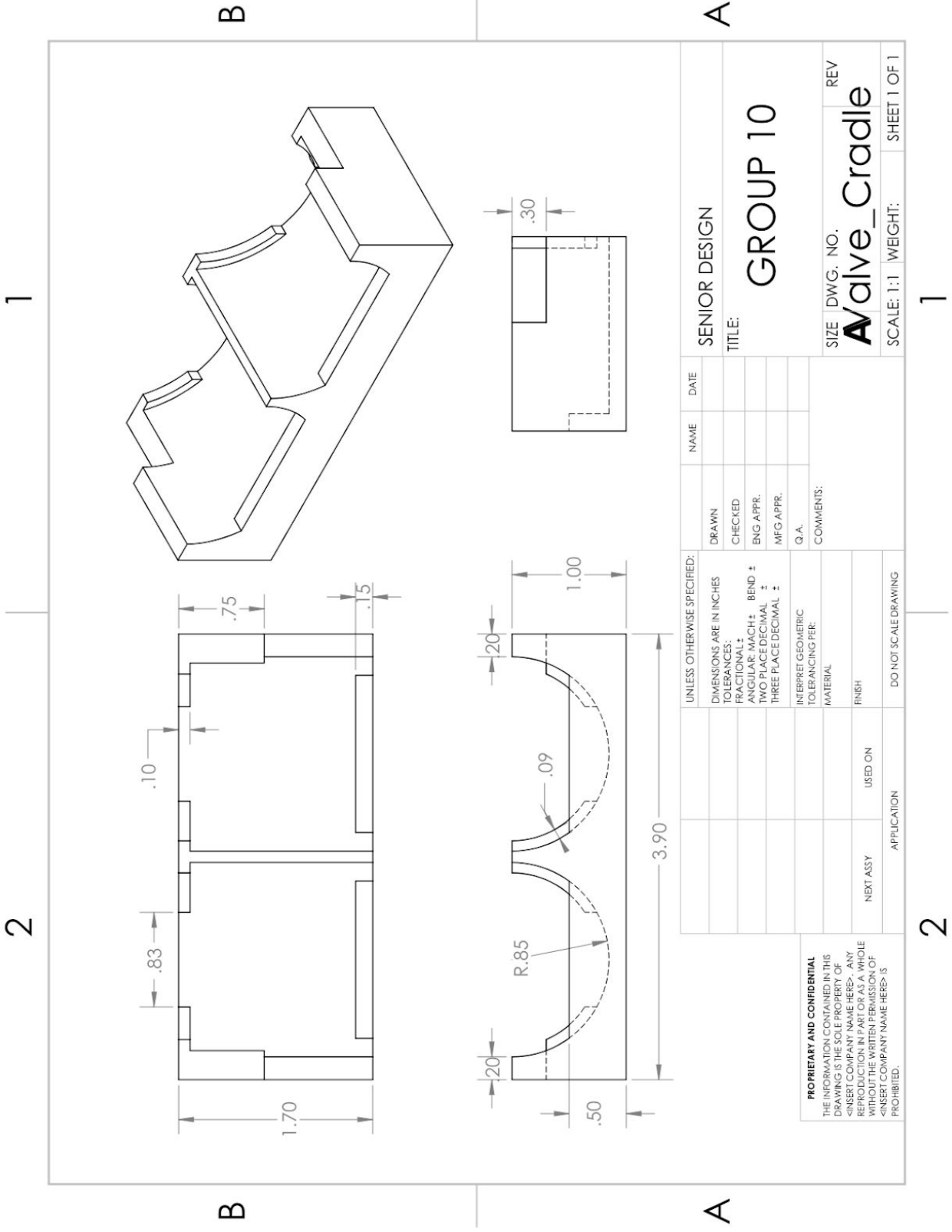
Part	Part Number/Description	Price	Lead Time
Support Beam (7)	80/20 10 series 12 inch long beams (https://8020.net/1010.html)	Provided by the Hengen Lab. \$32.97	Available in Lab
Hex Bolts (14)	Standard end fastener, M6 (https://8020.net/3895.html)	14 x \$1.55 = \$21.70	Available in Lab
Solenoid valve (2)	SNS 2W025-08 1/4"DC12V NPT Brass Electric Solenoid Valve Normally Closed Water, Air, Diesel (https://www.amazon.com/SNS-2W025-08-Electric-Solenoid-Normally/dp/B0794X25XP)	2 x \$15.69 = \$31.38	2 business days
Arduino Mega	Arduino MEGA 2560 microcontroller with 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. (https://www.arduino.cc/en/Main/arduinoBoardMega2560?setlang=en)	Provided by the Hengen Lab. \$38.50	Available in Lab
Arduino UNO	Arduino UNO microcontroller with 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. (https://www.arduino.cc/en/Main/arduinoBoardUno/)	Provided by the Hengen Lab. \$22.00	Available in Lab
Raspberry Pi	Raspberry Pi 3 Model B. Single board computer (https://www.raspberrypi.org/products/raspberry-pi-3-model-b/)	Provided by the Hengen Lab. \$29.99	Available in Lab
Micro SD card	Samsung 128GB 100MB/s (U3) MicroSD EVO Select Memory Card with Adapter (MB-ME128GA/AM) (https://www.amazon.com/Samsung-MicroSD-Adapter-MB-ME128GA-AM/dp/B06XWZWYVP/ref=pd_lpo_vtph_bs_t_1?_encoding=UTF8&psc=1&refRID=Q5JF43SM3SJXF2JDSD9D)	Provided by the Hengen Lab. \$19.99	Available in lab

