Ricardo Leduc

Mechatronics Engineering Major
Portfolio

Tecnologico de Monterrey 2018-2023
UC Berkeley 2021-2022 (Exchange Program)

Index

Ricardo Leduc Almaraz

leducricardo@gmail.com
www.linkedin.com/in/ricardoleduc
+52 5531496919 (MEX)

Hardworking and passionate mechatronics engineer, oriented towards robotics and related fields. Enthusiastic about shaping ideas into reality using CAD, programming, and similar tools. Team leader, always determined to find innovative and creative solutions.

EDUCATION

University of California Berkeley, Berkeley, CA *—Aug 2021 - May 2022*
Study abroad year focused on mechanical engineering
Tecnológico de Monterrey (ITESM), Mexico City, Mexico *—Degree Expected May 2023*
Bachelor of Science, Mechatronics Engineering
Mercersburg Academy, Mercersburg, PA *—Aug 2015 - May 2016*
Study abroad year in high school
The American School Foundation A.C*,* Mexico City, Mexico *—Aug 2014 - May 2018*
High School Diploma, IB SL Spanish Certificate, IB SL English Certificate

RELATED PROFESSIONAL & ACADEMIC EXPERIENCE

Group Project, Mechatronics Design (ITESM) *— Aug 2022 - Dec 2022*
- Created as a final project for a Mechatronics Design course at Tecnológico de Monterrey, GardenBot's goal was to function as a drought-tolerant watering system while also being autonomous. The prototype, inspired by the roomba, is a rover-style robot that drives around watering your garden automatically. In charge of the day to day tasks as well as the mechanical and electronic aspects. Helped structure the team through QFD, FMEA, among other processes.

Group Project, MECENG 102B (Berkeley, CA) *— Jan 2022 - May 2022*
- Winner of the ASME Makers Grant Social Impact Award. Semester project, assigned with designing and creating a mechatronic project with a unique mechanical component, coding, and electrical aspects. The project is a distance sensor for the visually impaired using ultrasonic sensors and brush motors.

Mentee, ULAB (Berkeley, CA) *—Aug 2021 - May 2022*
- Undergraduate-led research lab focused on astrophysics. Developed a pipeline to generate radial velocity curves in order to read raw data in the identification of binary systems and exoplanets.

Mentee, Start-Up Nation Mentorship (Berkeley, CA) *—Jan 2022 - May 2022*
- Mentorship program aimed to pair extraordinary student leaders internationally with Israeli executives across the spectrum of industry and government to help ensure that the next generation has the knowledge and relationships they need to succeed.

Group Project, Manufacturing Technologies (ITESM) *— July 2021 - Aug 2021*
- Final project, tasked with creating a virtual CNC milling machine. In charge of the electrical/programming part using arduino, Fusion 360, and Proteus.

Technology Assistant, THEOS (Mexico City, Mexico) *— Aug 2019 - Jan 2020*
- Helped develop a Learning Management System (LMS) at a national level to optimize the field work of technicians. Continued with the implementation of a digital signature in the company's internal processes. Additionally, helped create a change in the culture of THEOS employees.

