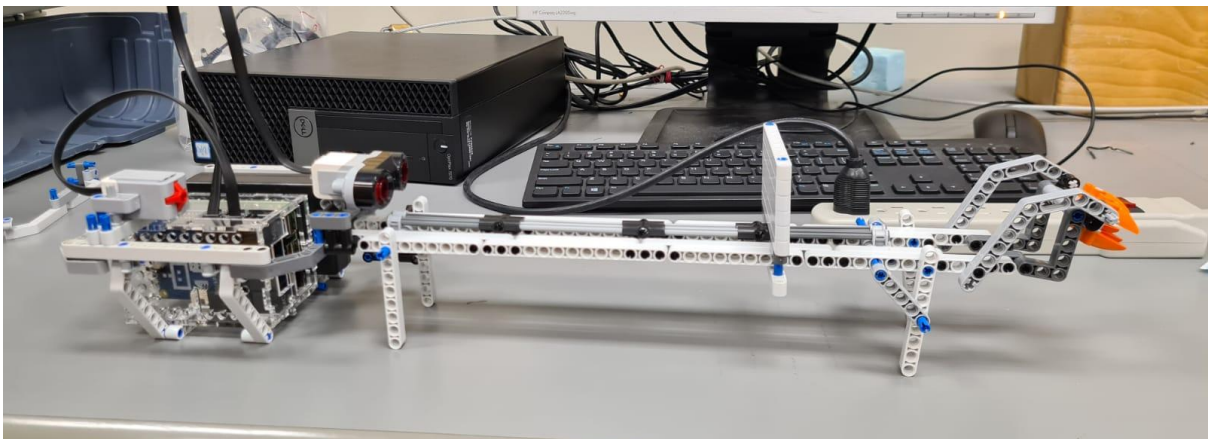
Lab 3: Part A

Group 33

Names and IDs:
Aly Mohamed - 260902616
Robert Aprahamian - 260889522
Alexandre Saroufim - 260886799

