Report Exploratory Making

Bente Elst
student number: 1451189

Cube:

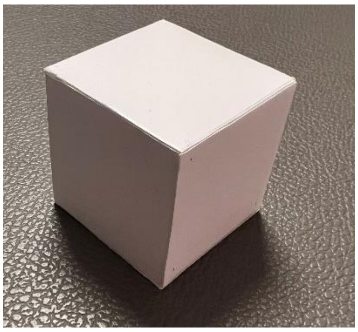


Figure 1, side/top view cube

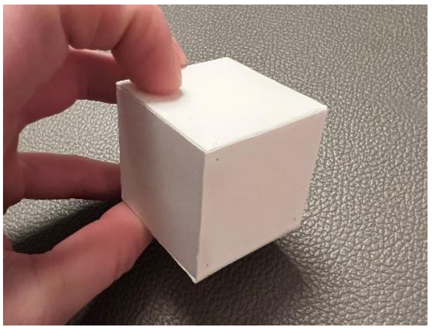


Figure 2, side/bottom view cube

Cylinder:


Figure 3, side/top view cylinder

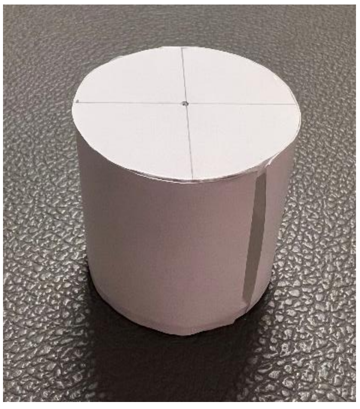


Figure 4, view of gap on cylinder side


Figure 5, bottom/side view cylinder

Cube-Cylinder Combi:

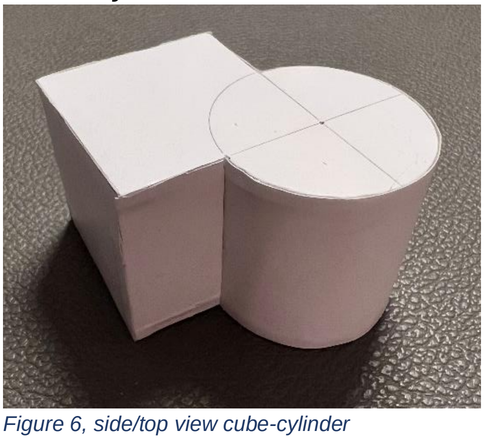


Figure 6, side/top view cube-cylinder

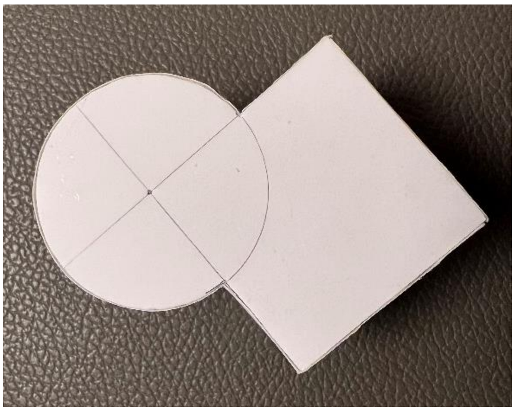


Figure 7, top view cube-cylinder

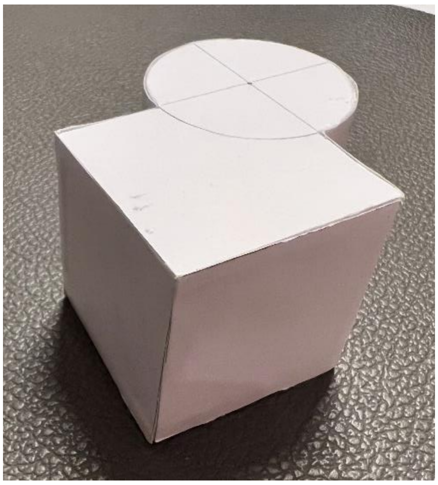


Figure 8, side view cube part of cube-cylinder


Figure 9, connection cylinder side to cube side

Cone:


Figure 10, side/top view cone


Figure 11, top view cone


Figure 12, connection side part


Figure 13, Bottom view cone

Slider:

Figure 14, side view closed slider

Figure 15, top view open slider

Figure 16, side view open slider

Rotator:

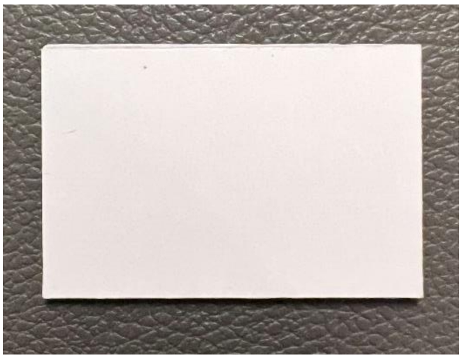


Figure 17, top view closed rotator

Figure 18, side view closed rotator

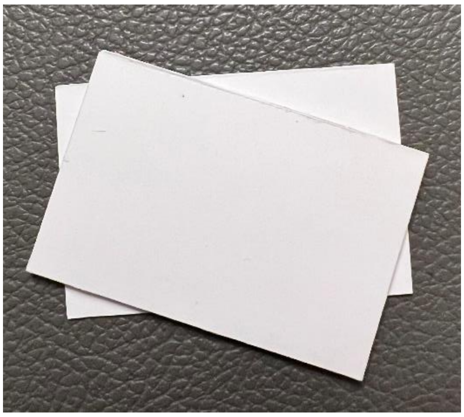


Figure 19, top view half open rotator

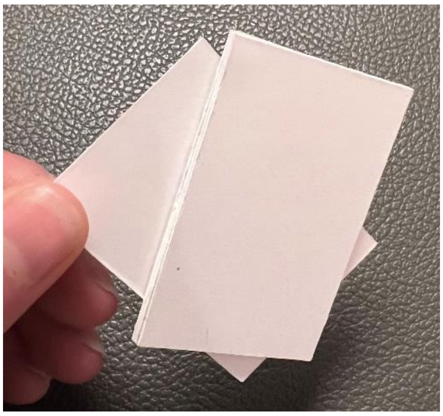


Figure 20, side/top view half open rotator

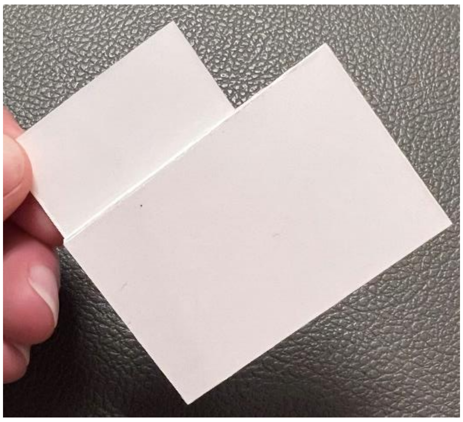


Figure 21, top view open rotator

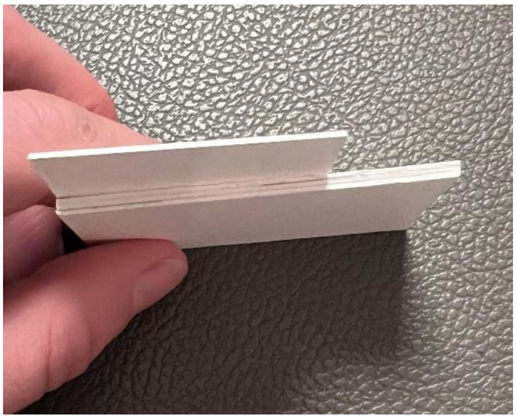


Figure 22, side view open rotator

Beam:

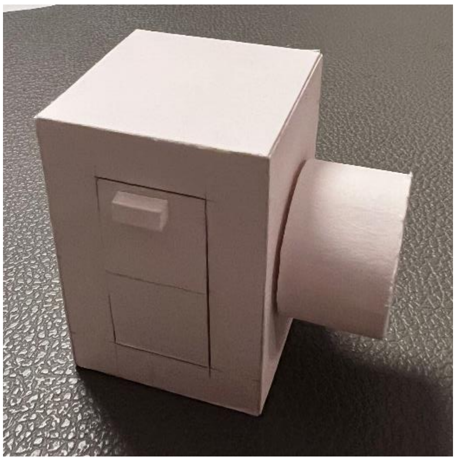


Figure 23, Side rotating knob and 4 bar linkage

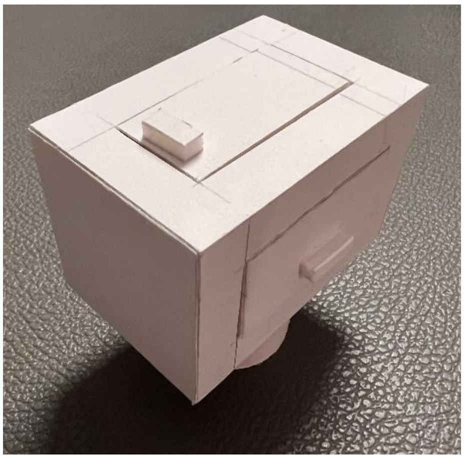


Figure 24, side door and drawer

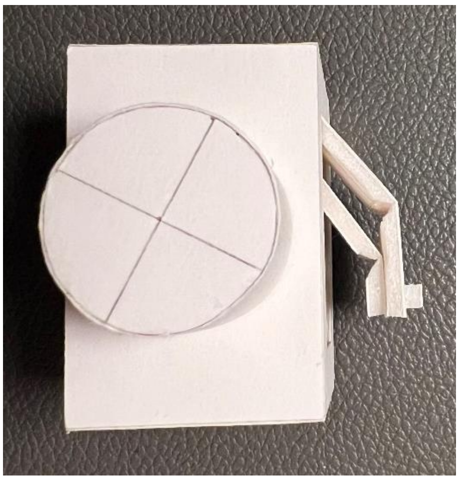


Figure 25, four bar linkage open side view

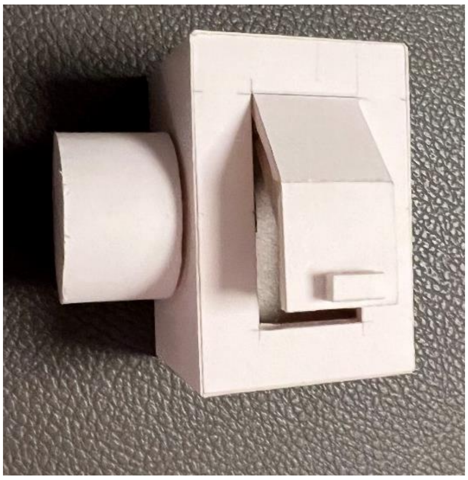


Figure 26, four bar linkage open top view

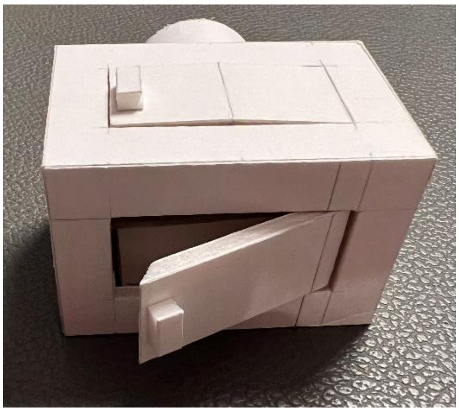


Figure 27, half open drawer top/front view

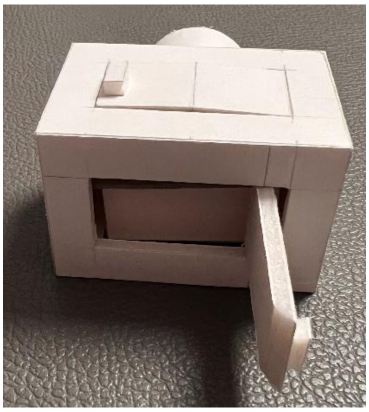


Figure 28, open drawer top/front view

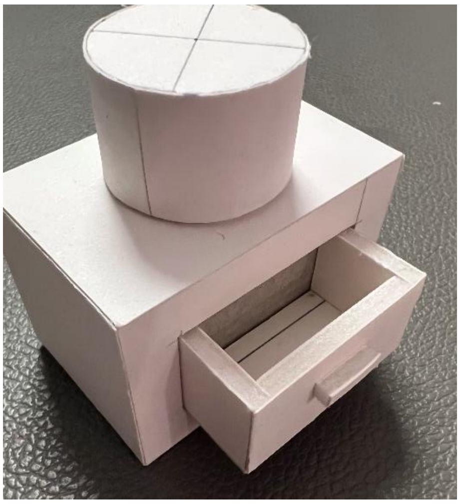


Figure 29, open drawer top/side view


Figure 30, closed drawer front/side view

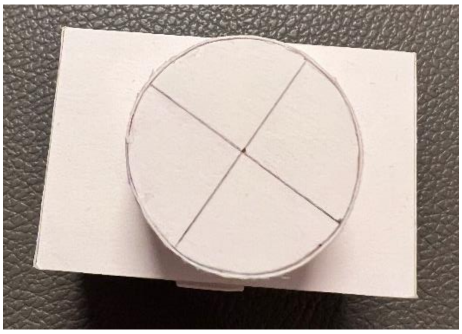


Figure 31, rotating knob front view


Figure 32, rotating knop side view with spacer visible

Slot Construction:


Figure 33, top view slot construction


Figure 34, side/bottom view slot construction

Figure 35, top view inside slot construction

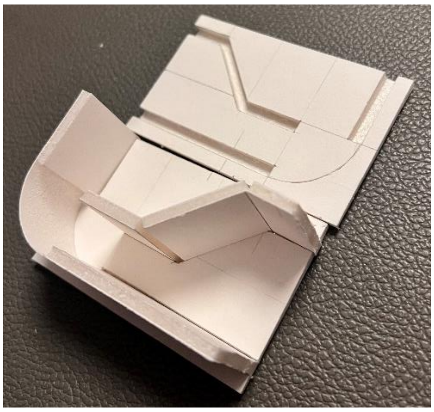


Figure 36, top/side view inside slot construction

Marble Machine:

The marble machine has been constructed mostly with the use of slots *(Figure 43, 44).* It consists of four main parts, the road from the start to the mechanism, the mechanism itself, the road from the mechanism outside and road outside of the box.