LEADERSHIP EXPERIENCE

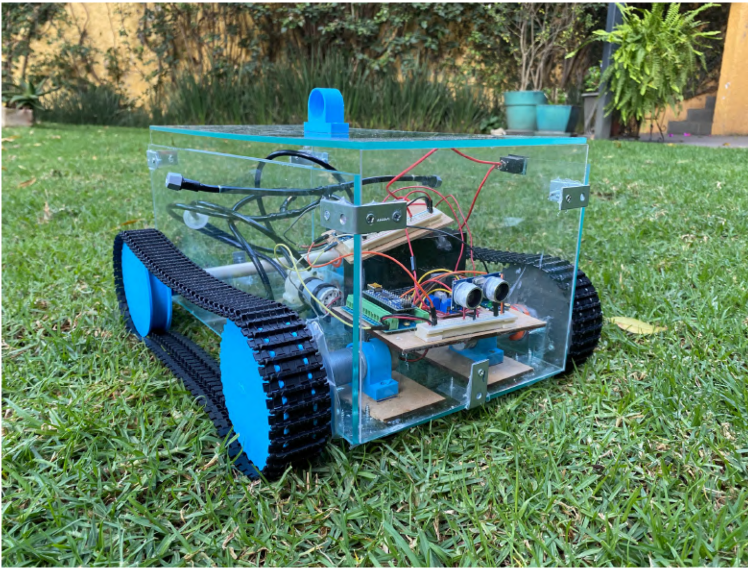
SAIMT, Vice president *— Aug 2019 - Dec 2020*
- Vice president for the Mechatronics Engineering student government. In charge of organizing and leading projects focused on student's mental health, networking, and Mechatronics related events.

Astronomy Club, President/Founder *— Aug 2017 - Jun 2018*
- Supervise and carry out the logistics to organize informative spaces to spread interest in astronomy. The club started with over 20 students throughout the school year and is still active today.

BOWLS Campus Santa Fe, Volunteer *— Nov 2018*
- Student congress focused on connecting leaders and students. Took place in 2018, more than 800 students and 10 speakers attended.

SKILLS & CERTIFICATIONS

- Fluent in Spanish and English
- CSWA certified (Mechanical Design SolidWorks)
- Experienced with programming languages and softwares such as HTML, Javascript, CSS, Python, and MATLAB

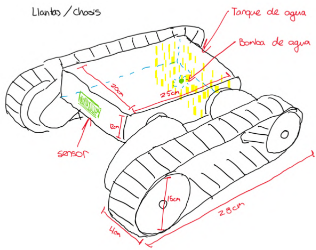


GardenBot

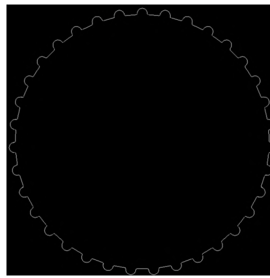
ITESM
August-December 2022

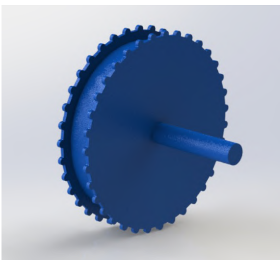
Based on the Roomba, GardenBot is a rover-style robot that drives around watering your garden automatically. Created as a final project for a Mechatronics Design course at Tecnologico de Monterrey, GardenBot's goal was to function as a drought-tolerant watering system while also being autonomous.

MECHANICAL ASPECT

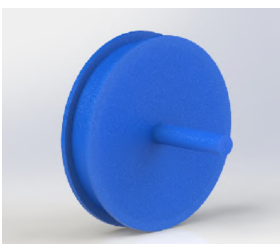


During the design process, we took into consideration the terrain and how we wanted the robot to move. That's why we decided to use moving tracks/links similar to those of a tank. This way, the system could rotate on its own axis and only need 2 motors.

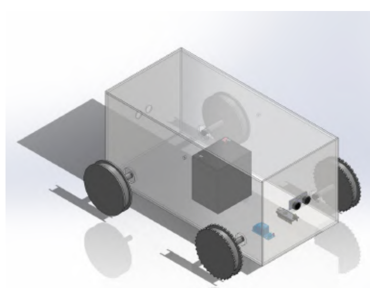




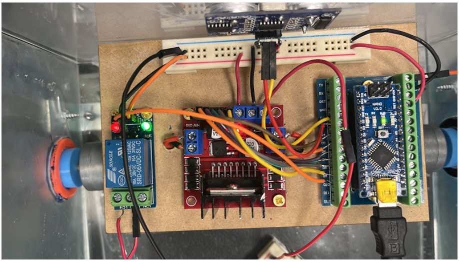
We designed and 3D print the wheels (one with teeth similar to a gear and the other with no teeth). Before printing the wheels, we laser cut it to test it with the links. This let us know if we calculated the number of teeth correctly.