Monitor	Acer R240HY bidx 23.8-Inch IPS HDMI DVI VGA (1920 x 1080) Widescreen Monitor (https://www.amazon.com/Acer-R240HY-bidx-23-8-Inch-Widescreen/dp/B0148NNKTC)	Provided by the Hengen Lab. \$179.99	Available in Lab
Tubing	3/8 in. I.D. x 1/2 in. O.D. x 20 ft. Clear Vinyl Tubing (https://www.homedepot.com/p/UDP-3-8-in-I-D-x-1-2-in-O-D-x-20-ft-Clear-Vinyl-Tubing-T10007008/304185162)	\$8.50	Picked up in store
Tubing connectors (4)	1/4 in. O.D. x 3/8 in. Nptf Polypropylene Push-To-Connect Male Connector (2) (https://www.homedepot.com/p/John-Guest-1-4-in-O-D-x-3-8-in-Nptf-Polypropylene-Push-To-Connect-Male-Connector-803359/300753485) 1/4 in. x 1/4 in. MIP Lead-Free Brass Hose Barb Adapter (2) (https://www.homedepot.com/p/Everbilt-1-4-in-x-1-4-in-MIP-Lead-Free-Brass-Hose-Barb-Adapter-800119/300096726)	2 x \$3.87 = \$7.74 2 x \$3.22 = \$6.44	Picked up in store
Spout (2)	1" Bend Tubes - Ball Point (http://www.ancare.com/products/watering-equipment/open-drinking-tubes/1-bend-tubes-ball-point)	2 x \$2.26 = \$4.52	Available in Lab
Speakers	Stereo Enclosed Speaker Set - 3W 4 Ohm (https://www.adafruit.com/product/1669?gclid=Cj0KCQjw19DIBRCSARIsAOnfRejB0_RyxxKXdNA6q5YGZhd2htKvhiAxrDEpSDkq0cG6bg4hZfyPf74aAvcpEALw_wcB)	\$7.50	4 business days
Capacitive sensor	Adafruit 12 x Capacitive Touch Shield for Arduino - MPR121 (https://www.adafruit.com/product/2024)	\$12.50	Picked up in store
Wall power supply	12V DC 2A Wall Power Supply Adapter with 2.1mm x 5.5 Plug 2A(2000MA) AC 100-240V to DC 12Volt Transformers, Switching Power Source Adaptor for 12V 3528/5050 LED Strip Lights (https://www.amazon.com/Adapter-100-240V-Transformers-Switching-Adaptor/dp/B07GRZB5Y9/ref=asc_df_B07GRZB5Y9/?tag=hyprod-20&linkCode=df0&hvadid=242082743678&hvpos=1o2&hvnetw=g&hvrand=15552248720268098738&hvpo)	Provided by the Hengen Lab. \$8.98	Available in Lab