The road from the starts to the mechanism is a simple straight piece with a small curve at the start as can be seen in figure 43.

The mechanism is a wall *(Figure 41)* that falls over when the marble hits and transforms into part of the road *(Figure 42)* the marble uses to travel to the end. It is connected to the structure by way of slots seen in Figure 41 to 44.

The road from the mechanism to the outside road is a curve on place on the inside (*Figure 43)*, this curve is connected by a slot on the backside *(Figure 43)*. There is no slot on the front side *(Figure 44)* since this will interfere with the slots for the outside road.

The road outside of the box is connected with a slot place on the outside *(Figure 45,46)*. The side of this road does not stop on the outside, it travels one material thickness into the box as well *(Figure 46)*. This is partially for extra stability of the outside road and partially to create a smooth connection between the out- and inside.

Figure 37, side/front view marble machine

Figure 38, top/side view marble machine

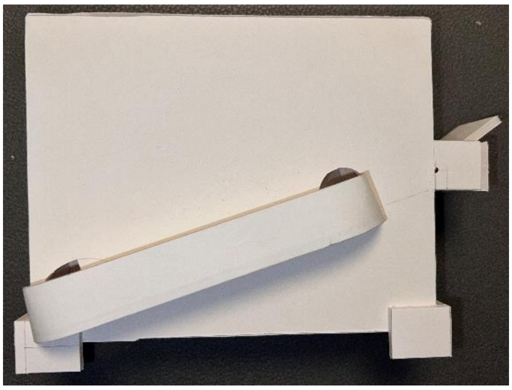


Figure 39, front view marble machine


Figure 40, bottom view Marble machine


Figure 41, mechanism closed

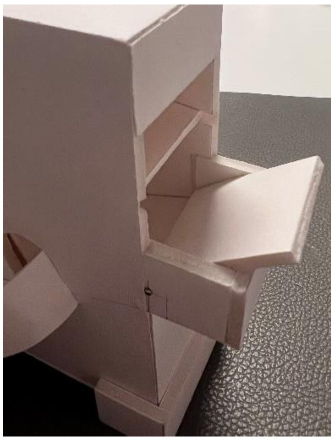


Figure 42, mechanism open

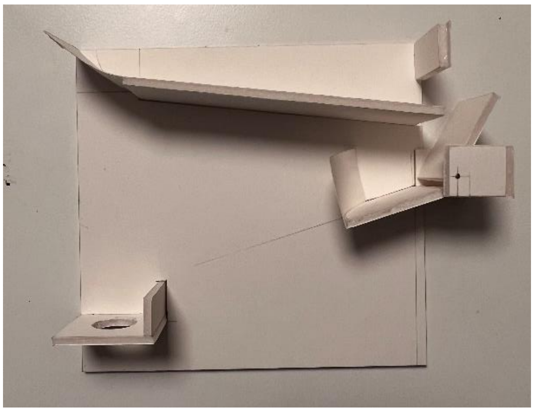


Figure 43, inside marble machine back side

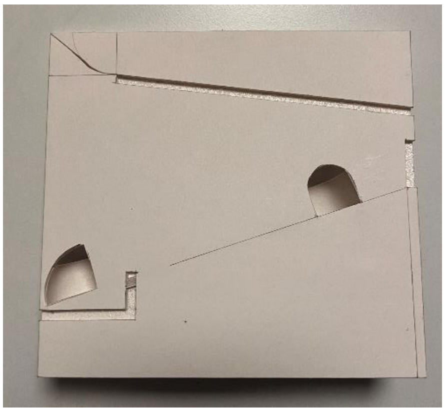


Figure 44, slots inside marble machine front side


Figure 45, outside road marble machine connected to slot


Figure 46, connection outside to the end

Light Controller:

The explorations started of with a square slider mechanism (*Figure 47-49),* I liked the idea of pulling something apart for either the on/off switch or the light intensity. But this particular mechanism did not look good and did not work very smoothly.

I tried the turning knob (Figure 50,51) as another idea for regulating the light intensity. Once I made it, I also saw the potential of selecting a light color by way of this turning knob. However, I did not like the aesthetics as much.

I circled back to the slider idea, by creating a more smoothly working and aesthetically pleasing round slider (Figure 52,53) . It felt satisfying to play around with this design.

With the inspiration from the turning knob idea *(Figure 50,51)* as a color selector, I looked into the idea of a color wheel *(Figure 54-57)*. I liked the idea of choosing and mixing your own colors to create your perfect hue, but the mechanism used here did not work exactly like I wanted to. The knobs kept turning sideways, since the small piece of foamboard in the slots was not sturdy enough.

I sidetracked a bit with the idea of the handheld button *(Figure 58-60)*, since I wanted to look into something that would lay better in your hand and could be used with only one hand. Once I started playing around with the finished product it did feel very easy to use, but too easy in my opinion. I liked having to use both hands in my previous explorations more, since this felt like a more rich interaction to me. Another reason I did not go forward with the handheld idea, was that I knew I needed to create three functions in total. Which makes your options quite limited if want to make all functions possible with one hand in the same position the whole time. If the other two functions don't focused on ease of use, than it takes away the effect of the handheld button as well.

I circled back to the color wheel exploration, by doing a deeper exploration into a better mechanism. Color wheel 2 *(Figure 61-64)*, is made from cardboard instead of foamboard, this material is more sturdy, so this makes sure it does not turn sideways and moves in a straight motion. Another thing that helped with this was the fact that I changed the pieces that went into the slots form a square to a longer rectangle. Color wheel 2, also has larger pieces on the backside so the knobs cannot come off, these backpieces align with the circles perimeter when the knobs are all the way out *(Figure 62).*

I started looking into how to connect different functions together, so I added the round slider (on/off) together with a turning circle (intensity) *(Figure 65, 66)*. The Turning circle is inspired by the turning knob *(Figure 50,51)*. I think the circle looks more aesthetically pleasing. This circle is constructed out of two circles connected by a pin in the middle, the round slider is connected to the bottom circle. The two functions do not interfere with each other and go together well aesthetically.

Since I liked the color wheel idea, I chose to add this to the round slider and turning circle as well *(Figure 67-69)*. The color wheel is still made with the same materials and construction as in color wheel 2, except that the knobs are made out of foamboard this time. The round slider is connected to the cardboard base of the color wheel, so the knobs can still move around freely.

With all the functions constructed together, I decided to add some colors a graphics to the design *(Figure 70)*. For the turning circle used for intensity, it is difficult to figure out which way to turn for more and for less intensity, so I added the drawing in *Figure 73*. For the color wheel, you cannot know which knob is meant for which color, so I gave every knob a primary color, seen in *Figure 72*. When the switch is pulled apart with the round slider, the switch is turned on, as seen in *Figure 71*.

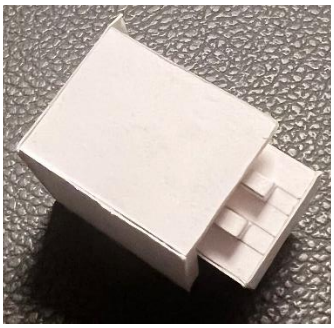


Figure 47, square slider closed

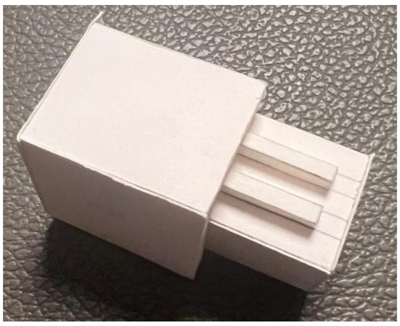


Figure 48, square slider open

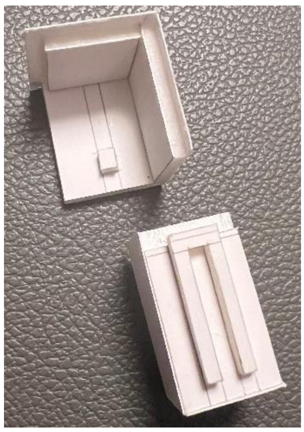


Figure 49, square slider inside

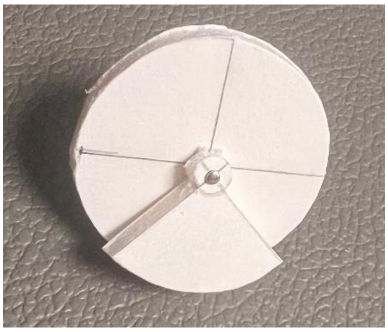


Figure 50, turning knob front view


Figure 51, turning knob side


Figure 52, round slider closed


Figure 53, round slider open


Figure 54, color wheel closed top view

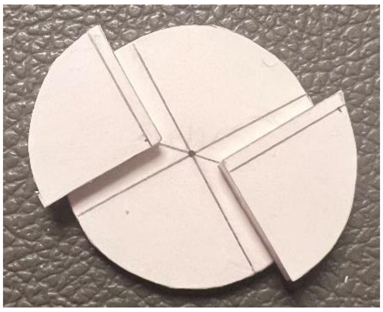


Figure 55, color wheel open top view

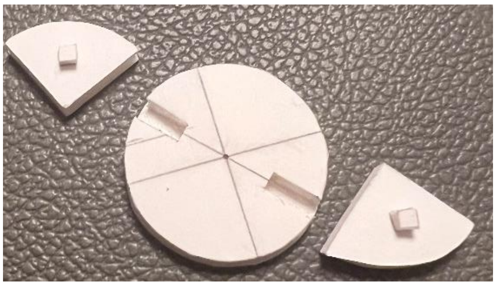


Figure 56, top view color wheel mechanism


Figure 57, side view color wheel

Figure 58, closed handheld button

Figure 59, open handheld button

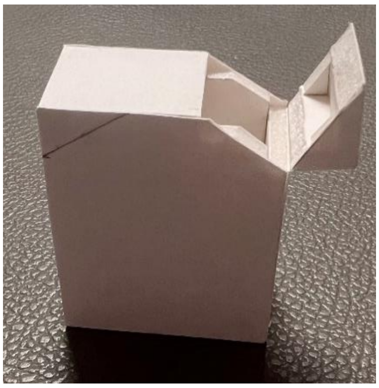


Figure 60, top/side/inside view handheld button

Figure 61, top view color wheel 2, closed

Figure 62, bottom view color wheel 2, open

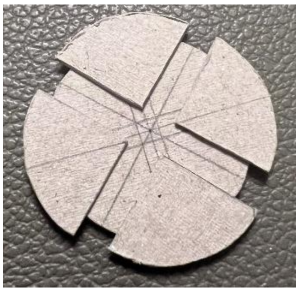


Figure 63, top view color wheel 2, open

Figure 64, side view color wheel 2, closed

Figure 65, side view round slider with turning circle, open

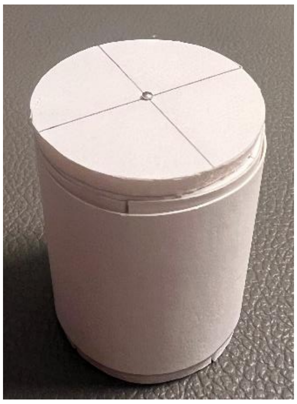


Figure 66, top/ side view round slider with turning circle, closed

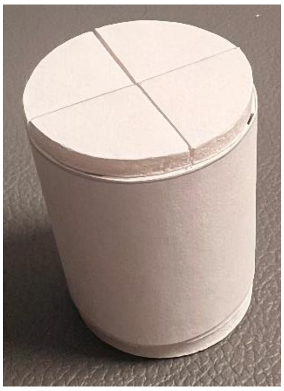


Figure 67, front view light switch , Off

Figure 68, side view light switch, turned on

Figure 69, front view light switch color wheel open.