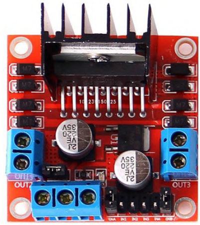


The base was made of acrylic. Each side was dimensioned in SolidWorks and laser cut. Additionally, holes were made for the irrigation system (two on each side and two on the bottom). We made an aseembly using SolidWorks to get a better feel of the robot and its capacity. After the base was assembled, we 3D print additional features for the system, such as a base to hold the motors.

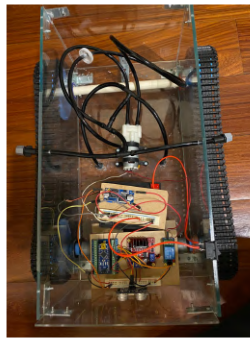


ELECTRONICS





A motor driver was utilized for the 12V DC motors. This was also what bridged the communication between the Arduino and the motor. To control the water pump, we made use of a relay. Finally, a 12V 4Ah battery powered the robot, this was connected to a voltage regulator that sent 5V to the Arduino, relay, and motor driver.



CODE

```
void goForward () {
 digitalWrite(rele, HIGH);
 digitalWrite(Enable_1, 100);
 digitalWrite(Enable_2, 100);
 digitalWrite(Input_1, HIGH);
 digitalWrite(Input_2, LOW);
 digitalWrite(Input_3, HIGH);
 digitalWrite(Input_4, LOW);

}

void goBackward () {
  digitalWrite(rele, LOW);
  digitalWrite(Enable_1, 100);
  digitalWrite(Enable_2, 100);
  digitalWrite(Input_1, LOW);
  digitalWrite(Input_2, HIGH);
  digitalWrite(Input_3, LOW);
  digitalWrite(Input_4, HIGH);

  }
```

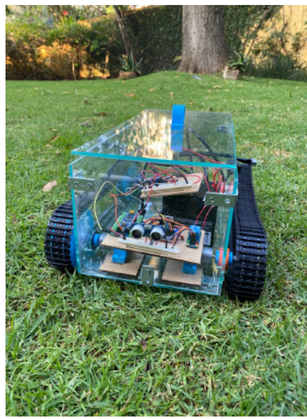
```
void turnRight() {
  digitalWrite(rele, LOW);
  digitalWrite(Enable_1, 120);
  digitalWrite(Enable_2, 120);
  digitalWrite(Input_1, HIGH);
  digitalWrite(Input_2, LOW);
  digitalWrite(Input_3, LOW);
  digitalWrite(Input_4, HIGH);

  }
void stopMoving() {
  digitalWrite(rele, LOW);
  digitalWrite(Enable_1, 0);
  digitalWrite(Enable_2, 0);
  }
```

```
if (distance < 23)
{
stopMoving();
delay(2000);
goBackward();
delay(400);
stopMoving();
delay(2000);
turnRight();
delay(3200);
stopMoving();
delay(2000);
}
else
{
goForward();
delay(2000);
}
```

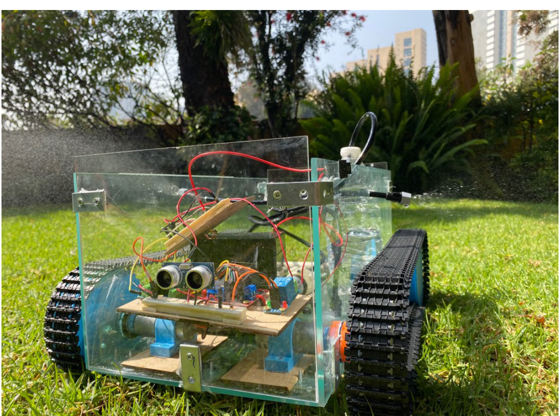
Firstly, we defined each pin for the Arduino and the "enable" of the motor driver, which told the motor which direction to turn. After that, we created a series of functions that either told the robot to move forward, stop moving, go backward, or turn right. These functions were then put into an if statement, telling the program that if the sensor detects a distance less than 23cm it should stop the irrigation system, stop moving, go backward, and turn right. If it doesn't detect anything it starts watering and moving forward, simulating the movement of a Roomba.