	ne=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9022860&hvtargid=pla-522274682760&pssc=1)		
Driving Motor (2)	Velleman VMA411: Arduino Compatible MOS Driving Module (https://www.vellemanstore.com/en/velleman-vm-a411-arduino-compatible-mos-driving-module)	2 x \$4.45 = \$8.90	Picked up in Store
Bread board	Spark Fun Solder-able Bread Bread Board (https://www.sparkfun.com/products/12070)	\$4.95	Picked up from Klaesner Lab
Rubber stopper	Rubber Stopper 6 series (http://www.ancare.com/products/watering-equipment/stoppers/rubber-stoppers)	2 x \$0.73 = \$1.46	Available in Lab
HDMI to HDMI cable	https://www.amazon.com/AmazonBasics-High-Speed-HDMI-Cable-1-Pack/dp/B014I8SIJY/ref=asc_df_B014I8SIJY/?tag=hyprod-20&linkCode=df0&hvadid=177496114009&hvpos=1o1&hvnetw=g&hvrand=6367069310414155760&hvpone=&hvp two=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9022860&hvtargid=pla-195029131035&pssc=1	Provided by the Hengen Lab \$5.99	Available in lab
Power cable (monitor)	5V 10 A power supply (https://www.amazon.com/ALITOVE-Adapter-Converter-Charger-5-5x2-1mm/dp/B01M0KLECZ/ref=sr_1_3?hvadid=234341840299&hvdev=c&hvlocphy=9022860&hvnetw=g&hvpos=1t1&hvqmt=e&hvrand=12922531752076792666&hvtargid=aud-646675773986%3Akwd-58148118197&keywords=5v+10+amp+power+supply&qid=1555390182&s=gateway&sr=8-3)	Provided by the Hengen Lab. \$20.99	Available in lab
Power cable (Pi)	Raspberry Pi Micro USB Power Supply 5V1 2.5A (https://www.microcenter.com/product/462652/Micro_USB_Power_Supply_5V1_25A_-_White?src=raspberrypi)	Provided by the Hengen Lab \$10.99	Available in Lab
Wires	Velleman 30cm 40-Pin M-F Flat Cable Jumper Wire. M-M Flat Cable Jumper Wire (https://www.microcenter.com/product/476369/30cm-40-pin-m-f-flat-cable-jumper-wire)	\$7.99	Picked up in store
Alligator clips	https://www.microcenter.com/product/456418/sm-all-alligator-clip-test-lead	\$6.99	Picked up in store

USB A to B	https://www.microcenter.com/product/486571/-usb-a-to-micro-b-cable-236	\$2.99	Picked up in store
Tin foil	https://www.amazon.com/Reynolds-Wrap-Aluminum-Foil-Square/dp/B00I9J3Q8W/ref=sr_1_7?hvdid=153652607925&hvdev=c&hvlocphy=9022860&hvnetw=g&hvpos=1t1&hvqmt=b&hvrnd=9768381855861996104&hvtargid=aud-646675773986%3Akwd-4756803090&keywords=aluminum+foil+rolls&qid=1555608764&s=pantry&sr=8-7	\$9.98	Available in Lab