Figure 70, painting light switch

Figure 71, side/top view final concept , on

Figure 72, top view final concept, off

Figure 73, bottom view final concept, off

Presentation Panel:

As the setting of my presentation panel I chose to create a person hitting a boxing arcade machine. As the interaction part of the design I wanted to add something to turn the machine on. For this interaction I tried out two different ideas. The coin slot idea *(Figure 74,75)*, is a slit in the front panel you can push a coin trough, the coin then falls into a slot behind the front panel where it falls on a button *(Figure 74)*, pressing the button in. The lever idea *(Figure 76,77)*, is lever connected to a potmeter, when the lever is pushed downward, the potmeter will turn with it, changing in value. The potmeter is connected to a piece of foamboard by pressing its pins into the foamboard, this foamboard is connected to the back of the front panel with a slot *(Figure 77)*. From these two ideas I liked the lever more, since the coin slot will give a challenge of how and when to get the coin out of the slot again, thereby releasing the button.

For the motor I already had an idea in my mind of how to use them in my presentation panel. The servo will be controlling the arm hitting the boxing ball, the boxing ball will then move with the stepper motor and then the score will be displayed with the DC motor.

To integrate the servo motor into the design, it needs to be place against the front panel. When I followed the servo motor mount tutorial I soon found out that the top of the servo sticks out of the foamboard quite a bit. This will create a problem if I want to connect the servo securely to the front panel, so I looked into a way to even out this height difference *(Figure 78-80)*. I stacked another piece of foamboard with a cutout for the servo on top of the mount *(Figure 78)*. After that the gear still stuck out, so I added another peace of foamboard on top of that with a round cutout for just the gear *(Figure 78)*, this piece is the front panel in this case. This caused the foamboard to perfectly align with the top of the servo *(Figure 79,80)*.

To integrate the stepper motor I had to look into how to use the stepper motor mount to move a lever perpendicular to the foamboard it is attached to (the base plate). I came up with the idea of a slot in the foamboard to attach the mount to *(Figure 81,82)*. The steppers gear was too far away from the foamboard so I decided to move the stepper as close to base plate as possible. Because of this I had to make a little cutout in the slot to accommodate for the screw holes on the stepper motor *(Figure 81)*. This worked great, but is was not

very sturdy, so I decided to use two stepper motor mounts connected to two slots later on in the development.
To integrate the DC motor I started of with following the motor mount tutorial as well. During this I found out that the mount is made to mover things parallel to its base plate, since I wanted to make something move perpendicular to its base plate, I created a slot in this plate to attach the mount to *(Figure 83)*. In my head I had the idea of presenting the score with a turning cylinder that starts rotating and then stops on a certain score. So I made a rectangle hole in the base plate *(Figure 83)* to accommodate for the cylinder and then I created a cylinder to attach to the DC motor *(Figure 84)*.
The next step was to figure out how to place al the motors and mechanism onto the design so the are positioned correctly and will not interfere with each other. I measured out the approximate sizes the lever and motors would take in, and I cut out pieces of paper these sizes. I played around with the placement of these papers and the sizes of the panels to create a layout that works *(Figure 85)*. After cutting out the panels, I attached the dc motor mount and the servo motor mount including the piece to even out the height difference to the front panel *(Figure 86,87)*. In the top panel I created two slots and a cutout to attach the stepper motor and the boxing ball *(Figure 88,89).* However, once I constructed this, I saw that it would be more appropriate for the boxing ball to move in another way, so I created two new slots and a new cut out 90 degrees form the first slots *(Figure 90,91)*. I also moved the DC motor upside down *(Figure 92,93)*. since it interfered with the servo mount when it was constructed the other way. Lastly I created a slot for the potmeter and a cutout for the lever on the front panel and attached those two to the design *(Figure 92,92)*.
I started my construction of the final design by measuring out the placement for each motor and the potmeter from the mock-up construction *(Figure 92,93)*, tweaking placements or sizes that were a bit of in the mock-up along the way.
For the new DC-motor mount *(Figure 94,95)*, I made sure the motor would be as close to the front panel as possible. After, I constructed the scoreboard. To know the size of the cutout needed for the scoreboard, I measured the length from the DC motor gear to the back of the front panel. I measured the same distance from the middle of the scoreboard and looked at the width at this point. This was width for the cutout *(Figure 95)*. I glued clear plastic piece to attach to the DC's gear to the scoreboard aligning the hole with the middle of the circle *(Figure 94)*.
For the servo I made a new servo mount, making it as small as possible and gluing the mount to the front panel in the same way as in the mockup *(Figure 96,97)*.
I created two slots 5mm apart both in the middle panel and in the top panel. The cutout for the boxing ball is slightly wider than a material thickness to make sure the boxing ball won't get stuck moving trough the cutout *(Figure 98)*.
The potmeter and lever are constructed in the same way as in the mock-up, but based on more precise measurements *(Figure 99,100)*.
150 ohm resistors are connected to the positive wires of the LEDs, with heat shrinks. The wiring to the LEDs is attached with heat shrinks as well *(Figure 101)*. I created three holes next to the lever and inserted the LEDs in them *(Figure 102)*.
With this the design is fully constructed *(Figure 103-108)*. I wired the electronics according to the schematic in Figure 109. I added a power source to the servo and to the motor shield to provide enough power for all the motors.


Figure 74, coin slot top view

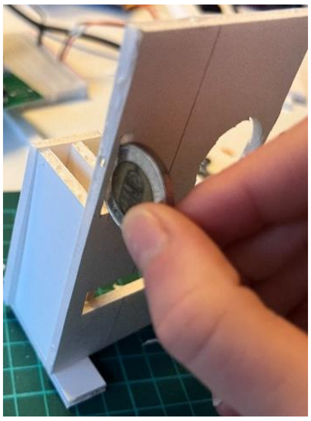


Figure 75, coin slot front view

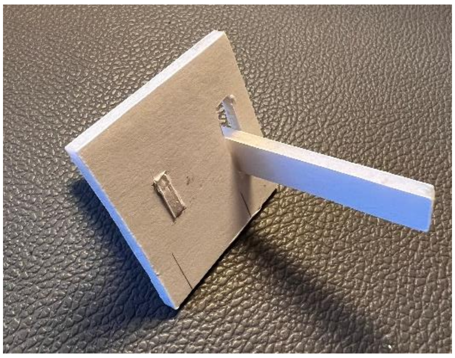


Figure 76, lever front view

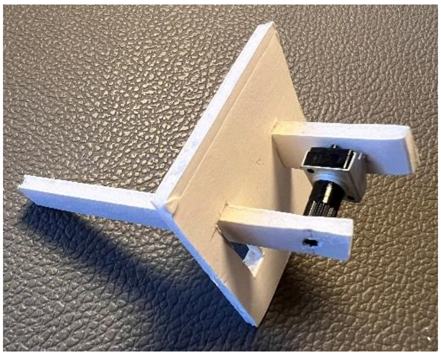


Figure 77, lever back view

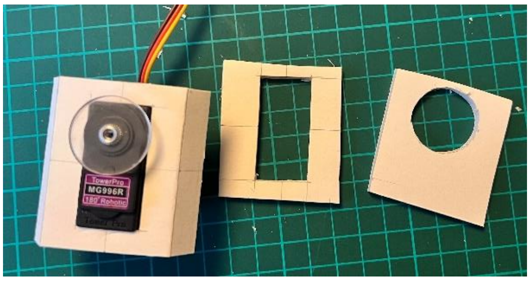


Figure 78, integration servo, separate parts

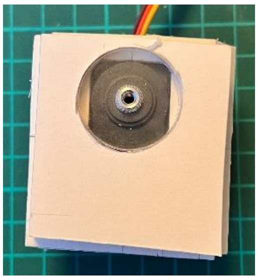


Figure 79, integration servo top view

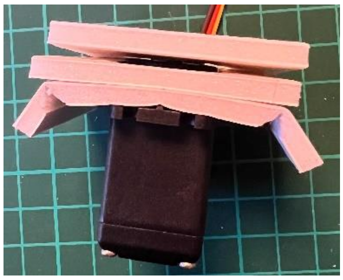


Figure 80, integration servo side view

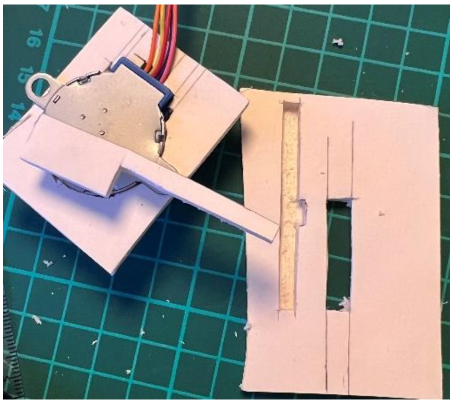


Figure 81, integration stepper, separate parts.

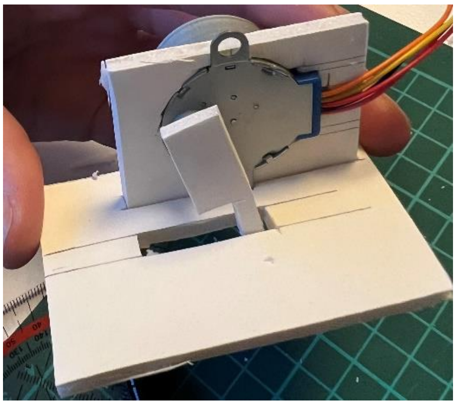


Figure 82, integration stepper, front/top view.

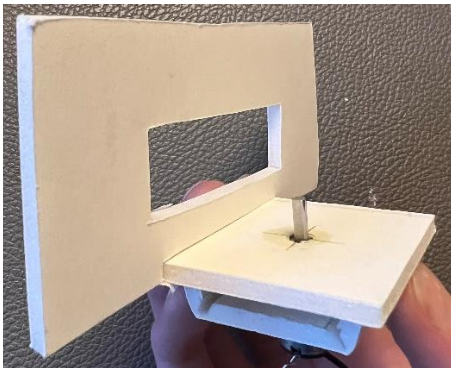


Figure 83, integration DC, top/side view.

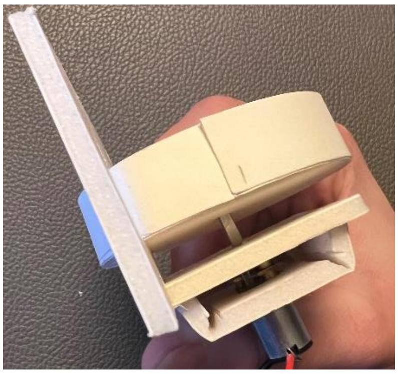


Figure 84, integration DC with rotating circle side view.

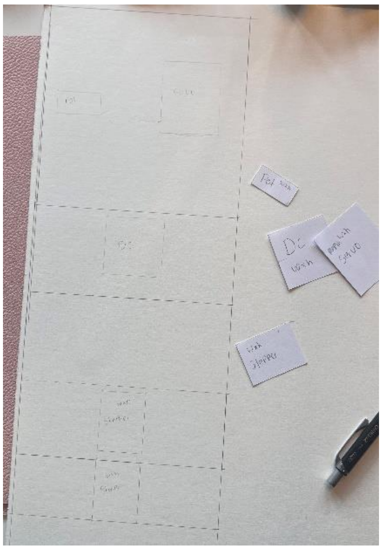


Figure 85, placing motors in design.

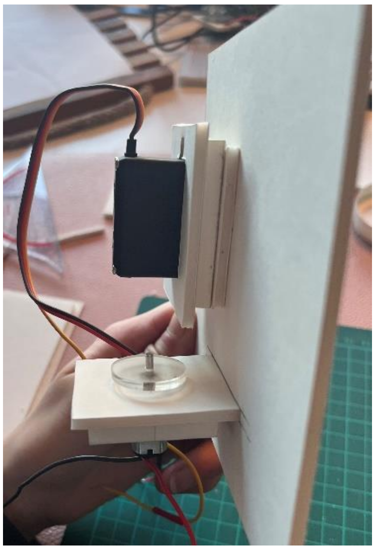


Figure 86, side view servo and DC integrated in design.

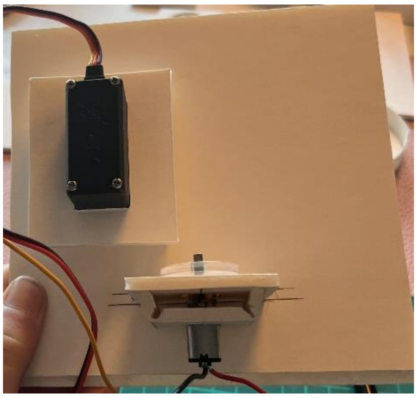


Figure 87, front view servo and DC Integrated in design.

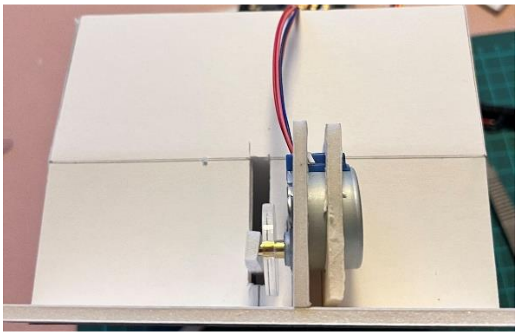


Figure 88, top view stepper integrated in design.