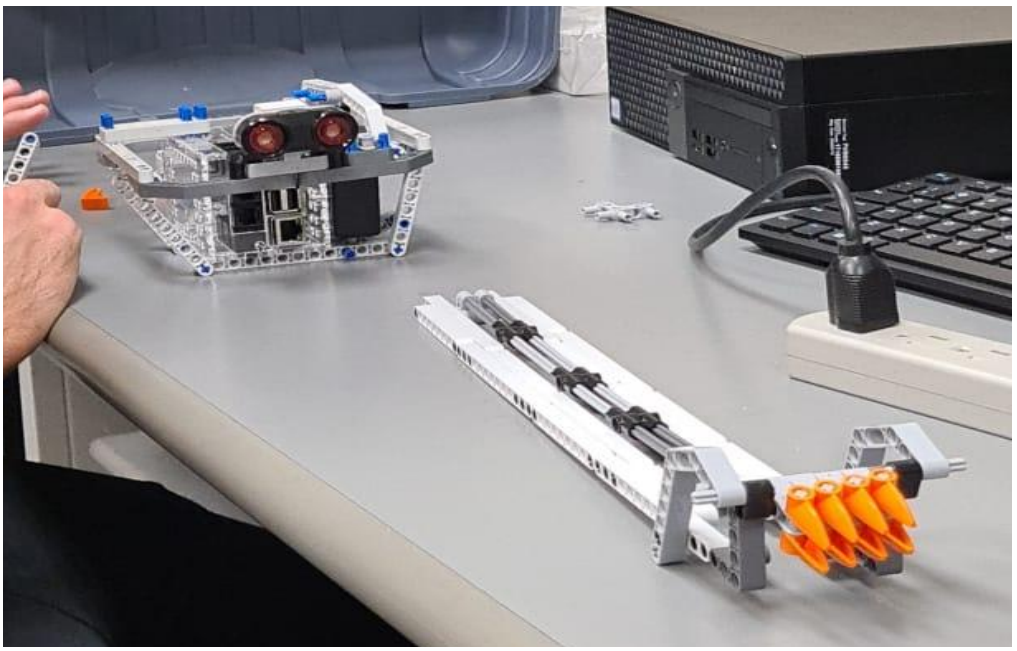
Section 1: System design

Our instrument is a new generation guitar, it relies on the ultrasonic (US) sensor and a touch sensor. When the touch sensor is pressed, the ultrasonic sensor reads the distance to the the plaque we are placing in front of it (explained in the pictures below)., and plays the note corresponding to that distance. Users can know what distance corresponds to which note through the black marks, each distance between two black marks contains a specific note.
Our system is an electronic guitar, it is constructed of two subsystems: the head and the neck. The head is the part containing the BrickPi, the battery, the touch sensor and the US sensor. And the neck contains the black marks that determine notes and the plaque that can be moved to change the note played.
The inputs to this system are: the touch sensor, the ultrasonic sensor (through calculating the distance to the plaque) and the output here is a music note.



Here is our final system!

This picture shows both big subsystems in our system, on the left there is the BrickPi, the US sensor and the touch sensor, on the right there is the plaque (not in this picture), and the neck with the black elements that delimit the range for every music note.

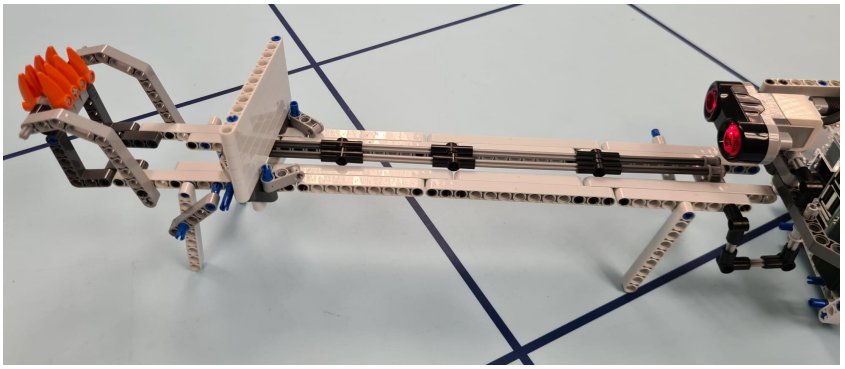


Section 2: Subsystem design

These two subsystems are connected through lego design that puts the neck right in front of the ultrasonic sensor, so we decided to leave a 3 cm minimal distance between the plaque and the US sensor to avoid having a result of 255 cm when the plaque is too near to the sensor.