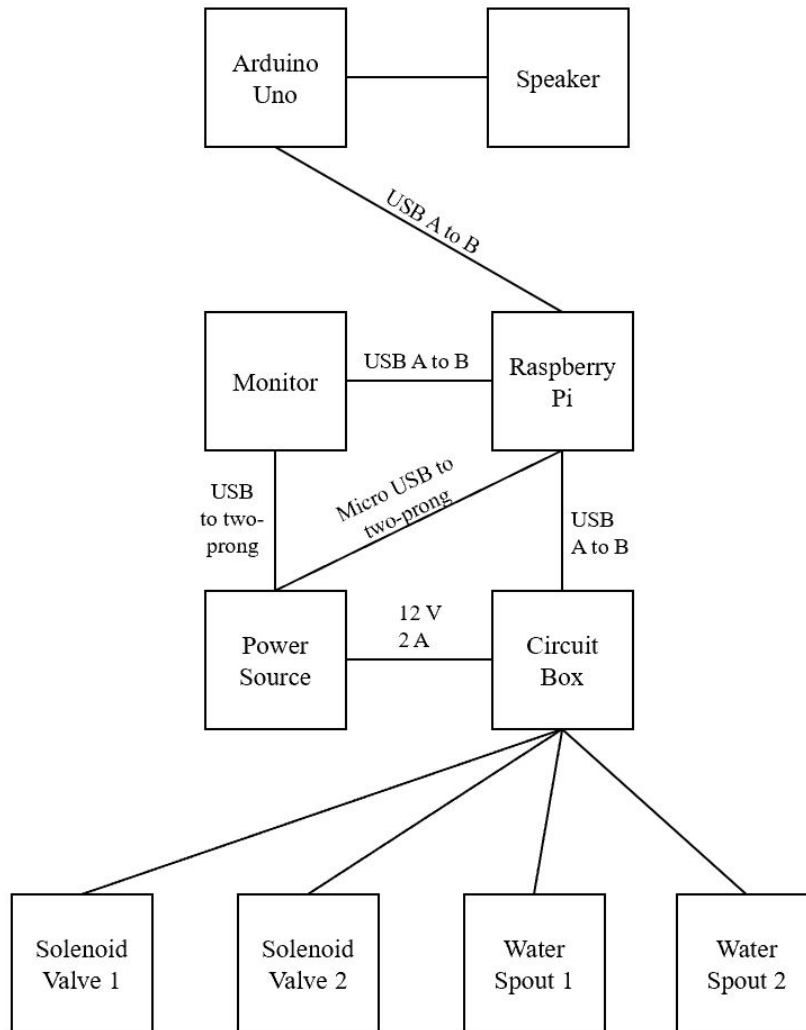
C. Valve Cradle - Drawing



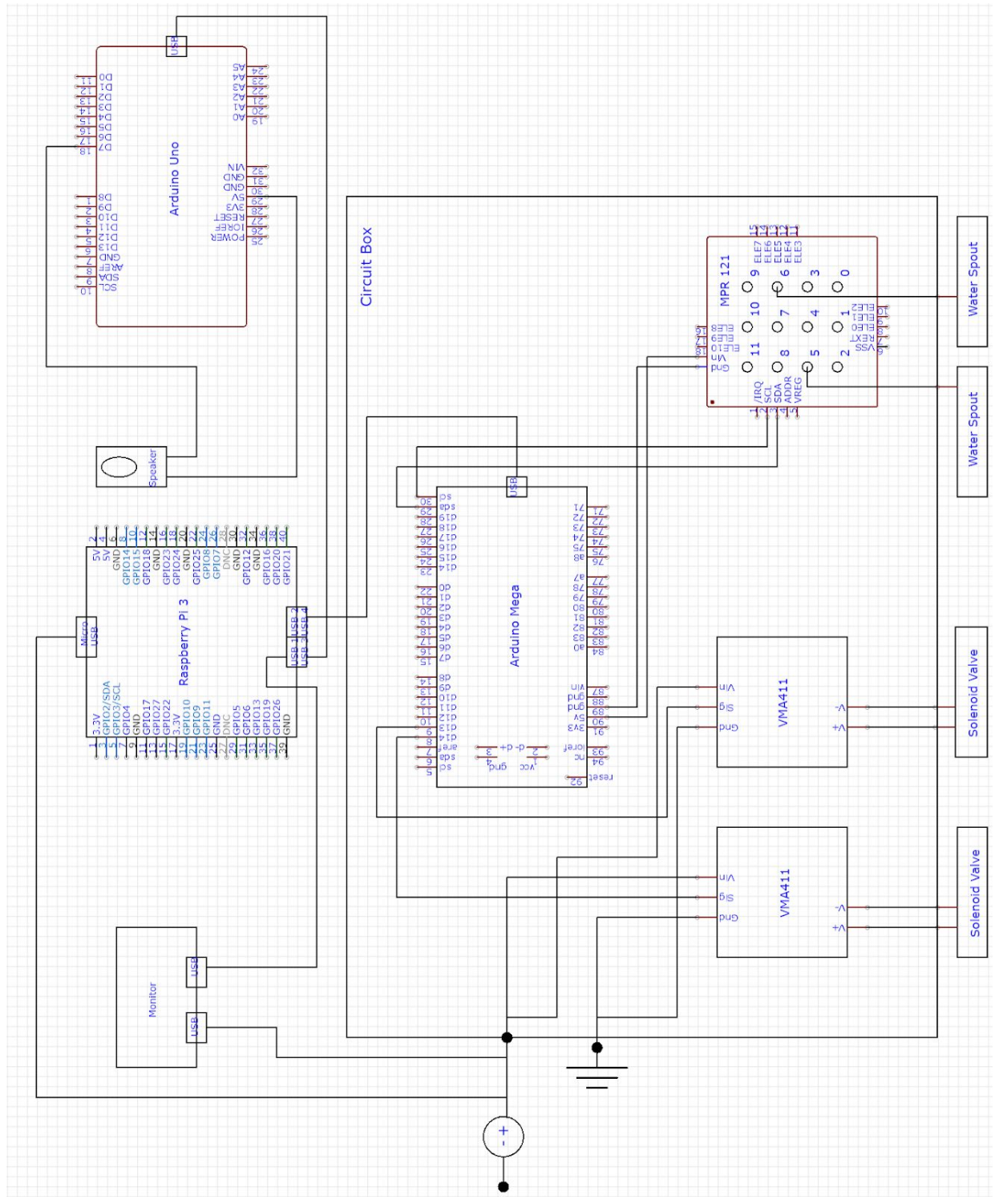
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONS: ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±		DRAWN	NAME	DATE	SENIOR DESIGN	
INTERPRET GEOMETRIC TOLERANCING PER: MATERIAL		CHECKED			TITLE: GROUP 10	
FINISH		ENG APPR.			SIZE	DWG. NO.
NEXT ASSY		MFG APPR.			A Valve_Cradle	
USED ON		Q.A.			SCALE: 1:1	WEIGHT:
APPLICATION		COMMENTS:			SHEET 1 OF 1	
DO NOT SCALE DRAWING					1	

PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE PROPERTY OF
 <INSERT COMPANY NAME HERE> AND
 REPRODUCTION IN PART OR AS A WHOLE
 WITHOUT THE WRITTEN PERMISSION OF
 <INSERT COMPANY NAME HERE> IS
 PROHIBITED.

D. Wiring Diagram



E. Circuit Diagram



F. Code

controlRAT.py: The script that allows the Raspberry Pi to coordinate the water delivery system, image presentation on the monitor, and tone delivery. When the time is between 9 am and 10 am, it runs the testing portion of the code. It loads folder containing paired graded images, as well as a file named decode.csv. This text file contains which of the sides - left or right - in the graded image pairs that should be labelled as the correct choice; a value of 0 indicates that the left side is correct, and a value of 1 indicates that the right image is correct. The image pair that will be presented is chosen at random. A character that indicates the correct side choice is then sent as serial data to the Arduino, which reads it and alters which valve will open in response to a lick. controlRAT.py also contains a function for reading serial input from the Arduino Mega - this input is parsed into variables "lick," "spout," and "reward." The values of these variables allow the Raspberry Pi to track the mouse's behavior and will be elaborated upon more in the description for MPR121andSolenoid.ino that follows the python code. While the current time is outside of 9 am -10 am testing time, controlRAT.py sends a serial signal to the Arduino Mega to open either valve upon any lick.

```
import matplotlib.pyplot as plt
import cv2
import random
import time
import serial
import numpy as np
from datetime import datetime
import csv

# Track time
localTime = time.asctime(time.localtime(time.time()))
print(localTime)

def sendChoice(choice):
    """Sends a single character to the Arduino Mega's Serial monitor,
    indicating which screen is correct."""

    if choice is 1:
        print("Right image is correct")
    else:
        print("Left image is correct")

    ser = serial.Serial('/dev/ttyACM0',9600, timeout=.1)
    ser.flushInput()
    time.sleep(1)
    ser.write(choice.encode())
    time.sleep(1)
    #ardu.close()

def checkLick():
    """Reads three numbers from the Arduino Mega's Serial Monitor, and
    returns their values (which can be either 0 or 1)"""

    ser = serial.Serial('/dev/ttyACM0',9600, timeout=.1)
    ser_bytes = ser.readline()
    decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
    lick = decoded_bytes[0]
```

```

    spout = decoded_bytes[1]
    reward = decoded_bytes[2]
    return lick, spout, reward

def sendNoTest():
    """Sends a single character to the Arduino Mega indicating that it is
    not time for testing."""

    ser = serial.Serial('/dev/ttyACM0',9600, timeout=.1)
    ser.flushInput()
    time.sleep(1)
    message = 2
    ser.write(message)
    time.sleep(1)
    #ardu.close()

def playTone():
    """Sends a single character to the Arduino Uno, indicating that it
    should play a tone."""

    ser = serial.Serial('/dev/',9600, timeout=.1) # Arduino Uno
    ser.flushInput()
    time.sleep(1)
    message = 1
    ser.write(message)
    time.sleep(1)
    #ardu.close()

def checkTime():
    """Returns 1 if the current time is within the 9am-10am testing
    period. Retrurns 0 if it is not time for testing"""

    nw = time.localtime(time.time())
    if nw[3] is 9:
        return 1
    else:
        return 0

# load folder of images (each image has two sides)

