

CDT 31160
Practicum in Robotics

Prof. Charles R. Crowell	Prof. Michael A. Villano
126 Haggar Hall	215 Haggar Hall
574 277-4774	574-631-9894
ccrowell@nd.edu	mvillan1@nd.edu

Course Syllabus and Guidelines (rev. 011216)

Description



This course will allow students to work with the Nao humanoid robot platform shown above. Students will learn about how to control the sensory and motor capabilities of the robot to produce specific sequences of robot behaviors and/or to allow the robot to respond to particular inputs from the external environment. Students will work with the instructors to identify the specific behaviors and response sequences to be created.

The class is self-exploratory in nature so students will need to learn on their own how the Nao platform works in order to complete several specific weekly assignments (noted below) over the first six weeks. In addition to the instructors, there are a variety of learning resources available in the lab and online to assist with learning how to use the Nao platform. The weekly assignments to be completed will involve projects such as developing specific behavioral sequences for Nao, exploring Nao's various motor and sensory abilities, or exploring the various environments with which Nao can be programmed.

In addition to the six weekly assignments, students will complete a final project to be presented at the National Robotics Week Exhibition on campus in April. The 2016 Robotics week is April 2-10 and the on-campus exhibit will be on Sunday April 10th in the Compton Ice Arena from about noon

to 4p. The final project will be a collaborative effort of the class, all members of whom should plan to attend the exhibit on behalf of the *e*Motion & *e*Cognition Lab to demonstrate their project to attendees at a booth/table reserved for the lab. The project should be designed to be interactive and of interest to the general audience, who will be children and adolescents from 4-16 and their parents (see below for more details and suggestions regarding the exhibit).

Recent final projects for the National Robotics week exhibit have involved creating dance sequences (<https://twitter.com/NDNRW/status/587320706544828416>) or other behaviors for the Nao (<https://twitter.com/NDNRW/status/460472899615199232>), programming the Nao to demonstrate Notre Dame related behaviors (<https://www.instagram.com/p/1I4SyEjLlu/>, <https://www.instagram.com/p/1L3ISCjLP1/>, <https://www.instagram.com/p/1TiOmsDLEA/>, <https://www.instagram.com/p/1YzAasDLAV/>), or creating an interactive sequence where Nao plays a game with the audience or imitates something an audience member does (<https://twitter.com/NDNRW/status/458612577669361665>). A picture archive of the last several exhibits can be found at: <https://engineering.nd.edu/NDNRW/previous-nrw-photos>. Look for the Naos in the pictures.

This class will use the two weekly meetings each week for different purposes. The Tuesday meetings each week will involve demonstrating the completed assignments for that week, as noted in the assignment schedule below. Instructors will be present on Tuesday for observation and discussion of each assignment. The Thursday meeting each week will be used for individual work on your own. Once all the scheduled assignments have been completed, students can use both class meetings each week to work on their final projects on an exclusive basis. Meetings with instructors can be arranged during this time on an as-needed basis.

Since this is a 3-hr class, students are expected to put in an effort equivalent to other three-hour courses. A 3-hr. class involves roughly 40 hours of class time in a semester. For every hour in class, it is not unusual to spend 2-3 hours outside of class in related work. This rule of thumb would total 120-160 hours over the course of a semester.

As you can see, the course requires students to complete one or more projects using Nao and its associated systems. To be successful in this course, students must spend time spent with the robot in the *e*Motion & *e*Cognition lab. The following links provide access to important manufacturer documentation for the NAO, including manuals and tutorials. NOTE: This documentation is for the latest release (2.1.4) of the NAO operating system and SDK, as well as the former release (1.14.5), both which may be different from the system we currently have installed. However, despite the differences, the info at this link still will be useful. The tutorials will be especially helpful including the 4-part course video (<https://community.aldebaran.com/en/content/nao-robot-course>), though it's a little hard to hear.

<https://community.aldebaran.com/en/resources/documents>

<https://community.aldebaran.com/en/resources/tutorials>

Also, you can browse the manufacturer's website at <http://www.aldebaran-robotics.com/en> for additional information that may be useful, such as the download section, the forum, and the store.