FUTURE IMPROVEMENTS

The GardenBot was always intended to be created with a home base, where it would come back, recharge, and refill the water tank. This would require more sophisticated sensors and programming. Nevertheless, it is something that we will consider as we continue to improve the GardenBot.



The implementation of AI can also help this robot reach its full potential to assist the battle against climate change, and more specifically, water waste.

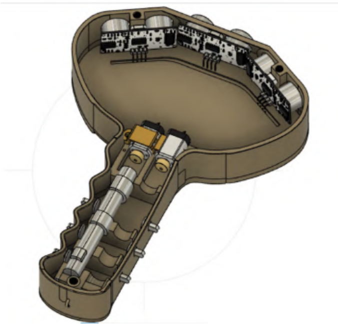


VIVID

UC BERKELEY
January-May 2022

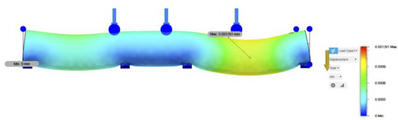
Awarded for the project with the greatest potential to address an existing societal challenge, e.g., climate change, or inequity. Along with 4 other peers, we developed a proximity detection device called VIVID that allows visually impaired people to identify how close an object is to them. This was achieved utilizing an ESP32 microcontroller, ultrasonic sensors, and brush motors. Using a pin system, the user can feel different combinations of pins in their hands according to the readings of the ultrasonic sensor (how far away an object is).

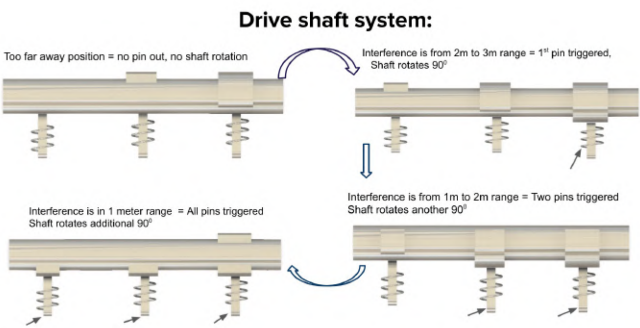
MECHANICAL ASPECT



The prototype features a 3D-Printed main body, 3D-printed pins, and a machined aluminum driveshaft. All these features were developed in Fusion 360. In the early stages of the project, we ran a finite element analysis (FEA), to determine if the 3D-printed driveshaft would be able to withstand anticipated axial forces exerted by fingers on each pin.

I was one of the main designers for this part of the project





As explained in the diagram above, the drive shaft system functions as a short-stroke, variable linear actuator for three pins simultaneously linked to one motor. These pins then exert pressure on the hand, which facilitates haptic feedback. In simpler terms, each side of the handle has three pins. Depending on the distance of the object, a specific number of pins would nudge the hand of the user.