```

```

path = "/Users/eleanordegen/Desktop/RATimages/*.*)"
allImages = glob.glob(path)

# Load and parse decision text file
leftRightChoices = list()
with open('RATdecode.csv') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    line_count = 0
    for row in csv_reader:
        if line_count == 0:
            print(f'Column names are {", ".join(row)}')
            line_count += 1
        else:
            leftRightChoices.append(row[0])
            line_count += 1
    print(f'Processed {line_count} lines.')

# recordAllLicks:
# 0 (left spout) or 1 (right spout) in first column
# 0 (incorrect choice) or 1 (correct choice) in the second column
# time and date of lick in third column
recordAllLicks = list()

testTime = checkTime() # test 9am-10am
testTime = 1
while testTime is 1:

    # Randomly pick which image to present
    randInd = random.randint(0, len(leftRightChoices)-1)
    pairGrad = allImages[randInd]

    # Check which side is correct (0 for left, 1 for right)
    choice = leftRightChoices[randInd]

    # Display images
    GRAD = 'pair of graded images'
    window = cv2.namedWindow(pairGrad, cv2.WINDOW_NORMAL)
    cv2.resizeWindow(pairGrad, 1500,1500)
    cv2.imshow(pairGrad,img)

```

```

cv2.moveWindow(pairGrad,0,0)

# send the decision to the water control Arduino
sendChoice(choice)

start = time.time() # track time image is displayed
end = time.time()
elapsed = end - start

countLicks = 0
lick = 0
while lick is 0:
    end = time.time()
    elapsed = end - start
    cv2.waitKey(1000)

    # monitor for licks
    # lick = 0 or 1 (lick or no lick)
    # spout = 0 or 1 (left or right)
    # reward = 0 or 1 (deliver water or not)
    lick, spout, reward = checkLick()
    localTime = time.asctime(time.localtime(time.time()))

    # If there is a lick: deliver water, pause for 5 sec, show next image
    if lick is 1 and reward is 1:
        countLicks = countLicks+1
        recordAllLicks.append([spout, reward, localTime])
        if spout is 0:
            print('Rodent licked the correct spout (left) at {}'.localTime)
        elif spout is 1:
            print('Rodent licked the correct spout (right) at {}'.localTime)

        # play reward tone
        playTone()

        time.sleep(20) # 20 sec pause of system (while rodent drinks)
        break
    elif lick is 1 and reward is 0:
        countLicks = countLicks+1

```

```
recordAllLicks.append([spout, reward, localTime])
if spout is 0:
    print('Rodent licked the incorrect spout (left) at {}'.localTime)
elif spout is 1:
    print('Rodent licked the incorrect spout (right) at {}'.localTime)
    time.sleep(40) # 40 sec pause of system (time-out punishment)

cv2.destroyAllWindows()

testTime = checkTime() # test time

while testTime is 0:
    # instead of a screen choice (0 or 1), Arduino will receive a 2 to indicate no
testing
    sendNoTest()
    testTime = checkTime() # test time
```