You can use my Aldebaran login: ccrowell@nd.edu; pswd: emotecog. Please be respectful in your use of my account and do not embarrass me with inappropriate posts.

Class Meetings

The class meets officially from 3:30-4:45 on Tuesday and Thursday. As noted above, Dr. Villano and I will be present for the Tuesday meeting for demos and discussion of relevant matters related to robots in general or the Nao in particular. The Thursday meeting time will be reserved for you to work with the Nao on your own. We will have a robot development station in B09B with all of the relevant software needed to control the Nao and create projects. Any available simulators also will be installed on this machine. We ask that you try to use this Thursday time as much as possible we share that room for experiments at other times during the day. However, we will provide the lab access code so you can work in B09B in the evenings or when no studies are running.

Homework assignments

As a way of getting you started working with Nao, we recommend the following homework assignments to be demoed in class over the next six weekly Tuesday meetings after the first one.

1. Simple behaviors - at least one limb movement and one vocalizations in sequence or in parallel.
2. Complex behaviors - Multiple movements of different limbs combined with vocalizations and LED changes.
3. Behaviors in response to sensory stimulation - a behavioral sequence in response to a sensory input in a loop.
4. Programming with Choregraphe - using different kinds of box inputs and outputs to achieve some logical control over behaviors (e.g., timing, counting, randomization)
5. Nao programming - implementing a simple behavioral sequence using external programming methods, or by changing the features of a Choregraphe box.
6. Using Domer – Set up a simple control sequence for the Nao using the Domer system developed by Dr. Villano.

Guidelines

Here is what we expect for an "A" grade in this course. Students are expected to attend all group progress report meetings and any individual meetings scheduled with the course instructors. It is the student's responsibility to schedule individual meetings as necessary and appropriate.

Students will need to provide us with the agreed-upon working final project(s), together with appropriate documentation regarding the project. All final project(s) should be showcased during the Notre Dame National Robotics Week exhibit in April, 2016 at the *eMotion & eCognition* (our lab) exhibit table. This exhibit will be held in the Compton on April 10th. More details about the exhibit are available at <https://engineering.nd.edu/NDNRW>.

Project documentation should explain how the project was done and how it works.

Our definition of a working project is as follows. The final product should:

1. Have a description with objectives and a complete set of approved specifications developed in the planning stage.
2. Operate or demonstrate what is indicated in the specifications statement while meeting the objectives listed in the project description.
3. Contain and use all relevant information defined in the planning and specification process.
4. Have appropriate documentation and code where appropriate.

All projects, demonstrations, and write-up are due on the final exam day for this course, or on the last day of finals, whichever date we agree on. We will evaluate the project based on how it works and how closely it conforms to the above criteria. Also, we will consider meeting attendance. If all criteria are not met, or there were unexcused absences from meetings, a grade less than "A" will be assigned depending on the number of absences and extent of project completeness/usability. For team/group projects, all members will get the same grade.

Project Reports

For each project, an individual or team write-up is required. This document should describe what each contributor did in the project and, most importantly, should document the steps involved in development as well as what was learned regarding whatever tools were employed in the project. For example, if your project involved the NAO SDK, we would like the specific steps listed for accessing and using these materials, as well as all parameters used. In addition, if there are things you learned how to do and tips or tricks you learned that are not clearly documented in existing materials, we would like these included in the report. By documenting these steps, you will be providing a guide for others who may have to do some of these things at a later point in time.

In the report, we would also like at least a rough statement of the time you devoted to the project in relation to the 120-160-hr rule-of-thumb noted above.

If any project was not completed as per the original specifications, please clearly indicate what was actually done and what remains to be finished. Also, make sure to indicate the location of all relevant project files, together with file names and any specific instructions for accessing or activating your materials.

Please submit these project write-ups to both Drs. Crowell and Villano no later than the final exam day for this course, or the last day of finals, whichever date we agree upon.

In summary, please note that all write-ups should:

1. Be emailed to us on or before the final exam day for this course or end of final week, whichever date we agree on.
2. Include a brief description of what you did and how much time you spent.
3. Describe what tools you employed along with documentation about specific steps to follow in using those tools.