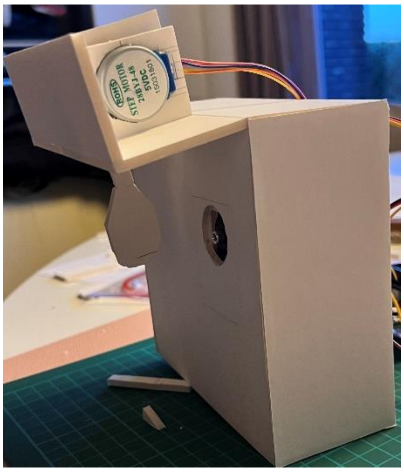


Figure 89, side view stepper integrated in design.

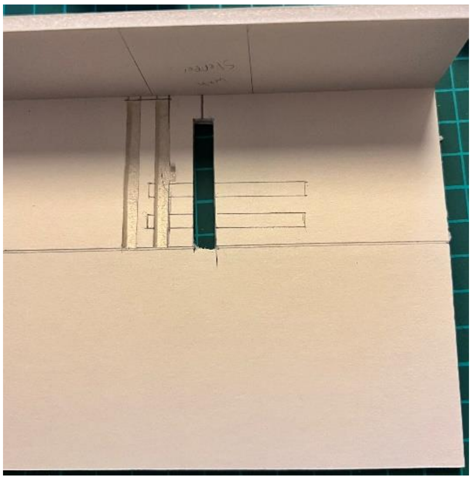


Figure 90, different placing stepper motor.

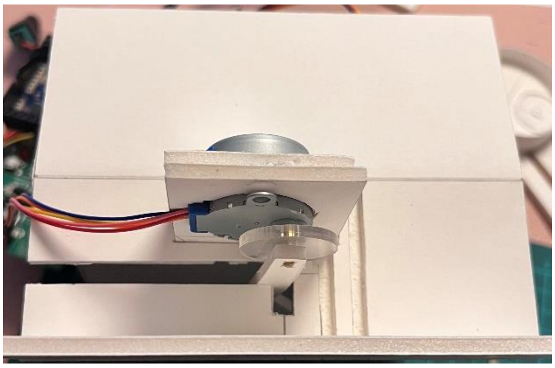


Figure 91, stepper placed in new position.

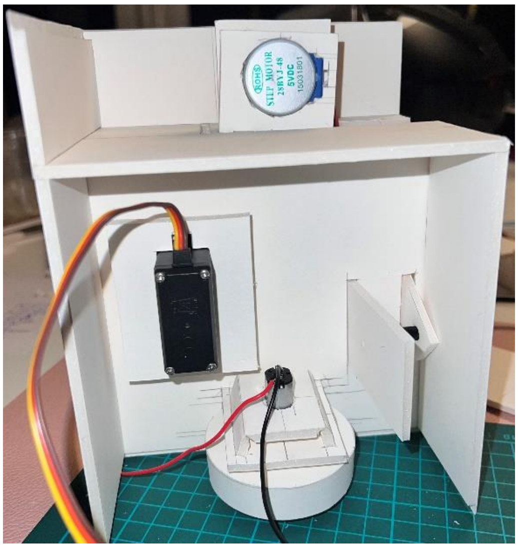


Figure 92, motors and potmeter placement in design ,angle 1.

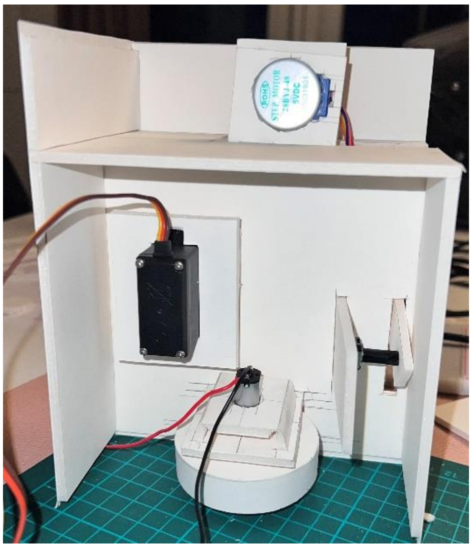


Figure 93, motors and potmeter placement in design , angle 2.

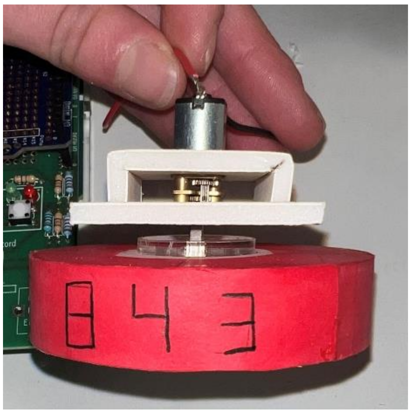


Figure 94, dc motor mount with score wheel.

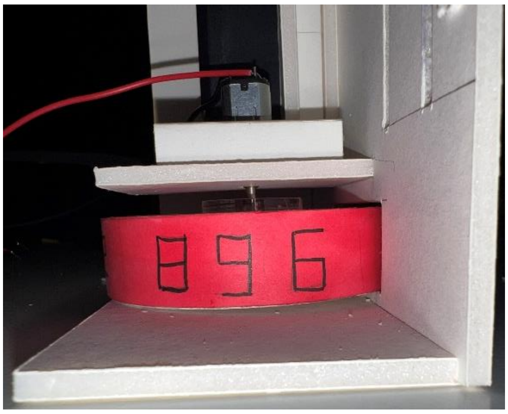


Figure 95, dc motor with score wheel integrated in design.

Figure 96, servo mount glued to front panel.

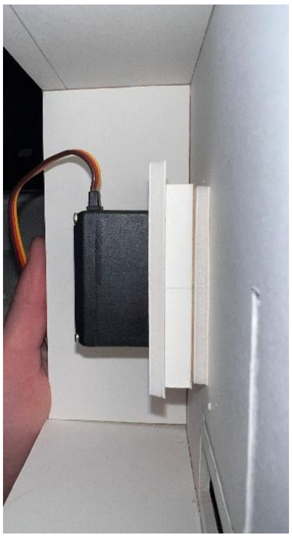


Figure 97, servo connected to servo mount.

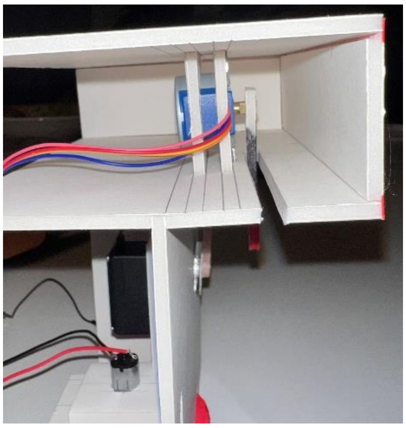


Figure 98, stepper integrated in design.

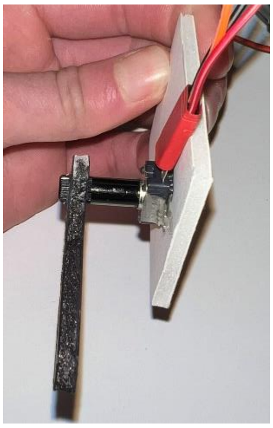


Figure 99, potmeter with lever.

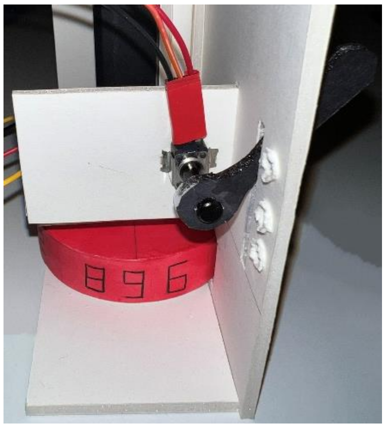


Figure 100, lever integrated in design.

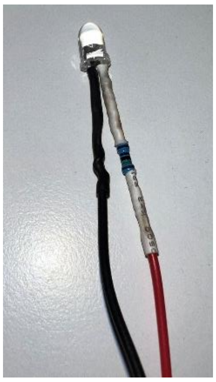


Figure 101, led + resistor.

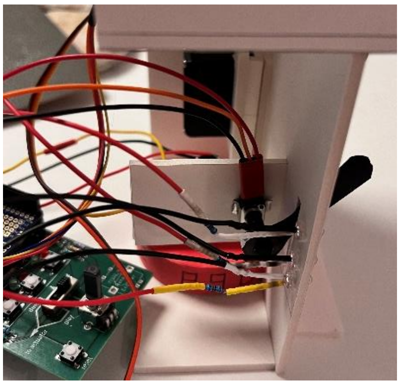


Figure 102, leds integrated in design.

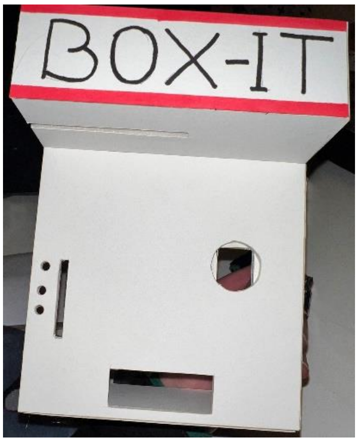


Figure 103, cutouts in the design.

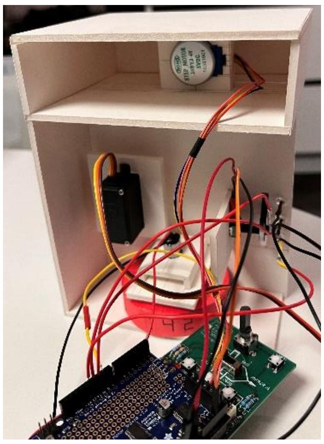


Figure 104, everything integrated in the design (angle 1).

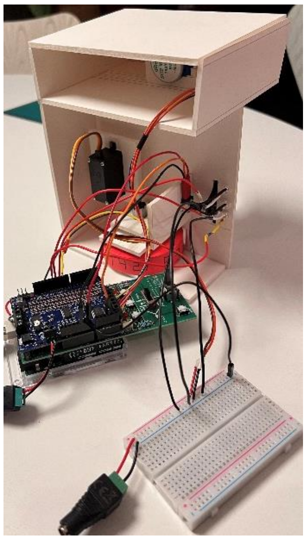


Figure 105, everything integrated in the design (angle 2).

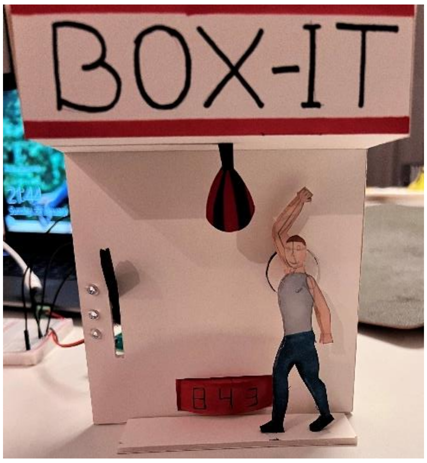


Figure 106, front view finished design.

Figure 107, top/side/front view finished design.

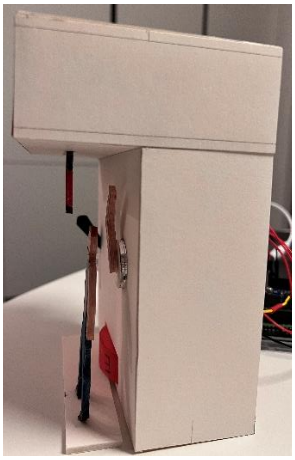


Figure 108, side view finished design.