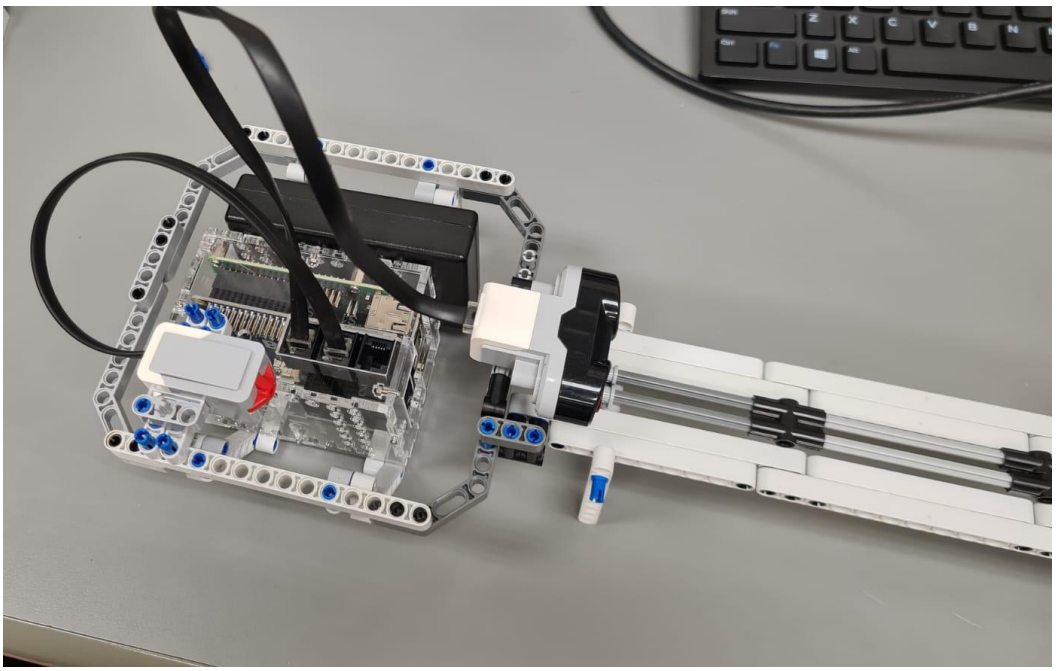
The neck

We have a long neck that contains a plaque that can move, then depending on its place, a note will be played. We also have a guitar head (in orange) for decoration to keep the main design of a guitar in our instrument, however these are not functional. One important thing in the neck is the bars holding it, they are crucial to keep the system stable and to make the US sensor measurements accurate, if these bars are not positioned correctly, the US sensor wouldn't be able to detect the plaque precisely.

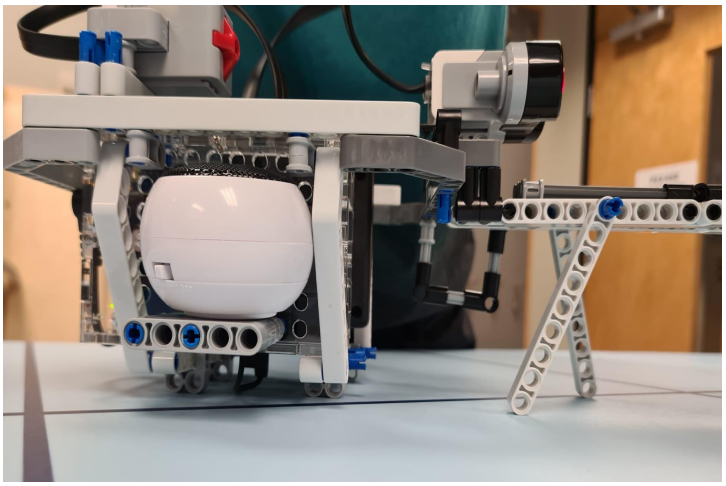


The body
As mentioned before, the body contains the BrickPi, the US sensor, the battery, the speaker and the touch sensor. The design is, we have a hoding base, that holds all these elements, we put the BrickPi and the battery on this base. The BrickPi is also held with lego from upwards to make the body more stable, and on the BrickPi wi have the touch sensor fixed there. The battery is positioned next to the BrickPi, the US sensor is positioned on the right limit of that base, and right in front of the connection with the neck.

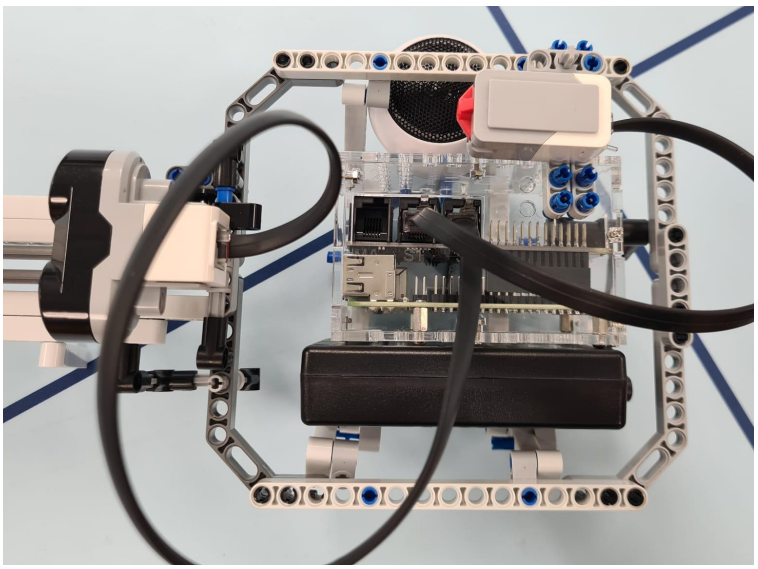
In the following pictures, we can see the positioning of elements in the guitar body as described above.