4. Indicate names, locations and instructions (if needed) for accessing all files/materials related to your projects.
5. Include a self-assessment as detailed below.

Self Assessment

In this class, we will ask students at the end of the semester to provide us with a self-assessment of their overall performance. This exercise helps students reflect on their progress, activities and effort in relation to the project objectives and guidelines we established earlier in the course, and it helps us understand their perspectives.

Taking into consideration your overall effort in relation to course expectations and guidelines, the progress you will have made on your completed project, your efforts to keep us informed, and the extent to which you solicited our input throughout the semester, please provide the following honest and realistic assessments:

1. The lowest grade your performance deserves. Why?
2. The highest grade your performance deserves. Why?
3. The grade you really and truly think your performance deserves. Why?

Feel free to combine all the whys into one statement. Note that if all three of your estimated grades are the same, it is possible you are not being completely realistic and objective. Also note that it will not help your cause in our eyes if your response comes across as unrealistic.

Please append the self assessment to your final project reports.

Example Previous Project Description (contains helpful info about National Robotics Week Exhibition)

Our definition of a working project is as follows. The final product should:

1. Have a description with objectives and a complete set of approved specifications developed in the planning stage.
 - The project was intended to entertain and educate children at the National Robotics Week Fair on April 12, 2015. Given that this occurred at Notre Dame and many people in the area are fans, we assumed that many of the fair-goers would be Fighting Irish fans, but we also wanted to ensure that non-fans and parents were entertained as well. It was also important to us to show how much the NAO can do in terms of sensors and movements.
2. Operate or demonstrate what is indicated in the specifications statement while meeting the objectives listed in the project description.
 - For the Irish fans, we created the following programs:
 - i. 1812 Overture
 - ii. Alma Mater (mostly for alumni)

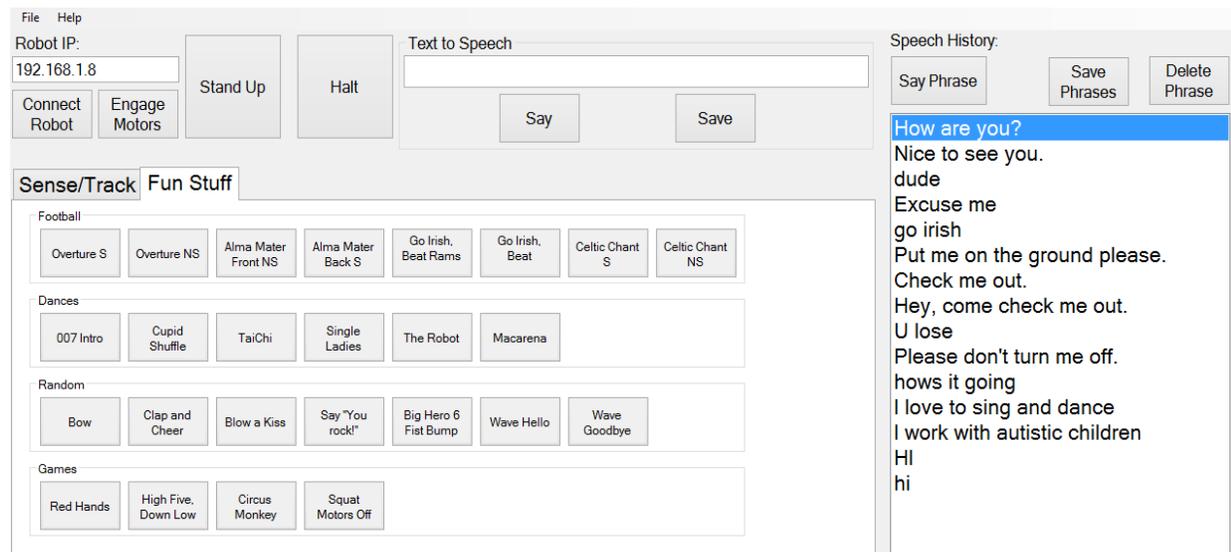
- iii. Go Irish Go
 - iv. Celtic Chant
 - For fair-goers who wanted to see more complex movements or dances, we created or included the following programs:
 - i. Cupid Shuffle
 - ii. Single Ladies
 - iii. Macarena (mostly for parents)
 - iv. Tai Chi Chuan
 - v. 007
 - For quick demonstrations, we created or included the following programs:
 - i. Bow
 - ii. Kneel
 - iii. Clap and Cheer
 - iv. Blow a kiss
 - v. Say, “You rock!”
 - vi. High Five
 - vii. Sit
 - viii. The Robot
 - ix. Wave Goodbye
 - x. Wave Hello
 - For children who obviously could interact with the robot without fear for either party’s safety (children are older, closely watched by their parents, calmer, or held by siblings or parents), we created or included the following programs:
 - i. Fist Bump from Big Hero 6
 - ii. Red Hands Game
 - iii. High Five, Down Low Game
 - For younger children who might not understand the aforementioned dances or references, we created the program Circus Monkey in which the NAO rubs his stomach and taps his head at the same time.
 - We included the Squat and Motors Off program as well so that the NAO could rest and so that we could demonstrate how his limbs could move more easily while he was in a kneeling position with his motors off.
 - If none of those were interesting to a fair-goer, DOMER allows the user to type in whatever they want the NAO to say, which was useful in that we would have the NAO say, “Hello, (child’s name)! How are you?” and hold conversations which delighted the children and even the adults with them.
3. Contain and use all relevant information defined in the planning and specification process.
- We knew that the audience would be primarily younger children from what Professor Crowell and Professor Villano told us, but we also wanted to be sure that we had something for every audience. We made sure to have both Notre Dame and