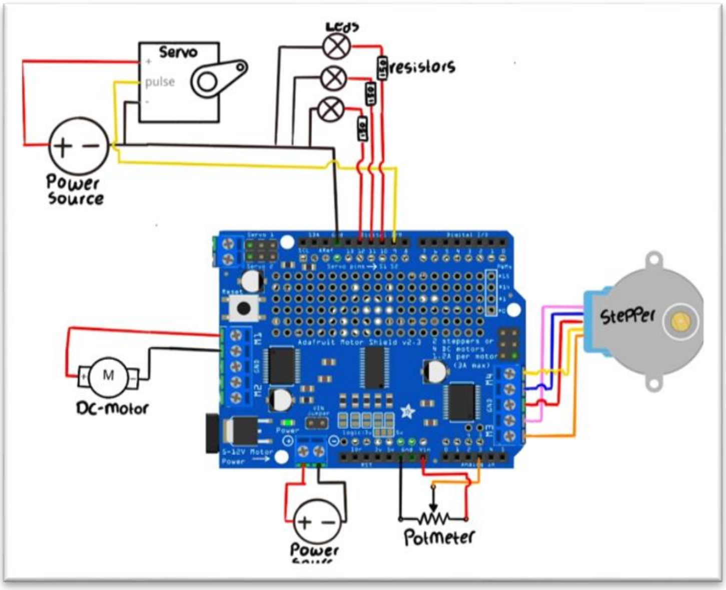


Figure 109, schematic electronics presentation panel

Code presentation panel:

```
//include the necessary libraries

#include <AccelStepper.h>

#include <Servo.h>

#include <Wire.h>

#include <Adafruit_MotorShield.h>

#include "utility/Adafruit_MS_PWMServoDriver.h"


//setup motors

Adafruit_MotorShield AFMS = Adafruit_MotorShield(); //initiate the adafruit motorshield

Adafruit_DCMotor *myMotor = AFMS.getMotor(1); //DC motor is connected to port M1

Adafruit_StepperMotor *myStepper = AFMS.getStepper(16, 2); // Stepper motor is connected to port #2
```

```
Servo myServo;

int modeChoice = 0;
int controlFlag = 0;
int logicControl = 0;

int motionVar = 0;
int arrayPosition = 0;
int DCmotorRuns = 0;

//leds are connected to pin 10,11 en 12
int led1 = 11;
int led2 = 12;
int led3 = 10;

//arrays to make the motors mover correctly
int servoMotionArray_B1[50] = {89, 89, 89, 89, 89, 89, 96, 118, 129, 131, 131, 131, 131, 131, 131, 131, 131, 131, 127, 121,
114, 106, 99, 96, 94, 91, 90, 88, 87, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89, 89};

int StepperMotionArray_B1[50] = {-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-
500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-500,-
500,-500,-500,-500,-500,-500,-500,-500,-500};

int StepperMotionArray_B2[50] =
{50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,
50,50,50,50,50,50,50,50,50,50,50};

int DCMotionArray_B0[50] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

int DCMotionArray_B1[50] = { -10, -12, -34, -81, -137, -160, -166, -166, -166, -166, -166, -166, -166, -166, -166, -166, -166, -
156, -144, -135, -128, -120, -112, -109, -105, -101, -99, -97, -96, -92, -90, -89, -86, -82, -79, -77, -74, -70, -67, -65, -62, -58, -
53, -49, -46, -42, -40, -38, -36, -31};

unsigned long mainTimer = 0; // a variable to time recording and playback
int timeBetweenSamples = 100; // the time between samples.
int loopControl = 0;

int potPin = A3; //potmeter connected to pin A3
int potValue = 0; //value to store potmeter input in

// these are dedicated functions to make the steppermotor move
void forwardstep1() {
```

```
  myStepper->onestep(FORWARD, SINGLE);
}
void backwardstep1() {
  myStepper->onestep(BACKWARD, SINGLE);
}

//create aStepper object
AccelStepper aStepper(forwardstep1, backwardstep1); // use functions to step



void setup() {
  Serial.begin(9600);
  AFMS.begin();
  myServo.attach(9); // servo is connected to pin 9
  myMotor->setSpeed(0);
  myMotor->run(RELEASE);



  //setup the stepper
  aStepper.setSpeed(0); // this is used in the 'speed' implementation
  aStepper.setMaxSpeed(200.0); // these are used in the 'position' implementation
  aStepper.setAcceleration(100.0);
  aStepper.moveTo(0);

//initialize the led pins as output
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
  delay(timeBetweenSamples); //bruteforce hacky solution for startup weirdness
}

void checkSerial() {
  potValue = analogRead(potPin); //read the input from potmeter
  if (Serial.available() > 0) {
    int modeChoice = Serial.read();
    if (modeChoice == 49) { //type 1 to start the switch function
      arrayPosition = 0;    // reset the arrayposition to start the motion from 0
      controlFlag = 0;      //go to case 0
    }
  }
```

```
}


void loop()
{
  //write down potmeter value in serial monitor
  Serial.print(potValue);
  Serial.print("= potmeter");
  Serial.println();

  if (mainTimer < millis() - timeBetweenSamples) { // playing only once per interval. // has 100 ms gone past
    mainTimer = millis(); // time is rewritten to current millis starting a new interval.

    logicControl = controlFlag;

    if (controlFlag != 2 && DCmotorRuns == 1) { // release the DC motor when not in 'case 2' and if the motor runs
      myMotor->run(RELEASE);
      DCmotorRuns = 0;
    }


    switch (logicControl) {

      case 0: //servo makes the arm punch
        if (potValue < 850) { //if the potmeter lever is turned on..
          motionVar = servoMotionArray_B1[arrayPosition];
          myServo.write(motionVar);
          arrayPosition++;
          Serial.print(arrayPosition);
          Serial.print(",");
          Serial.println(1);
          if (arrayPosition >= 50) { //if array position 50 is reached
            arrayPosition = 0;      // go back to array position 0
            controlFlag = 1;        //go to case 1
          }
          break;
        }
        else if (potValue > 850) {   //if the potmeter lever is not turned on, don't do anything


        }
```

```
case 1: //stepper, punchball goes up
 if (potValue < 850) {        //if the potmeter lever is turned on..
   digitalWrite(led1, HIGH);   //turn leds on
   digitalWrite(led2, HIGH);
   digitalWrite(led3, HIGH);
   motionVar = StepperMotionArray_B1[arrayPosition];
   aStepper.moveTo(motionVar); //using stepper position
   arrayPosition++;
   Serial.print(arrayPosition);
   Serial.print(",");
   Serial.println(2);
   if (arrayPosition >= 50) { //if array position 50 is reached
     arrayPosition = 0;       // go back to array position 0
     controlFlag = 2;         //go to case 2
   }
   break;
 }
 else if (potValue > 850) { //if the potmeter lever is not turned on, don't do anything

 }

case 2: //to let the DC wait for the boxbal to fully go up, without it it starts turning while the stepper is still working
 if (potValue < 850) {                       //if the potmeter lever is turned on..
   motionVar = DCMotionArray_B0[arrayPosition]; //contains zero's so motor stays still

   //decide which side the DC motor rotates to
   if (motionVar < 0) {
     myMotor->run(BACKWARD);
     myMotor->setSpeed(abs(motionVar));
   }
   else if (motionVar >= 0) {
     myMotor->run(FORWARD);
     myMotor->setSpeed(motionVar);
   }
   arrayPosition++;
   Serial.print(arrayPosition);
   Serial.print(",");
   Serial.println(3);
```

```
    if (arrayPosition > 49) { //if array position 50 is reached
      arrayPosition = 0;      // go back to array position 0
      controlFlag = 3;        // go to case 3
      myMotor->run(RELEASE);  //release DC motor to make it stop rotating
      DCmotorRuns = 0;
    }
    break;
  }
  else if (potValue > 850) { //if the potmeter lever is not turned on, don't do anything

  }

case 3: //DC motor presents score
  if (potValue < 850) {           //if the potmeter lever is turned on..
    motionVar = DCMotionArray_B1[arrayPosition];
    //decide which side DC motor rotates to
    if (motionVar < 0) {
      myMotor->run(BACKWARD);
      myMotor->setSpeed(abs(motionVar));
    }
    else if (motionVar >= 0) {
      myMotor->run(FORWARD);
      myMotor->setSpeed(motionVar);
    }
    arrayPosition++;
    Serial.print(arrayPosition);
    Serial.print(",");
    Serial.println(3);

    if (arrayPosition > 49) { //if arrayposition 50 is reached
      arrayPosition = 0;      // go back to array position 0
      controlFlag = 4;        //go to case 4
      myMotor->run(RELEASE);  //release DC motor to make it stop rotating
      DCmotorRuns = 0;
    }
    break;
  }
  else if (potValue > 850) { //if the potmeter lever is not turned on, don't do anything

  }
```

```
    case 4: //stepper brings boxball down again
      if (potValue < 850) {          //if the potmeter lever is turned on..
        //turn the leds off
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(led3, LOW);
        motionVar = StepperMotionArray_B2[arrayPosition];
        aStepper.moveTo(motionVar);  //using stepper position
        arrayPosition++;
        Serial.print(arrayPosition);
        Serial.print(",");
        Serial.println(2);
        if (arrayPosition >= 50) { //if array position 50 is reached
          arrayPosition = 0;      // go back to array position 0
          DCmotorRuns = 0;
          controlFlag = 0;        // go back to case 0
        }
        break;
      }
      else if (potValue > 850) { //if the potmeter lever is not turned on, don't do anything

      }
    }




  } // end timer loop
  checkSerial(); //using serial monitor instead of buttons




  //if you run stepper motors you need to uncomment one of the following instructions:
  aStepper.run();      //stepper position implementation
  aStepper.runSpeed(); //stepper speed implementation

}
```

Hot Beverage Machine Explorations:

I started of with exploring different inputs, the first one being a on/off pullout switch *(Figure 110,111)*, her you pull the cupholder out of the machine to turn it on. For a customization of the drink I tought of creating a lever that can be moved up and down to choose the intensity of your drink *(Figure 112,113)*. For another type of customizing your drink I tought of turning the cupholder itself *(Figure 114)*, once I made this is also saw the potential turning something to select your type of drink. As a derivative of the turning cupholder I made something to turn on the cupholder instead of the cupholder itself *(Figure 115,116)*, but I did not like this idea as much since it creates a visible knob while the turning cupholder does not.
I liked the idea of a up and down lever to use for intensity *(Figure 112,113)*, but the mechanism itself did not justify the vision I had so I did a deeper exploration into this concept by creating a new type of customization lever *(Figure 117,118)*, this felt way more smooth to use and created a more elegant type of motion.
For the last input idea I tried to work with horizontal and vertical motions since I felt like I had mostly explored rotary motions before this point, with the use of slots I created a rod that can move vertically and horizontally *(Figure 119,120)*. This idea did not work in the way I hoped to, it felt like an uncomfortable motion to manoeuvre the rod in this horizontal and vertical direction.

I started the output exploration with a simple idea of the cupholder appearing out from the machine *(Figure 121,122)*. This would look good aesthetically, in my opinion. I kept focused on the cupholder in my second exploratory idea, where I made the cupholder itself rotatable to stir the drink for you *(Figure 123)*. When playing around with this model, I discovered I could also move the cupholder up and down, which gave me the idea of presenting the finished beverage by lifting it up from a platform for example *(Figure 123,124)*.
Moving away from the cupholder, I focused on the drink dispenser. In my mind I had the idea that the liquid could maybe go down a slide to fill the cup, so I created a slide that can rotate down to dispense the drink and then rotate back up again *(Figure 125,126)*.
I also created a mock-up cupholder, just to see how I can make a good cupholder, and to maybe trigger some new ideas, by playing with the cupholder. No extra ideas came out of this model, but I used the model later on to visualise different ideas more in depth.

For the input I liked the up and down lever a lot and the on/off pullout cupholder, so I decided to combine the two to see if they can be intergrated into one design *(Figure 128)*. Once I constructed this model I starte thinking of a way to add a way of choosing your drink into this concept. Since I liked the rotation of the cupholder as a possible option in the exploration, I added a rotator on the bottom of the on/off pullout base *(Figure 129)*. Now I could turn the machine on, choose my drink and customize it, all by just moving the cupholder *(Figure 128-131)*, which I really liked about this combination. When I played around with the movements of this combination, it almost felt like a game controller, so came up with the idea of implementing a controller type of handle to attach to the cupholder *(Figure 135,136)*.
After this I looked back into the outputs, and came up with a new concept, which is two types of output in one. A door that opens up and then the dispenser slides horizontaly trough the door *(Figure 132-134)*. I liked how this makes the dispenser not visible when it is not used, and it made me think of a cuckoo clock, which for some reason, I liked. I kept the rotating cupholder to mix your drink idea as well, to conclude the outputs.
Eventhough I liked the controller handle, it was not practical or aesthetics pleasing to attach to the cupholder, so I tried to combine the handle into the cupholder *(Figure 137,138)*, which is way more practical and looks better as well.
After I focused on the horizontal movement of the dispenser as output, since I had only rotary motors I created a slider crank mechanism *(Figure 139,140)*. The idea worked correctly, but I did find that I needed to change something so the slider would stay in its slot.

The last exploratory phase was finding ways to integrate the motors and sensors into the design mechanisms.
I started off easy with the rotating cupholder. By making a DC mount in the base the cupholder sits on (so the top of the controller), so the DC motor pin sticks up straight in the middle of the controller and you can attach a holder to that pin *(Figure 141-143)*.
To sense the horizontal rotation of the handle I used a potmeter. First I created a potmeter mount by outlining its perimeter *(Figure 144)*. Since the potmeter is quite high , I layerd a second plate of foamboard on top of the one with the servo mount with spacer of 4 material thicknesses high inbetween. This second plate has a small hole for the potmeter to come trough but not so small that it touches the plate *(Figure 145)*. On top of this secon plate I layered the actual rotating piece, this piece has a small hole in the middle to grab onto the potmeters handel *(Figure 146)*.
For the vertical intensity lever *(Figure 147-149)*, I also used potmeters, these are also attached by the same potmeter mount but this time I only cut to the paper layer and not all the way trough. The reason for this is that

the box these potmeters are attached to also has to act as a slider for the on/off pullout mechanism so the outsides should be as smooth as possible. The the is one extra piece of foamboard inbetween the mount and the lever, this piece has a hole snug enough to keep the servo in place bit big enough so I won't grab onto it *(Figure 147-149)*.
As I stated previously, the box the lever is attached to is also the slider for the on/off pullout mechanisme. For this mechanism a added a button on the backside of the rails, so it will be pressed once fully pushed in and won't be pressed once pulled out *(Figure 150,151)*. The problem with this is that the button needs to much force to stay pressed, so this did not work. However I came up with a new idea with a slider switch to use instead, which will be show in the final concept.
For the door, I used a stepper motor. I created a small box on the backside of the door *(Figure 152)*, to attach the steppers pin to the stepper itself was attached by a stepper motor mount *(Figure 152,153)*.
For the horizontal slider dispensers, I used a servo motor put in place with a servo mount inserted in a slot. A rod is connected to the servo with a pin and it is also connected to the slider *(Figure 154,155)*. I also solved the issue of the slider not staying in its slot as visible in *Figure 156*.