CODE

```
int Distance(){
  int post;
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  float d1 = pulseIn(echoPin1, HIGH);


  float r1 = (d1*.0343)/2;
```

I helped develop the code which reads the data from the ultrasonic sensor and transmits it to the motor encoder, indicating its angle of rotation. We used cases to set the target position of the angle of rotation. Case 1 activates one pin, case 2 activates two pins, and case 3 activates three pins.

```
void readEncoder(){
  int b = digitalRead(ENCB);
  if(b > 0){
    posi++;
  }
  else{
    posi--;
  }
```

```
void loop() {
  times = millis();
  ds = Distance();
  long dif = times - Tus;
  // set target position
  if (dif>0){
    Tus = times + 1000;
    switch (ds) {
      case 0:
        target = 0;
        break;
      case 1:
        target = 57;
        break;
      case 2:
        target = 114;
        break;
      case 3:
        target = 171;
        break;
            case 4:
        target = 0;
        break;
   }
```

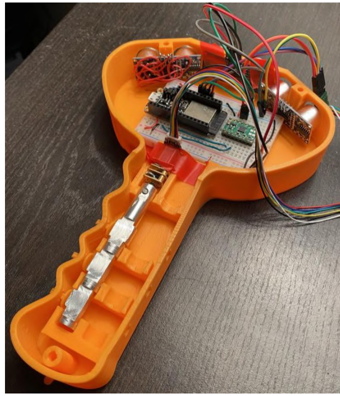
ELECTRONICS

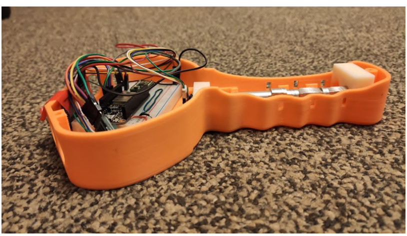
Another of my main tasks was the circuitry. We used an ESP32 microcontroller, a DC motor (with an encoder and driver), and ultrasonic sensors. The sensor was connected to the ESP32 which then relayed information to the encoder. The entire system consumes 15.16 Watts, with the motors drawing most of the power.

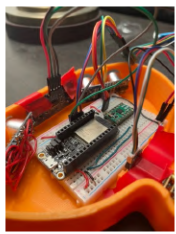
FUTURE IMPROVEMENTS

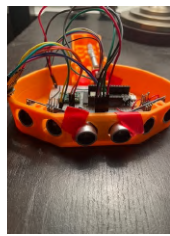
Future improvements could include the addition of a phased array ultrasonic system, which will allow for a longer range and more precision when compared to our current system. Depending on future research, we may opt to develop a modulated transducer. This could allow us to use obstacles as speakers, playing audio from each obstacle.

We won the American Society of Mechanical Engineers (ASME) Social Impact Award, given to the project with the greatest potential to address an existing societal challenge.









ULab

Physics & Astronomy Division

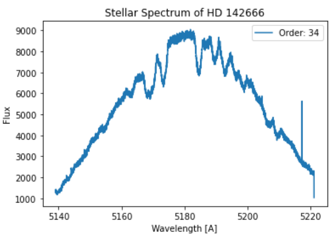
Creating a Pipeline to Generate Radial Velocity Curves from Raw APF Spectral Data

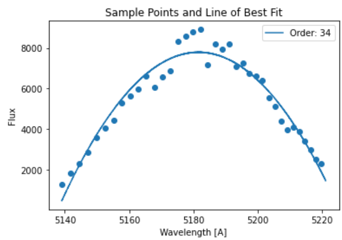
UC BERKELEY
August 2021 -May 2022

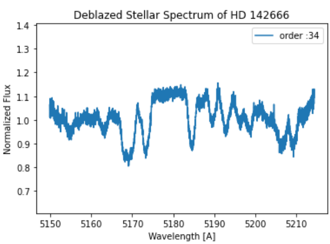
The goal of our project was to find binary star systems by analyzing stellar spectra to measure radial velocities. We collected data from the APF at Lick Observatory from 28 young stars. Gathering the stellar spectra of these star systems allows us to plot their radial velocity curves. If a binary companion exists, we can characterize them by using the radial velocity measurements to calculate constraints on the companion's mass and orbital period.