non-Notre Dame related programs to ensure that even those who were not fans could appreciate all that a NAO can do.

- We also knew it would be fairly loud and busy in the fair, so we did not create many programs that necessitated the use of sensors or cues because 1) verbal cues would be hard for the NAO to pick up in the loudness, and 2) we did not want the children to think they could touch the robots to make them do things because they saw us doing so. Telling a child not to touch a robot does not always work, so we wanted to minimize the interaction as much as possible for the safety of the NAOs.
 - We brainstormed while watching the NAO Evolution of Dance video on YouTube, but we also looked through past projects to get ideas of what previous classes had done.
4. Have appropriate documentation and code where appropriate.
- All of our projects included developing new boxes and sequences using Choregraphe. The final projects and the configuration file for DOMER (NRWF2015.cfg) that we used at the fair are located at C:\Users\emotecog\Desktop\Practicum in Robotics 2015_Carina and Jeff\Final Materials. Files that end with “NS” mean they are part of a pair and those do not have sound; on the other hand, files that end with “S” mean they are the ones in the pair that do have the sound. They are named:
 - i. 007
 - ii. 1812_Overture_NS
 - iii. 1812_Overture_S
 - iv. Alma_Mater_Back (used for the robot positioned behind the other)
 - v. Alma_Mater_Front (used for the robot positioned in front of the other)
 - vi. Big_Hero_6_Fist_Bump
 - vii. Bow
 - viii. Celtic_Chant_NS
 - ix. Celtic_Chant_S
 - x. Clap
 - xi. Cupid_Shuffle
 - xii. Go_Irish_Beat_Rams
 - xiii. Go_Irish_Beat_Tarheels
 - xiv. Go_Irish_Beat_Wolverines
 - xv. High_Five_Game
 - xvi. Kiss
 - xvii. Macarena
 - xviii. Red_Hands
 - xix. Single_Ladies
 - xx. Squat_Motors_Off
 - xxi. standup

- xxii. TaiChiChuan
- xxiii. The_Robot
- xxiv. Wave_Goodbye
- xxv. Wave_Hello
- xxvi. You_Rock

5. We used DOMER as the overall system manager with the following display broken up into “Football,” “Dances,” “Random,” and “Games.” The button names correspond to each program as outlined in the configuration file NRWF2015.cfg in the folder located at C:\Users\emotecog\Desktop\Practicum in Robotics 2015_Carina and Jeff\Final Materials.