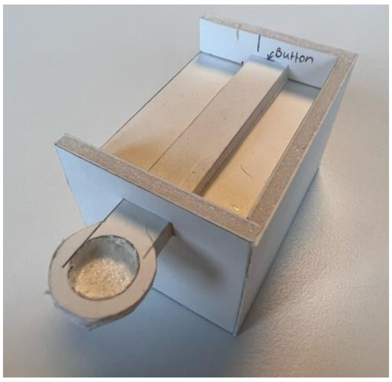


Figure 110, on/off pullout: pushed in

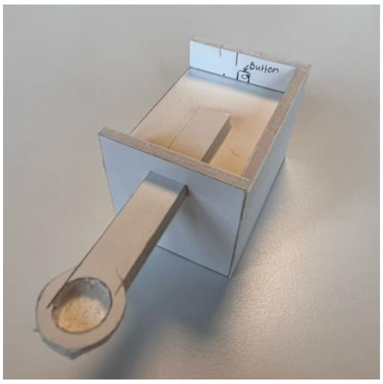


Figure 111, on/off pullout, pulled out

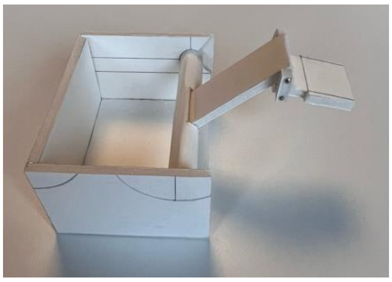


Figure 112, customization rotator (1)

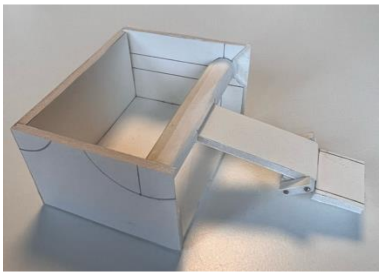


Figure 113, customization rotator (2)

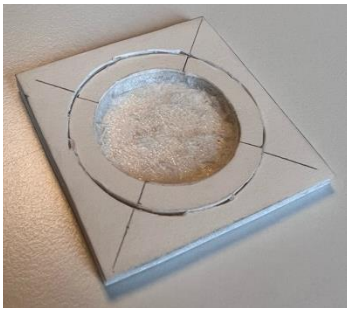


Figure 114, cupholder rotator

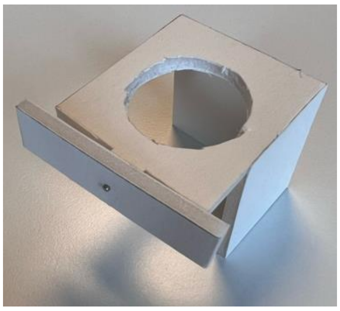


Figure 115, turning knob cupholder(1)

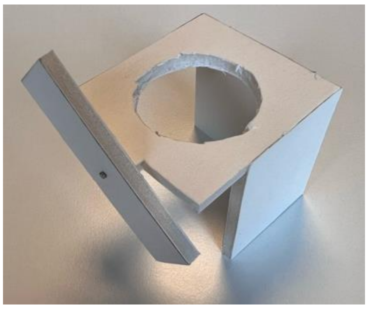


Figure 116, turning knob cupholder (2)

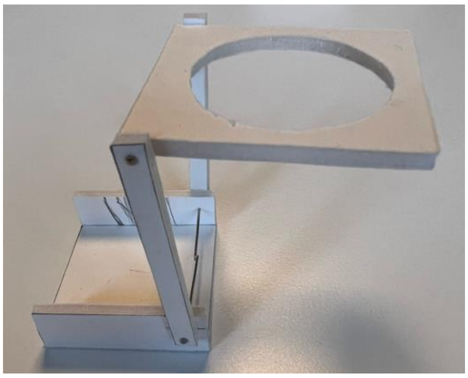


Figure 117, second customization rotator (1)

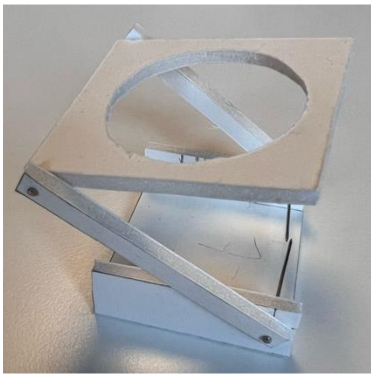


Figure 118, second customization rotator (2)

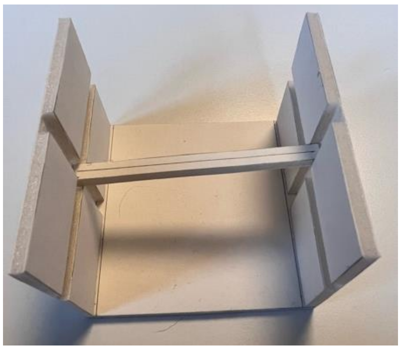


Figure 119, vertical/horizontal customization (1)

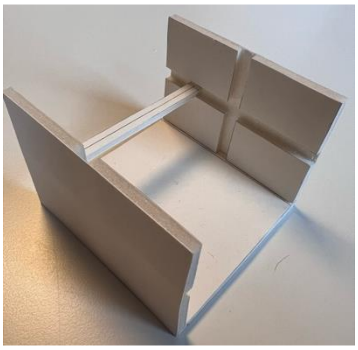


Figure 120, vertical/horizontal customization (2)

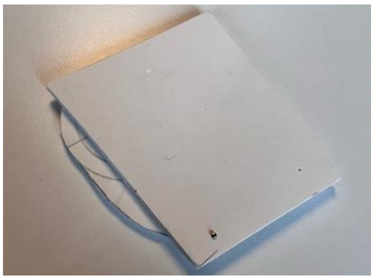


Figure 121, pullout cupholder (1)

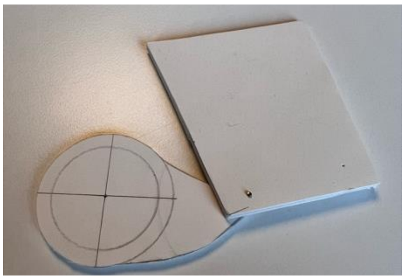


Figure 122, pullout cupholder (2)

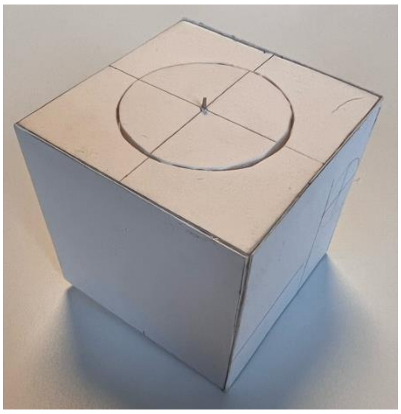


Figure 123, raising/turning cupholder low

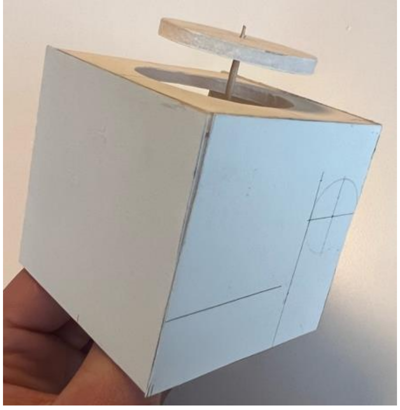


Figure 124, raising/turning cupholder high

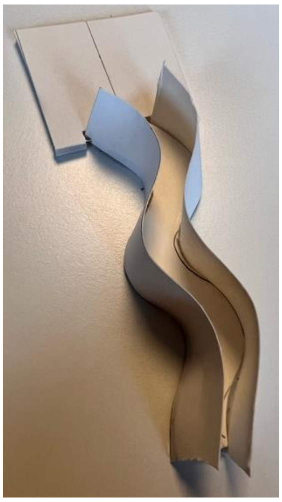



Figure 125, slide dispenser up Figure 126, slide dispenser down

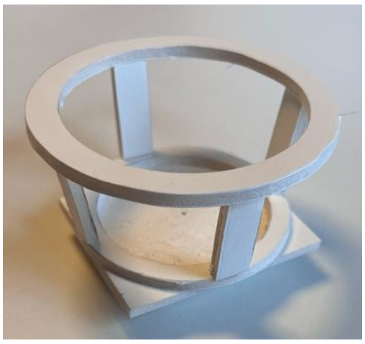
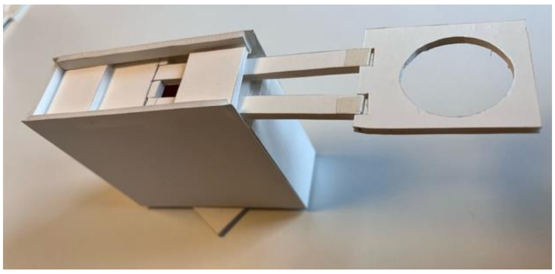



Figure 127, cupholder Figure 128, input combined

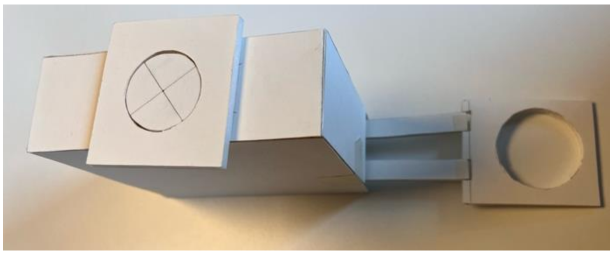
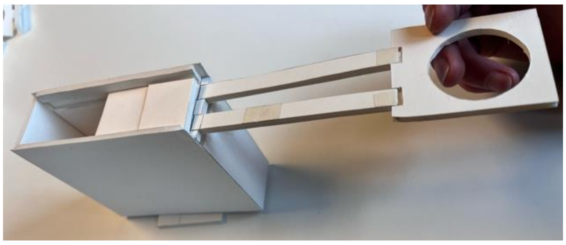



Figure 129, rotator on bottom (input combined) Figure 130, pullout on/off (input combined)

Figure 131, customization rotator (input combined)

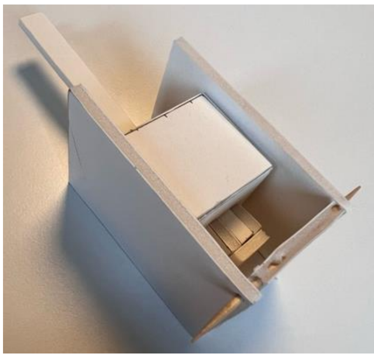


Figure 132, output combined

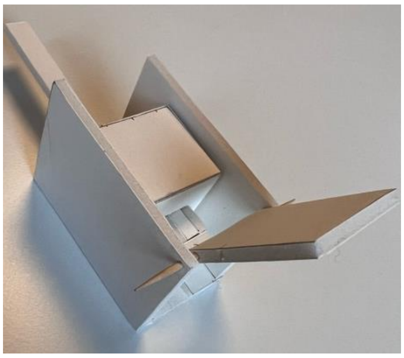


Figure 133, door open (output combined)

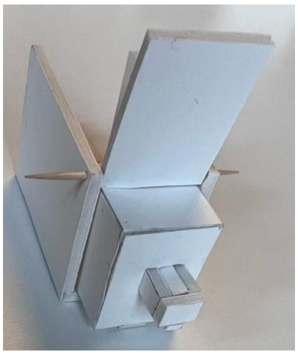


Figure 134, dispense slider out (output combined)


Figure 135, controller (1)

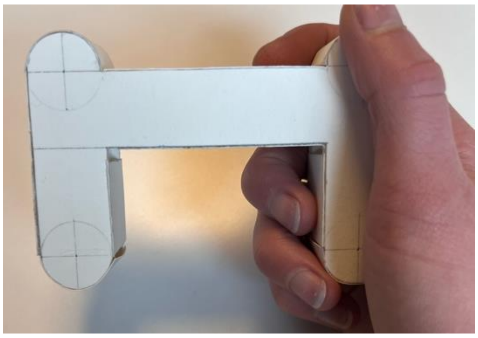


Figure 136, controller (2)