MPR121andSolenoid.ino: This Arduino code has been uploaded to the Arduino Mega that controls the solenoid valves and reads from the capacitive sensor. The Arduino Mega reads serial data from the Raspberry Pi indicating which is the correct side - left or right, and adjust its opening of the solenoid valves accordingly. Addition, the Arduino Mega sends serial data back to the Raspberry Pi in three characters, each a zero or one. The first of these characters indicates if there has been a lick (1) or if there was no lick detected for that loop (0). The second of these characters indicates which spout was licked (0 for left, 1 for right), and the third indicates whether the corresponding valve was opened or remained closed (0 for closed, 1 for opened). If there was no lick, controlIRAT.py on the Raspberry Pi ignores the valves of the second two characters.

```

#include <Adafruit_MPR121.h>

/*****
TMPR121 12-channel Capacitive touch sensor and IRF520 MOSFET Switch module circuit.
*****/

#include <Wire.h>
#include "Adafruit_MPR121.h"

#ifndef _BV
#define _BV(bit) (1 << (bit))
#endif

// You can have up to 4 on one i2c bus
Adafruit_MPR121 cap = Adafruit_MPR121();

// Keeps track of the last pins touched
// so we know when buttons are 'released'
uint16_t lasttouched = 0;
uint16_t currntouched = 0;
char receivedChar;
boolean newData = false;

#define control 8 // pin that controls driving motor 1
#define apple 10 // pin that controls driving motor 2

void setup() {

  // define the controlling pins as outputs
  pinMode(control,OUTPUT);
  pinMode(apple,OUTPUT);

  Serial.begin(9600);

  while (!Serial) { // needed to keep from learning too fast
    delay(10);
  }
}

```



```

// Default address is 0x5A, if tied to 3.3V its 0x5B
// If tied to SDA its 0x5C and if SCL then 0x5D
if (!cap.begin(0x5A)) {
  while (1);
}
}

int rightCorrect = 1;
int leftCorrect = 0;

void loop() {

  int incomingByte = 0;
  int lick = 0; // 0 or 1 (lick or no lick)
  int spout = 0; // 0 or 1 (left or right)
  int reward = 0; // 0 or 1 (deliver water or not)

  // Start with right=correct, left=incorrect
  // signal from python --> switch left/right assignments
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    int orig = leftCorrect;
    leftCorrect = rightCorrect;
    rightCorrect = orig;
  }

  int sensornumber1=6; // left
  // Serial.print("6 filtered: "); Serial.print(cap.filteredData(sensornumber1));
  // Serial.print("6 baseline: "); Serial.print(cap.baselineData(sensornumber1)); Serial.println();

  int sensornumber2=5; // right
  // Serial.print("5 filtered: "); Serial.print(cap.filteredData(sensornumber2));
  // Serial.print("5 baseline: "); Serial.print(cap.baselineData(sensornumber2)); Serial.println();

  // IRF520 MOSFET driving motor control
  int threshold1=150;
  if (leftCorrect == 1) {
    if (cap.filteredData(sensornumber1)<threshold1) {
      // Serial.print("SOLENOID ON READING (6): "); Serial.print(cap.filteredData(sensornumber1));
      // Serial.println();

      digitalWrite(control,HIGH); // turn driving motor 1 ON
      delay(1000); // Wait 1 sec

      digitalWrite(control,LOW); // turn driving motor 1 OFF
      delay(1000);

      lick = 1;
      spout = 0;
      reward = 1;
    }
  }
}

```

```

} else if(cap.filteredData(sensornumber1)<threshold1) {
    lick = 1;
    spout = 0;
    reward = 0;
}

int threshold2=2100;
if (rightCorrect == 1) {
    if (cap.filteredData(sensornumber2)<threshold2) {
//      Serial.print("SOLENOID ON READING (5): "); Serial.print(cap.filteredData(sensornumber2));
//      Serial.println();

        digitalWrite(apple,HIGH); // turn driving motor 2 ON
        delay(1000);

        digitalWrite(apple,LOW); // turn driving motor 2 OFF
        delay(1000);

        lick = 1;
        spout = 1;
        reward = 1;
    }
} else if (cap.filteredData(sensornumber2)<threshold2) {
    lick = 1;
    spout = 1;
    reward = 0;
}

Serial.print(lick);
Serial.print(spout);
Serial.print(reward);

// comment out this line for detailed data from the sensor!
return;

// delay so it isn't overwhelming
// delay(1000);
}