The US sensor is right on the limit of the guitar body, it is positioned such that it can see the plaque moving



The BrickPi holder is elevated by one level of the floor to leave space for the speaker plugin, and the speaker is put on a holder to make sure everything is stable.



Section 3: Validation and Testing

Test conducted	Test results	Use of the test in design
Code testing with hand to see if notes are played with right distances	Code works perfectly.	Helped us decide to proceed to the hardware design phase as we have a software system ready.
Test if the body in lego will hold our guitar being lifted from the ground	Hardware not reliable enough to lift our guitar	Helped us to take the safer decision of keeping it on the ground for total safety.
Testing the functionality of the whole guitar together (software and hardware united)	System works perfectly	We knew after it that our design is complete, reliable and ready to be presented.

For the last test here is a summary of how it went: The 4 notes are E4, G4, A4 and B4, with limiting distances being 6.5, 14, 19.5, and 28.25

Distance (in cms)	Expected note	Note played
4	E4	E4
6	E4	E4
11	G4	G4
13.5	G4	G4
18	A4	A4
20	B4	A4

The we had a problem at 20 cms, and we realized we mismeasured our limits. So we changed it to 6.5, 14, 21, and 28.25 and had the following values.

Distance (in cms)	Expected note	Note played
4	**E4**	**E4**
6	**E4**	**E4**
11	**G4**	**G4**
13.5	**G4**	**G4**
18	**A4**	**A4**
22	**B4**	**B4**
27	**B4**	**B4**

Section 4: Conclusions

We are very happy with the system we have, it can play music nicely without any airflow from the user, and also the tests conducted give very accurate results! Some of the successes are: the instrument can play 4 different notes, the instrument does not need any air flow from user, and the instrument can be held and played by hands! Also the idea we proposed in lab 1 (copied below) is working well and is very accurate. We also could overcome the technical challenges that we thought were complicated like: applying code on hardware and getting accurate results.

“Have one touch sensor control whether the sound is on or not (when touched it is on). Then have the ultrasonic sensor check the distance from a plaque, every range of distances will be corresponding to a musical note”

One failure is that our instrument is not very reliable when held with hands, it can technically be done, but the user has to be very vigilant while holding it. So we prefer using our instrument when it’s on a table.

We believe that our budget estimations in lab 3A were accurate, we took some more time to test, to make sure our instrument is working well, however for the other tasks we didn’t spend more time than expected. Also we performed well regarding milestones, we actually finished our tasks earlier than predicted, as we were done with the instrument on Oct 23rd (one day earlier than expected). And were done with testing on Oct 25th. We performed the tasks in the order specified in lab 3A, we first wrote code, then designed and built the instrument.

Regarding division of labor, we all worked together on the code and the lego design, and then also built it together, then we divided the report on the three of us.

Time spent per task

Task	Person assigned	Time spent
Revise Requirements	Aly Robert Alex	1
Roles and Responsibilities division	Alex	1
Gantt's chart	Robert Aly	2
Budget	Alex	1
Meeting minutes	Aly	0.5
Come up with the code to play our 4 different notes	Aly Robert Alex	6
Design the body of our instrument with lego	Aly Robert Alex	6
Build the body of our instrument with lego	Aly Robert Alex	6
Testing	Aly Robert Alex	11
Setting up the equipment (including time it takes to reach the lab and setup the BrickPi and anything else to start core work)	Aly Robert Alex	3
Meeting minutes	Aly	0.5