Specifics:

- In order to use DOMER, you must first upload the configuration file and then connect to the robot.
- The 007 requires that the robot falls over unless he is originally in the standing position.
- The robot also moves in a lifelike manner when you use the “Text to Speech” capabilities, so just be sure to be holding on to him even if he will “just” be speaking.
- If you write “HI” (all caps) the NAO will say “Hawaii.” For the greeting, either write “Hi” or “hi.” Otherwise, capitalization doesn’t really matter.

Fair:

We arrived at the fair around 10:30 to set up and test the robots. To our surprise, the major audience was not the adults but the children. The ‘Big Hero 6 fist bump’ was the most popular program for the children, and many children recommended our booth to their friends, so that we had groups of children gathering in front of our booth. We showed them “Macarena,” “Single Ladies,” “Cupid shuffle,” “Tai Chi,” “Celtic Chant,” and so on. Most children were too young to know the Macarena. For the adults, most of them were fans of Notre Dame, so they wanted to see the Notre Dame cheering songs. From the beginning, we did not plan to let the people touch the

robot to prevent it from being damaged. You need to repeat the instructions about not touching many times, in order to protect the robots. 70% of the children will follow the instructions.

Surroundings:

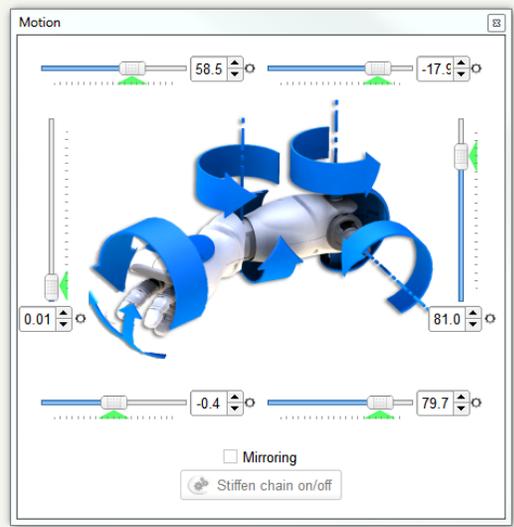
The table we are given is not good. It just has a plastic cloth on the table, which will increase the friction force between the table and the robots. When Ted bowed, and stood up, he would fall down. We had to hold the robots once they start to move. Future exhibitors should take a smooth pad along. That will definitely help a lot. PS: if the event is still in Compton, please dress warmly because it is cold. Since most of the audience is young children, be sure to use songs familiar to them, which may grab their attention. Also, the venue is quite noisy so next time the exhibitors should make the volume as high as possible. Do not rely on speech recognition.

Helpful Nao Learning Lessons from Previous Students

- Facial Recognition using Switch Cases
 - http://www.youtube.com/watch?v=QXmgF_L9wXE
 - At first, the NAO would recognize Person A immediately after Person B and would try to run both strings of actions at the same time. It would often get stuck in a perpetual loop if it continues seeing one of their faces. Both of those not entirely ideal events occurred randomly, so we could not figure out why it would either get stuck in a loop or recognize one person after immediately recognizing the other. It is helpful to duck out of the way right after it recognizes a face. The YouTube video we based our program on did not seem to have that problem.
- Animation Mode: Motion Widget vs. Manual Animation + Curves
 - Manual Animation
 - We found it is a lot easier to do the general and/or bigger movements manually by with the animation mode. This is done by:
 1. Unstiffening chains in the arms or legs (found in the Motion Widget)
 2. Moving the limbs to the position we want
 3. Right-clicking on the Motion Ruler at the top in the Timeline box
 4. Clicking “Whole Body,” “Head,” “Arms,” or “Legs” under “Store joints in keyframe” depending on what parts of the body we had in the correct position.