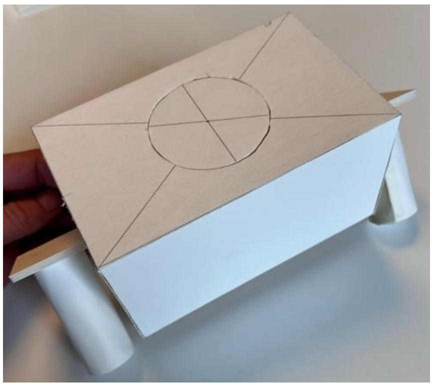


Figure 137, cupholder controller combined top

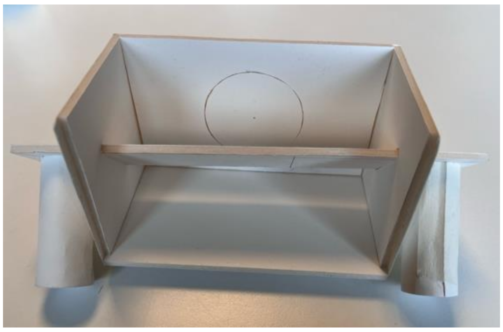


Figure 138, cupholder controller combined bottom

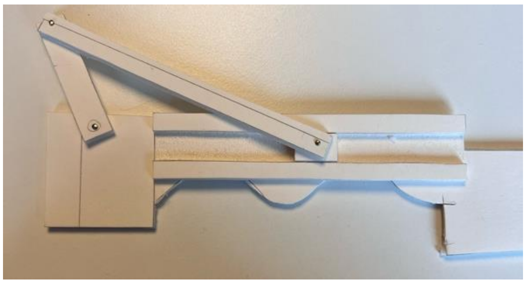


Figure 139, rotary to horizontal movement (in)

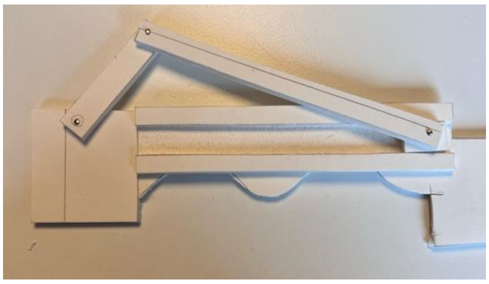


Figure 140, rotary to horizontal movement (out)

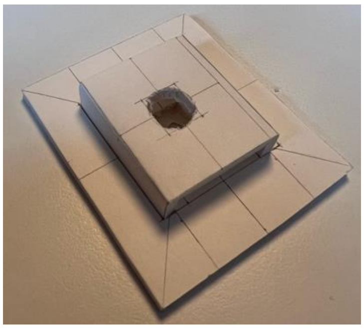


Figure 141, DC mount bottom cupholder

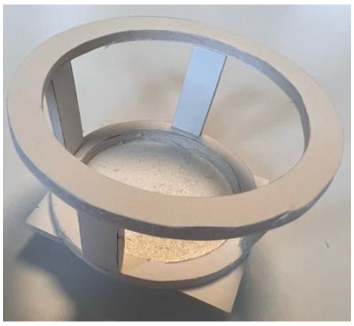


Figure 142, DC motor cupholder top

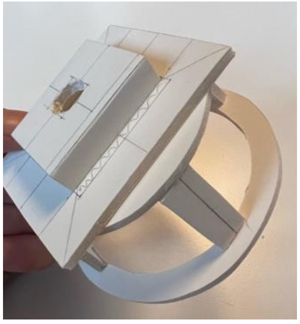


Figure 143, DC motor cupholder side

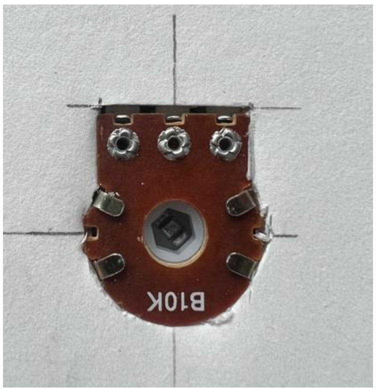


Figure 144, potmeter mount bottom view rotator

Figure 145, potmeter top view rotator

Figure 146, top view rotator assembled

Figure 147, customization rotator with potmeters

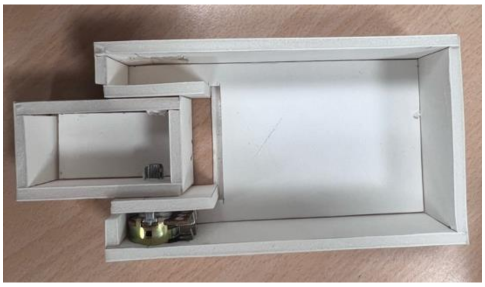


Figure 148, customization rotator with potmeter assembled

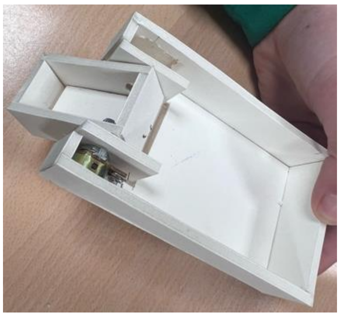


Figure 149, customization rotator rotated

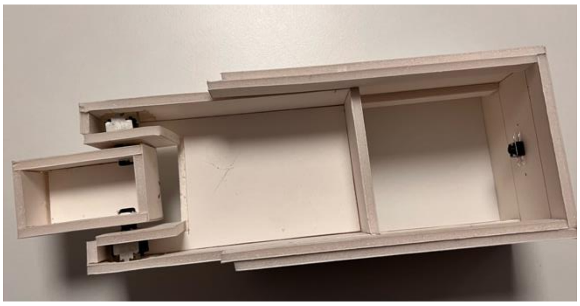


Figure 150, cust. Rotator in on/off pullout top view

Figure 151, cust. Rotator in on/off pullout bottom view

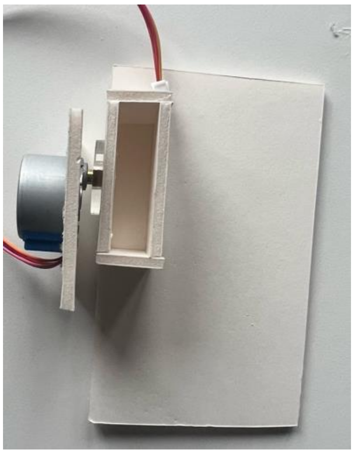


Figure 152, door stepper motor back view


Figure 153, door stepper half open, side/front view


Figure 154, servo rotary to horizontal for slide out dispenser side view

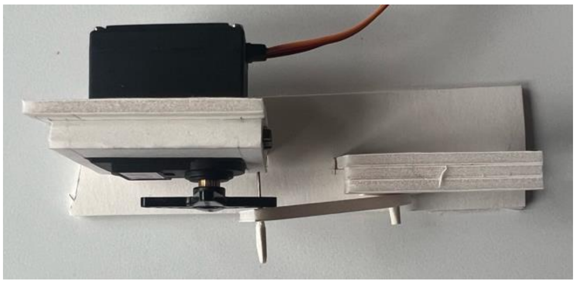


Figure 155, servo rotary to horizontal for slide out dispenser Top view

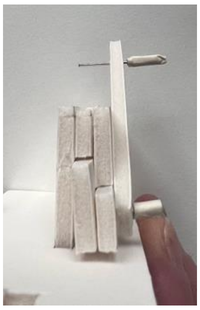


Figure 156, rotary to horizontal fro slide out dispenser front view

Hot Beverage Machine Final Design:

The final design consists of multiple layers, layer one is the bottom, it contains a potmeter mount to secure the potmeter in place and spacers to bridge the height of the potmeter (*Figure 157,158*).
Layer 2 contains a small hole in the middle to make room for the potmeter (*Figure 159,160*), layer 2 is slightly smaller than layer 1.
Layer 3 is the same size as layer 2 and is connected to the potmeter, so when layer 3 turns, the potmeter turns as-well. Layer 3 also, contains the input systems. The first part of this system is the controller. It contains a button to press when you have selected your choice and a DC motor to create the mixing the drink output (*Figure 161,162*). The lever (*Figure 166*) is connected to the controller with a hinge, the lever is also connected to the slider with two potmeters. The potmeters that connect the lever and the slider (*Figure 166*), act as a hinge while also recording how the lever is turned. The slider (*Figure 165*), keeps the potmeters in place by a potmeter mount on each side and a small hole in the pieces of foam board between the lever and the potmeter on each side. Besides the potmeters, the slider contains a cutout right in the middle, this cutout is to control a switch placed in the base. This switch decides wether the machine is on or off. The controller, lever, and slider connected are visible in *Figure 163 and 164*. The slider is connected to the base (*Figure 167-170*). The switch is connected in the middle of the base, directly under the slider, when the slider is pulled all the way forward, the switch will go forward as well. When the slider is pushed all the way back in, the switch will go backward. The switch also has a second function; it prevents the slider from being pulled all the way out of the base.
When I pulled the slider out during construction, the whole third layer lifted up in the air, because layer three is

only connected to a potmeter and cannot provide enough stability on its own.
To prevent layer 3 from lifting up, layer 4 and 5 have been installed (*Figure 171,172*).
Layer 4 (*Figure 171*) is the same size as layer 2 and 3 and is the top of the rotating part of the machine. The only function layer 4 has, is to provide layer 5 with something to push down on.
Layer 5 is half a circle the same size as layer 1 *(Figure 172)*, it is connected to the sidewalls and not to layer 4. This means that layer 5 does not turn, it is there to provide a roof over layer 4.
Layers 6 is the same size as layer 1 and 5, on layer 6, all the servos are placed (*Figure 173*). These servos control the drink dispensers by moving them forwards or backwards. The servos are connected to layer 5 by a servomotor mount placed in slots and the mechanism is created the same way as in the exploration of *Figure 154, 155*. The only difference is that the rod is now connected to the servo with a screw, since this provides more stability than a pin.
Layer 7 and 8 are the last two layers, they contain a stepper mount, this stepper is there to control the door (*Figure 174*). While the door (*Figure 175,176)* always moved vertically in the explorations, here it moves horizontally. The reason for this is that it was too difficult to implement the upward motion in a circular design, the stepper faces downward and creates a door-hinge. The small circular knob on the top of the design is there to provide enough space for the stepper motor (Figure 174).
For some of the wiring I created longer horizontal cutouts in the back of the machine (Figure 180), this is there because these wires are connected to the rotating part and the wires need room to rotate as-well.
The full design is visible in *Figure 177-182*, the wiring circuitry schematic is visible in *Figure 183*.

Figure 157, bottom layer top view

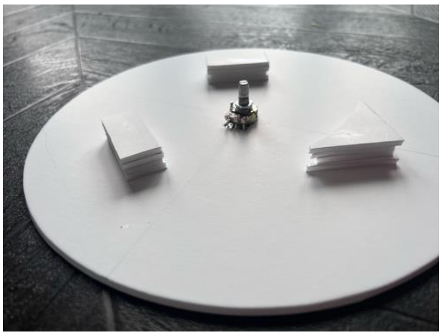


Figure 158, bottom layer side view

Figure 159, second layer top view

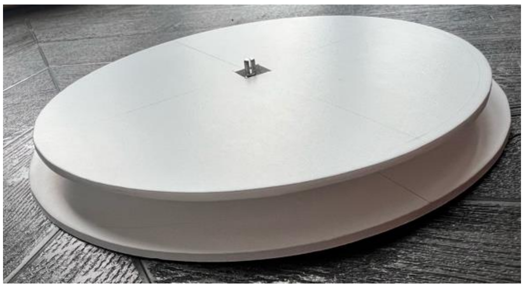


Figure 160, second layer side view

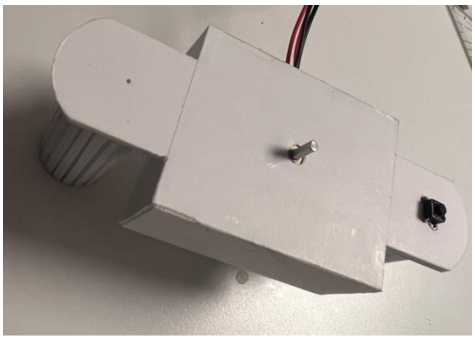


Figure 161, controller with DC and button top view.