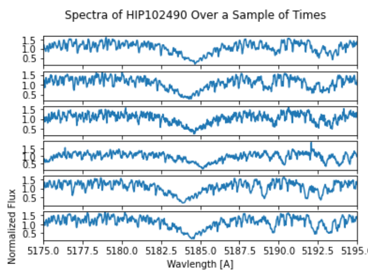
METHODOLOGY

We did our work in Python where we took in inputs of raw APF stellar spectra to generate radial velocity time series for a given star. To begin, we read the stellar spectra and started to clean the data of artifacts present because of the telescope's optics. This consisted of deblazing, normalizing, and cropping the data.

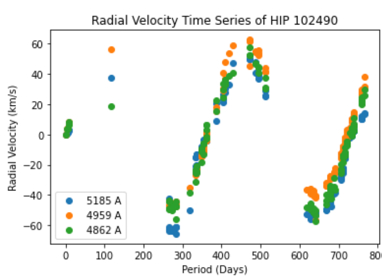






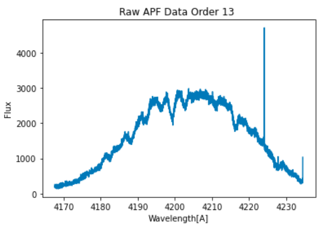


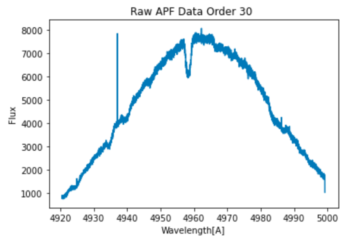
Our next step was to calculate changes in radial velocity over time. The Doppler effect causes changes in the observed wavelength of light due to changes in the velocity of an object. Calculating changes in observed wavelength allows us to convert our doppler shift measurements to velocity measurements. We created a python function to perform this task. After correcting for Earth-caused doppler shifts, We generated a radial velocity time series that matches a binary characterized in 2005 as shown to the right.



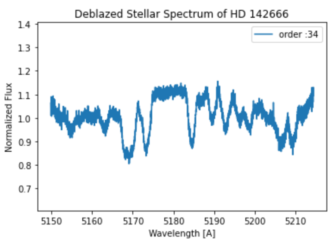
The two graphs below (raw stellar spectra) represent how much light the star is emitting. The lower orders correspond with how much light in the low wavelength region the star is emitting whereas the higher orders are showing how much light in higher wavelength regions the star is giving off. If we look at the successive observations, we can see how it is moving back and forth. This suggests that there may be a binary companion.

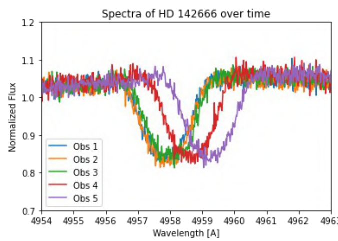
Raw stellar spectra:



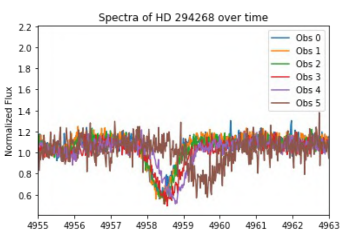


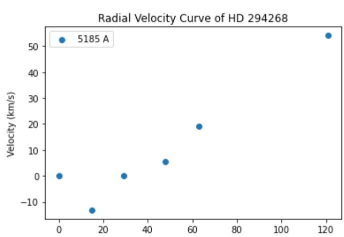
Deblazed data:





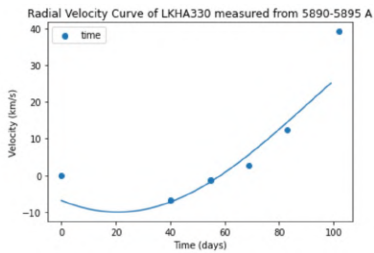
Since we know our code generates accurate radial velocity curves, we generated some using our data.