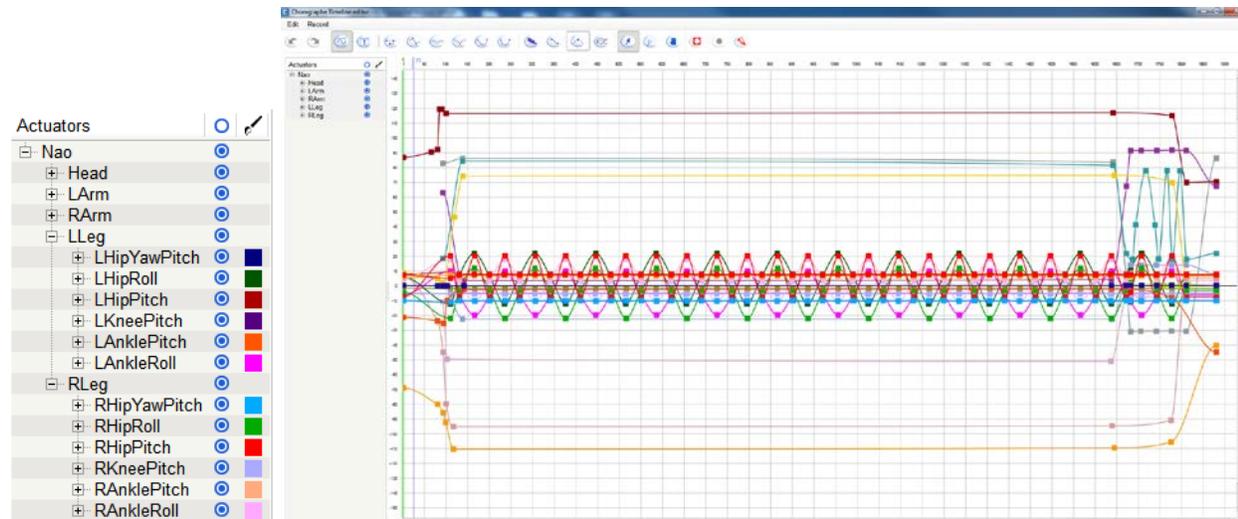


- Motion Widget
 - We would then tweak the newly created keyframe using the Motion Widget.



- Curves

- If we wanted smoother movements or to get a wider view to more quickly correct multiple keyframes, we would use the Curves Timeline editor. To get there,
 1. Click the pencil-like button next to “Motion” in the top left in the Timeline box.
 2. Click the third button from the left under “Edit Record” which shows a curved line. On mouse-over, that button should say “Curves.”
- We liked to color code the curves. If the Shoulder Roll on the Left Arm is dark blue, we make the Shoulder Roll on the Right Arm light blue. That made programming mirrored or similar movements a lot easier.
- Within Curves, you can choose to only see specific joints’ movements, which makes things a lot easier if you only want to adjust the Left Leg’s movements or those of specific joints within the Left Leg, for example. To do so, click whichever appendage you want to edit in Curves and either Shift or Ctrl click to select more. Curves will only adjust the last one you click, but you will be able to see all of them on the screen. Use the Ctrl key to change which joint you want to edit so that you don’t have to start over with the highlighting.



- Mirroring
 - One of the more helpful parts of the Motion Widget is the “Mirroring” option. This allows you to make the NAO’s arms or legs mirror each other (Right Arm will adjust to the same joint settings as Left Arm, for example). This is really helpful for dances (see the Macarena) or anything that might require the same sort of movement on both sides (see the 1812 Overture).
- Adding Music
 - Keep all your MP3s in one folder. It’ll make your life a lot easier.
 - Use Audacity to edit clips, including their length.
 - Choregraphe does not let you start a Timeline at a specific point in the music. Be prepared to hear a song about 100 times (a slight exaggeration) if you want to adjust the timing of a movement in the middle or towards the end of a song. Programming the Alma Mater and the Cupid Shuffle timing made us never want to hear those songs again.
- Fast Movements
 - Because the NAO has to constantly adjust his center of gravity while moving, he cannot do very quick movements with his legs. You must leave enough time between Timeline Keyframes for the robot to catch up.
 - Use Curves to make movements less jerky.
 - Even quick arm movements will throw the NAO off due to momentum. If you are going to do quick and big movements (see our Celtic Chant project), it is best to keep the NAO kneeling and be prepared to hold him because he will rock a little regardless. It isn’t the best visually, but nobody at the fair realized it or at least nobody really cared.
- Balance
 - The NAO has immense balancing capabilities, but (again) they must be with slow movements (see Tai Chi).