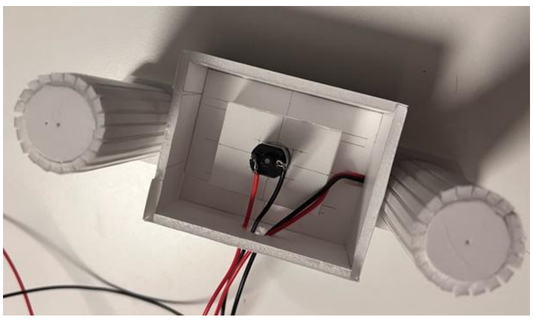


Figure 162, controller with DC and button bottom view

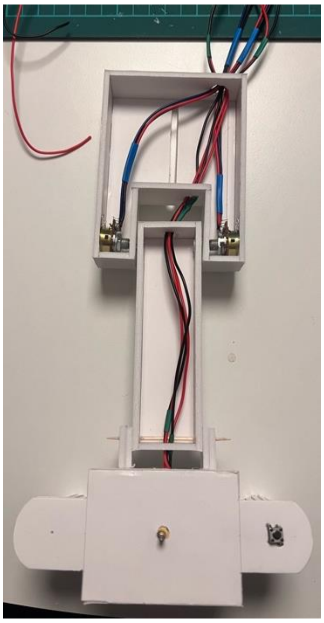


Figure 163, controller with lever And slider top view

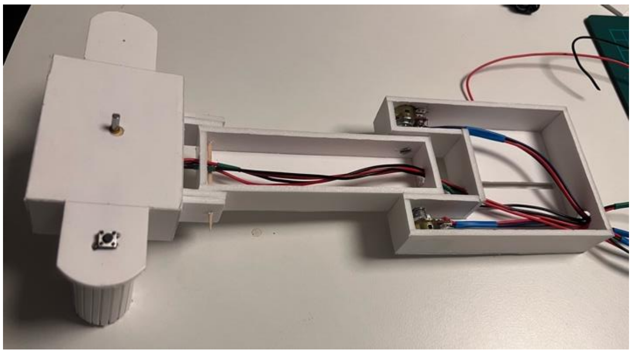


Figure 164, side view controller, lever and slider connected

Figure 165, slider with wiring and. Potmeters

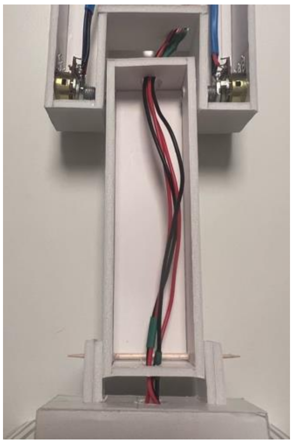
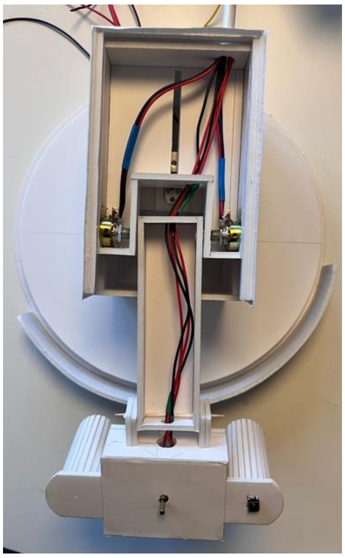


Figure 166, lever with wiring , hinge and potmeters



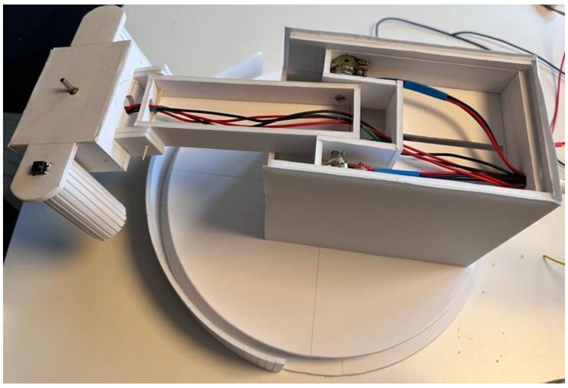
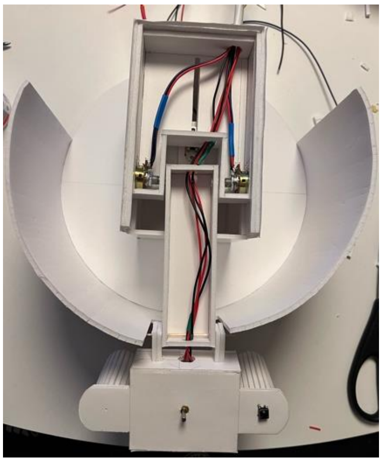


Figure 167, slider connected to base
On layer three

Figure 168, side view slider connected to base on layer three



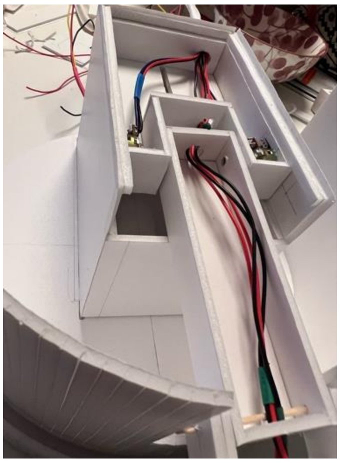
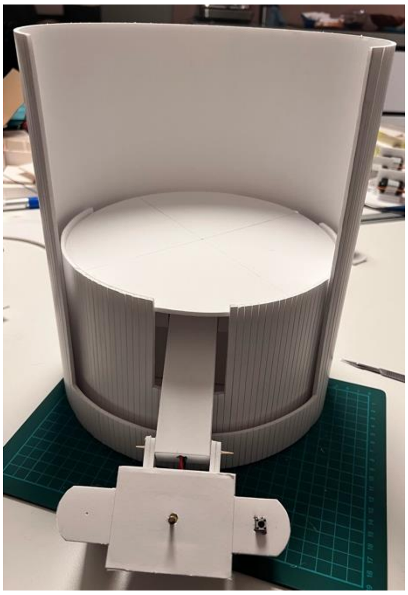


Figure 169, walls around turning part

Figure 170, side view slider, lever and base connection



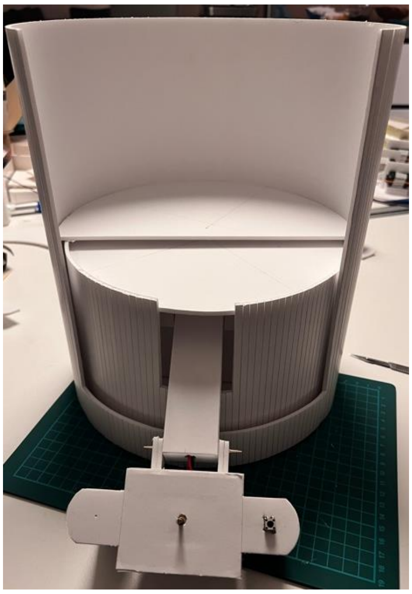


Figure 171, layer 4 = top of rotating part.

Figure 172, layer 5 half layer to keep rotating part down

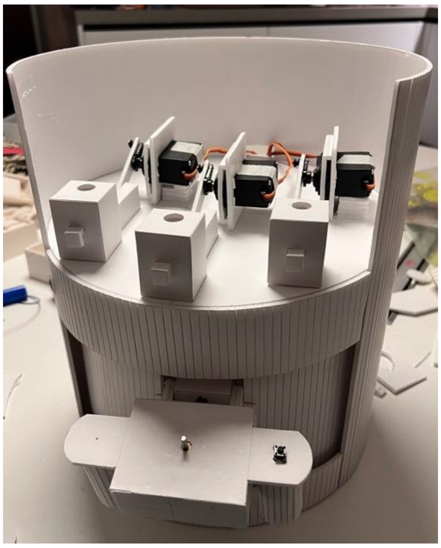


Figure 173, layer 6, servo dispensers


Figure 174, layer 7 & 8, stepper mount for door and top layer


Figure 175, top view door.


Figure 176, front view door

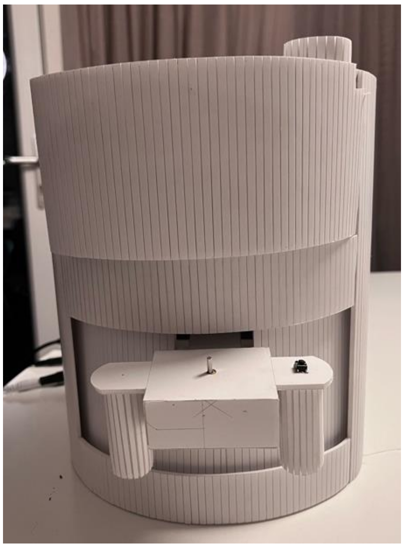


Figure 177, finished beverage machine (1).


Figure 178, finished beverage machin (2).

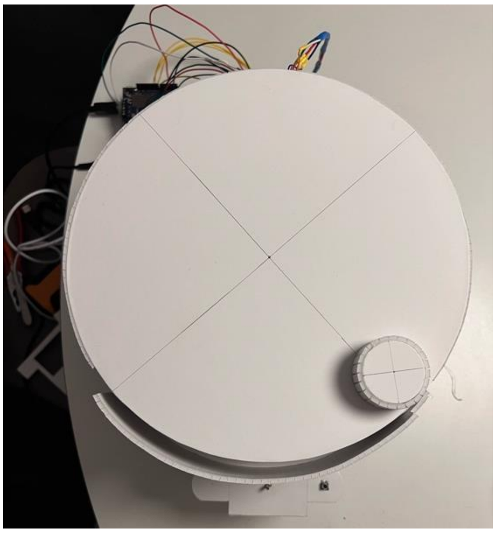


Figure 179, finished beverage machine (3)

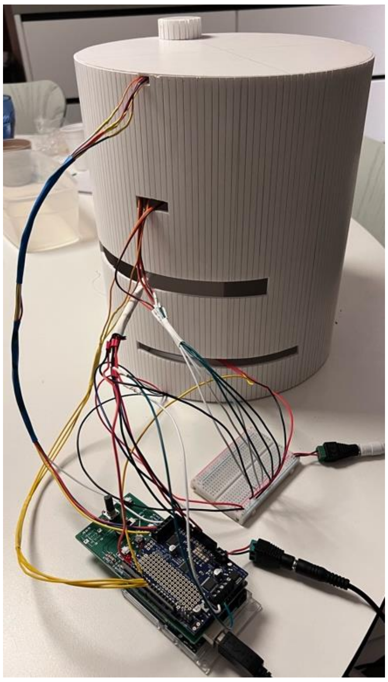


Figure 180, finished beverage machine (4).

Figure 181, finished beverage machine (5).

Figure 182, Finished beverage machine (6).

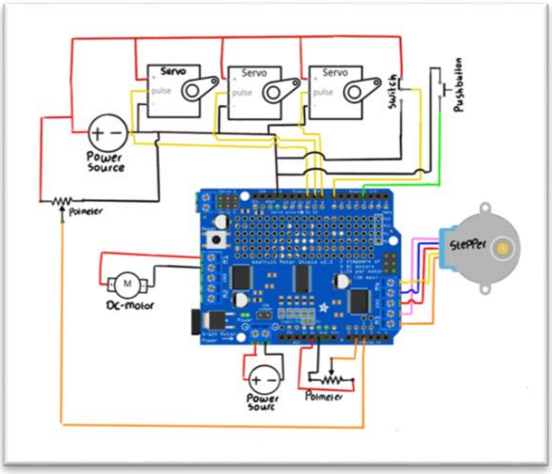


Figure 183, schematic electronics hot beverage machine

Reflection:

Before the start of exploratory making I never used ‘building models’ as an exploratory tool. I have used foam board before to create models, but only to convey my design to others.

In an exploratory setting I have always gone for sketching, writing down or discussing it and I often get stuck in a creative block during the exploratory phase. Therefore, I choose to follow this course to see if I could learn some new techniques and insights to improve my design exploration skills.

During the first three weeks of exploratory making I did not really feel like I learned much besides the technical skills to build models. But once we started making explorations this changed for me.

Having to build and interact with every concept, helped me see details I otherwise would not have seen, like how things feel in your hand or how a certain interaction feels kind of weird to play out.

I focused way more on the interaction of the design than I did previously when I used sketching or discussing concepts. I Feel like the focus lies a lot more on the function of a design or the aesthetics when you use things like sketching, writing down or discussing it. I previously never thought of interaction as being such a critical part of design. I designed a hot beverage machine, that still has the exact same function as any other beverage machine, but just by designing a different interaction or motion, it feels like a whole new design.

Ofcourse I also worked with a lot of mechanisms during this course, where others may find it very beneficial to build mechanisms and see details they otherwise would not have spotted, I have never had an issue with my mechanical insight. Therefore I did not feel like it offered me much in this field, but if I ever have a project with difficult mechanics I will keep this as a helpful option in mind.

The one big downside making has in my opinion is that it is very time consuming. I feel like this extra time is more than worth it when I do my exploration, but when it comes to implementing the explorations into a final design it felt unnecessarily long. This is why I would probably still opt for sketching the final design prototype before I build it, to prevent having to build it twice.

During my explorations I found myself working in a derivative exploration, the feedback I got from simple ideas, gave me new ideas, which gave me new inspirations as well. For example, the way my inputs for the hot beverage felt when I combine them, was like I was gaming with a controller, so this gave me inspiration for new ideas. This derivative exploration process really helped me with my creative block.

I learned to see the importance interaction and motion can play in design. How making my explorations gives me way more detail and feedback than sketching ever can, and how building helps me focus on other aspects then i would with sketching, writing or discussing. I also learned that making is not for me when it is not about the exploration, and lastly, I know that making models can help me out when I am in creative block. This will help me become a better designer in the future.