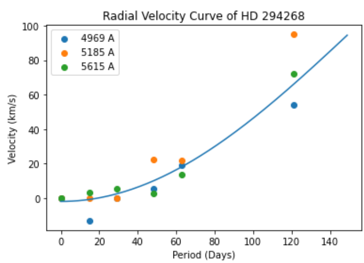




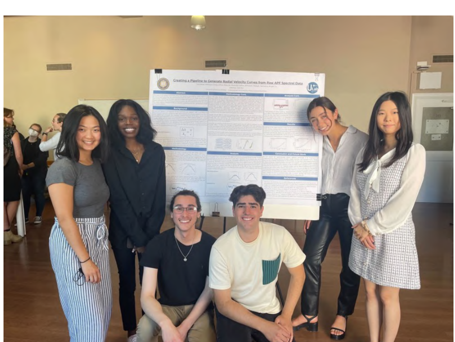
This radial velocity time series seems to show a sinusoidal trend, inferring that the star may have a companion. If the time series showed a horizontal line, it would suggest that there is no movement, and thus no binary companion.

Looking at the radial velocity curve, we can determine whether the given star has a radial velocity change over time that would fit a sinusoidal curve. This shows the star may have a companion that provides a gravitational pull, causing the primary star to fluctuate periodically. Below are examples of the radial velocity curves we generated for LKHA 330 and HIP102490. We fit curves to the graphs, which means the stars we analyzed could potentially be in a binary system.





Since the young stars we focused on have protoplanetary disks, our research could be continued to give insight into the phenomenon of planetary formation.

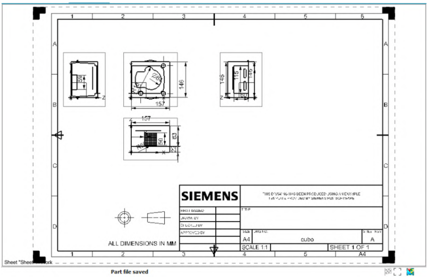


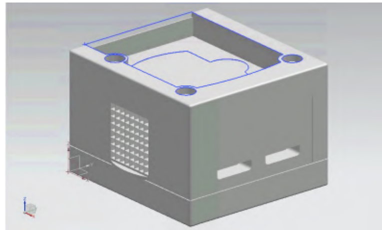
3D CAD Gamecube

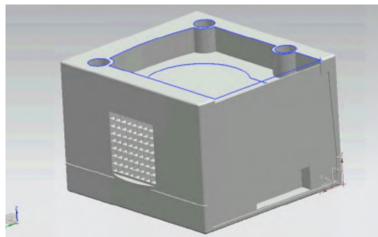
ITESM
January-May 2020

Using the software NX Cad, I was tasked with replicating a real-life object into Computer-aided design (CAD). Being a longtime fan of video games, I grabbed my Nintendo GameCube and challenged myself to replicate it as best as I could. One of the main goals of the project was to create individual parts and join them together in an assembly.

In order to recreate an object such as the Gamecube, it was crucial to identify each feature and design them. I did just that, creating 11 parts. Starting off with the base of the cube.

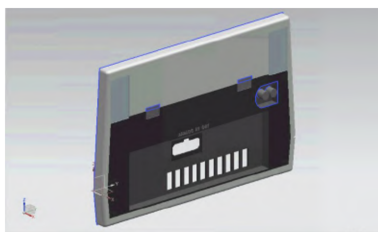




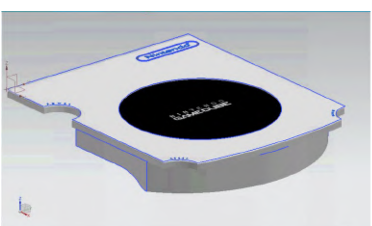


The main techniques used were the extruding, mirror, and pattern functions. After the main part was finished, I moved on to the more detailed ones.