```

testSpeaker.ino: This code is uploaded onto the Arduino Uno that controls the speaker. When it receives a serial signal from the Raspberry Pi, it plays a tone.

```
#define NOTE_AS4 466
int incomingByte = 0; // for incoming serial data

void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {

  if (Serial.available() > 0) {
    tone(8, NOTE_AS4, 800);
    // read the incoming byte:
    incomingByte = Serial.read();
    tone(8, NOTE_AS4, 800);
  }

  delay(2000);
}
```

G. Manual

To use the RAT system, perform the following steps:

1. Verify the hardware setup
 - a. The Raspberry Pi is connected to the monitor with an HDMI-HDMI cord
 - b. The Raspberry Pi is powered from a wall power supply to its micro USB port
 - c. The Arduino Uno is connected to the specified USB port on the Raspberry Pi
 - d. The Arduino Mega is connected to the specified USB port on the Raspberry Pi
 - e. The green taped wires from the driving motor are alligator clipped to the green taped solenoid valve, and the green taped wire from the MPR121 is connected to the spout corresponding to the green taped valve
 - f. The non green taped wires from the driving motor are alligator clipped to the remaining solenoid valve and the other wire from the MPR121 is connected to the remaining spout
 - g. The monitor is connected to power
 - h. The circuit box is connected to power
 - i. The reservoirs are filled with water



- Save a folder of .png images to the Desktop of the Raspberry Pi, titled RATimages. Two example images are provided below.



Images can be generated at:

https://www.cogsci.nl/pages/gabor-generator.php?option=com_content&view=article&Itemid=63&id=50&generate=yes&orient=45&size=180&env=none&std=12&freq=0.1&phase=0&red0=128&green0=128&blue0=128&red1=255&green1=255&blue1=255&red2=0&green2=0&blue2=0

- Save a file named decode.csv to the Desktop of the Raspberry Pi. An example format for this file is shown below.

decode

Image	Correct_Choice	Correct_Angle	Incorrect_Angle
1	L	45	-45
2	L	45	90
3	L	45	50
4	R	45	-45
5	R	45	90
6	R	45	50

- Open the terminal of the Raspberry Pi, and navigate to the controlling python code (saved on the Desktop) with the command “cd Desktop/”. Then, run the command “python controlRAT.py” to run the script.

5. Once controlRAT.py is running, sit back and relax. If the time is between 9 am and 10 am, the Raspberry Pi will automatically initiate testing.
6. For a summary of lick data, navigate to recordedLicks.txt on the Desktop of the Raspberry Pi. This text file is continually updated as controlRAT.py runs, saving data showing which spout was licked and whether a water reward was delivered.

References

1. Rivalan, Marion, et al. "An Automated, Experimenter-Free Method for the Standardised, Operant Cognitive Testing of Rats." PLOS ONE, Public Library of Science, 6 Jan. 2017, journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0169476.
2. Poddar, Rajesh, et al. "A Fully Automated High-Throughput Training System for Rodents." PLOS ONE, Public Library of Science, 6 Dec. 2013, journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0083171.
3. Sorge RE, Martin LJ, Isbester KA, Sotocinal SG, Rosen S, Tuttle AH, et al. Olfactory exposure to males, including men, causes stress and related analgesia in rodents. *Nat Methods*. 2014 Jun; 11 (6):629–32. doi: 10.1038/nmeth.2935 PMID: 24776635
4. Goltstein PM, Reinert S, Glas A, Bonhoeffer T, Hübener M (2018) Food and water restriction lead to differential learning behaviors in a head-fixed two-choice visual discrimination task for mice. *PLoS ONE* 13(9): e0204066. <https://doi.org/10.1371/journal.pone.0204066>
5. Lee, Seung-Hee, et al. "Activation of specific interneurons improves V1 feature selectivity and visual perception." *Nature*, no. 488, 16 Aug. 2012, pp. 379-83.
6. Prusky, Glen T., et al. "Behavioral assessment of visual acuity in mice and rats." *Vision Research*, vol. 40, no. 16, 1 July 2000, pp. 2201-09.