- Keep in mind that what works on one foot may not work on the other. We tried to program a slower version of the Jig, and we could only make it work on one foot despite having the same exact movements on the other foot. You will get frustrated. Don't worry. Just move on.
- If you want to lift one of the NAO's feet/legs, act it out yourself and see how your body compensates for the shift in weight. Have the NAO lean more to the opposite side and squat a little on that leg. It is a lot easier to make the other leg lift in different keyframes rather than one sweeping motion. The best is if he can first move his ankle so that his foot is on its tip, then lift his knee, and then kick outwards and upwards. That usually (but not always) works. This is all personal preference, of course, but that sort of movement tended to work best. We did not include any leg-lifting projects in the final ones because they did not tend to work well, but you can look at the various Jig projects to get an idea of what we mean. They are located at the folder under C:\Users\emotecog\Desktop\Practicum in Robotics 2015_Carina and Jeff\Working Projects.
- Wait
 - The Wait Box is your friend. It helps you time things correctly so you can make it seem like the two robots are speaking to each other when in reality each one is just waiting a certain amount of time before speaking.
 - See the Instagram video: <https://instagram.com/p/1TiOmsDLEA/>
 - We used it in the Big Hero 6 Fist Bump, and it helped us know exactly when the robot would move so we could warn the kids at the fair.
- Tactile Sensors
 - The foot bumpers are usually the best and most reliable.
 - If you have DOMER open on two computers, you can run the same program on two different NAO robots. Use the bumpers as cues so that neither of the programs can continue running until you touch the robot. This allows you to ensure that the timing is correct and synced up between the two robots.

Learning Resources:

- We spent a lot of time looking at the Evolution of Dance YouTube video to get ideas for the project and to see what the NAO might be capable of.
 - <https://www.youtube.com/watch?v=2laujomh0JY>
- The YouTube video "Choregraphe: NAO's Facial Recognition" was helpful, but we still had problems that we couldn't solve (see first Lesson Learned above).
 - https://www.youtube.com/watch?v=QXmgF_L9wXE
- The Aldebaran Community has a forum that could be very useful. We did not need it most of the time, but it is a good place to look through to find ideas and problems other NAO programmers have run into.
 - <https://community.aldebaran.com/en/forum>

- You can contact Aldebaran if you have any questions about Choregraphe, but they might not be able to answer specific coding problems. They usually tell you to check the forum if they cannot figure it out. Every person we spoke to was very nice and really tried to help.
 - <https://www.aldebaran.com/en/contact>
 - Phone number for US office in Boston: +1 (617) 986-6700
- Mostly, we figured things out through trial and error. That is how we work best, so we did not look up tutorials all that often because we could usually figure out the problem fairly quickly. Carina was better at general movements and ideas but Jeff could pinpoint little problems that were causing a lot of issues, and that sort of teamwork was incredibly helpful.
- Otherwise, a simple Google search will solve most issues fairly quickly.

iPhone APPS for controlling Nao:

You can download the apps directly from the apple store or you can download them from the app store from the official website of Nao. PS: you will need a registered account for the Nao, so that you can log into the Nao apps store, which means that only the Nao holders can sign in that account. I download two apps, which are 'iControlNao' and 'NAOControl'.

- iControlNao: it is free, but you need connect to the same wifi as the Nao is connected to. If both robots are on, you will find two devices: Ned and Ted. For this app, there are some default settings for Nao, like standing up, sitting down, turning left /right and going backward/forward. You can upload the behavior files to NAO, so that you will be able to see there will be the list of behaviors from the app. You can choose any one you want to run, but the problem is that once you start the behavior, you cannot stop it. You also cannot dis-engage the robots' motor, so it will be always engaged if you do not have the programs like motor on/off. Also the wifii intensity may fluctuate, so that the robot barely reacts.
- NAOControl: Not free, costs \$6. The app has all the functions that iControlNao does. But if you have a robot account, then you can log in the Nao app store and download more applications like math, chess, and other things. If you install the apps on the robot, with "NAOControl", you can also control all the behaviors already in the robot.