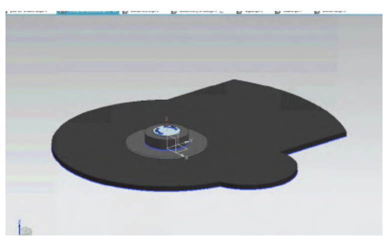
Control Input. Functions most commonly used: pattern, extrude, extrude cut, filet



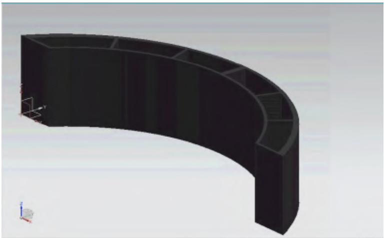
Dorsal part. Functions most commonly used: pattern, extrude, filet



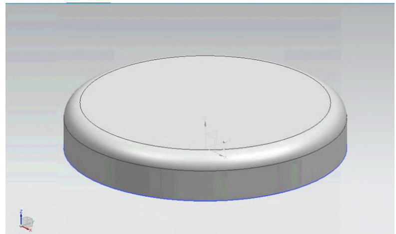
Lid. Functions most commonly used: mirror, extrude, filet



Disc input. Functions most commonly used: extrude

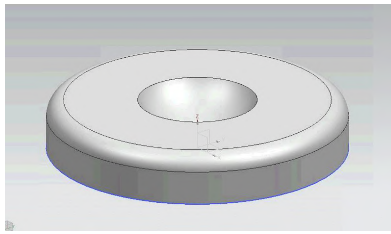


Back handle. Functions most commonly used: pattern, mirror, extrude cut.



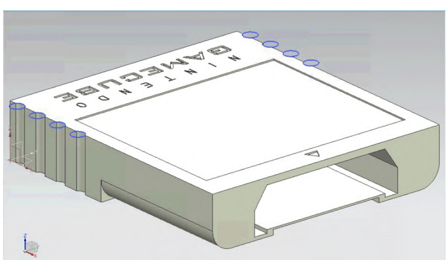
Power/restart button. Functions most commonly used: extrude, edge blending

Additionally, I designed features/objects that were not directly part of the Gamecube, such as the disc and the memory card. I even added details like the cover of the disc and the text in the memory card (as realistic as I could make it).



Eject disc button. Functions most commonly used: extrude, extrude cut, edge blending

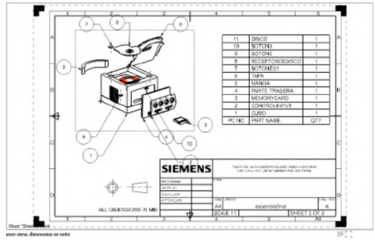
Disc. Functions most commonly used: extrude, extrude cut



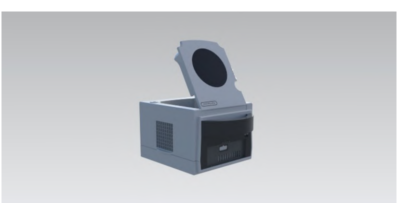
Memory card. Functions most commonly used: extrude, extrude cut, mirror, pattern

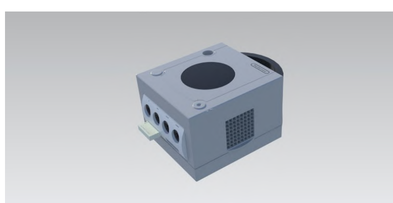
Finally, after designing and giving color to all the parts, I joined them in an assembly. I did this by making parts concentric to one another or touch-aligning them.

After it was done, I made drawings for each part, and the assembly (like the one to the right). They include their respective measurements, and this one specifically, details each part along with its name.



This project was my first experience with Computer-aided design (CAD) and NX Software. Therefore, seeing how it turned out in comparison to the real object, encouraged my passion for CAD and engineering.





Ricardo Leduc Almaraz

leducricardo@gmail.com
+52 5531496919 (MEX)
www.linkedin.com/in/ricardoleduc