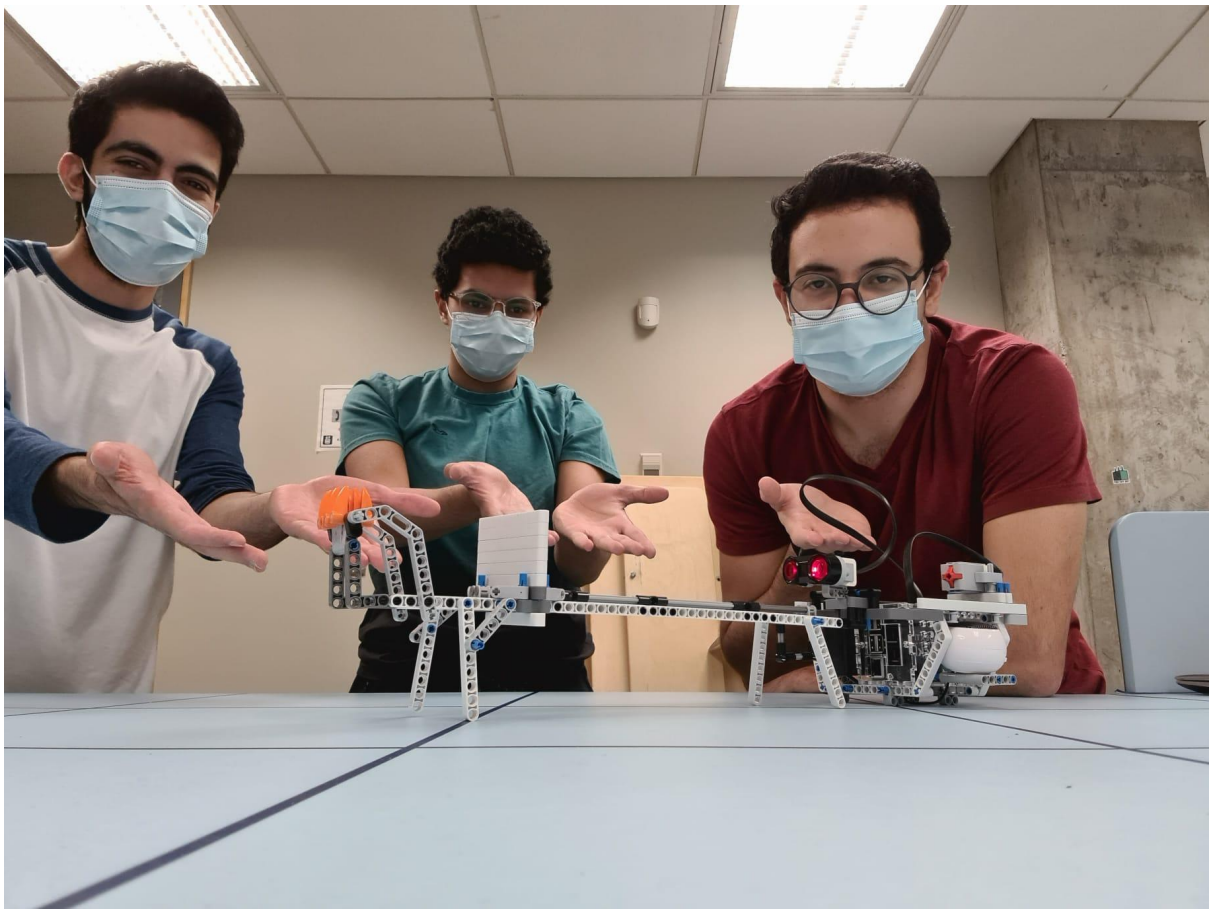
The team worked perfectly together. Communication was smooth and everyone contributed equally and with a lot of effort.

What went well?

Getting code done and then designing lego was good for this project, because we knew exactly what had to be done with code, we may use the same approach for the final project if it's similar to this one. Building the lego together was a good idea also as we were all sharing design ideas together and that helped us find efficient solutions to some problems.

What can be ameliorated?

We could make the design clearer before building the instrument, and had to take into consideration some factors we forgot to consider, it will be a lesson for the final project. We will think of the smallest details of design before we build our instrument.



Thank you!