

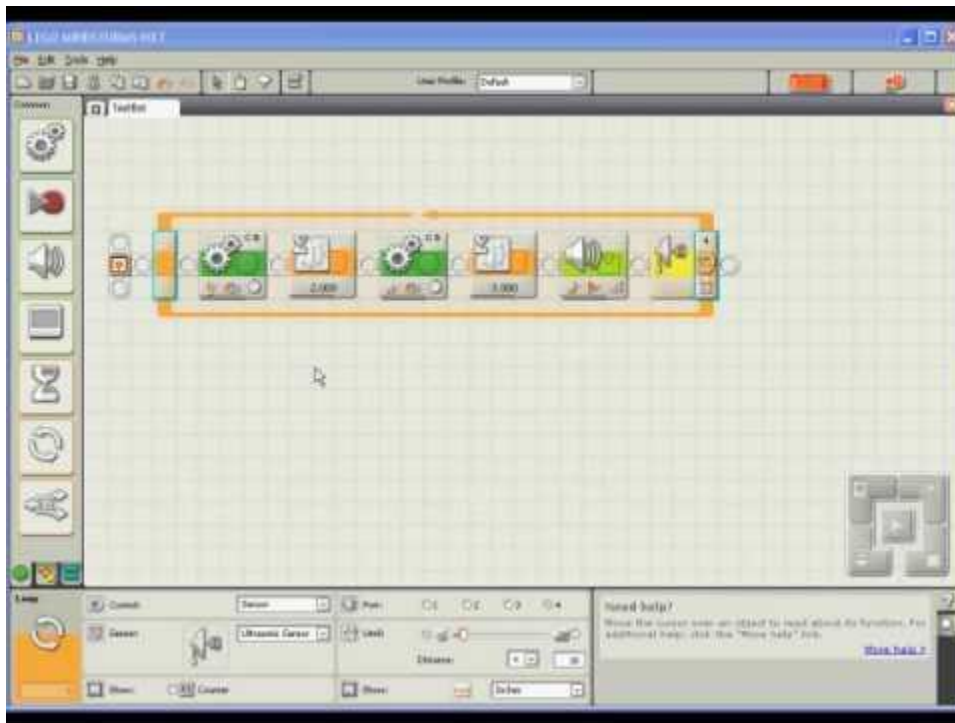
**Two Week Project GUI INTERFACES**— This is going to be very time consuming and hard to do. I'm not sugarcoating it. It will take all three partners to carry their load, you won't finish with one doing all the work and the other partner adding emotional support.

**Demos will be April 22<sup>nd</sup> between 11am and 3pm (what is expected in demo is below)**

Make a user interface on the touch screen that will allow a user to program the robot to do a series of events.

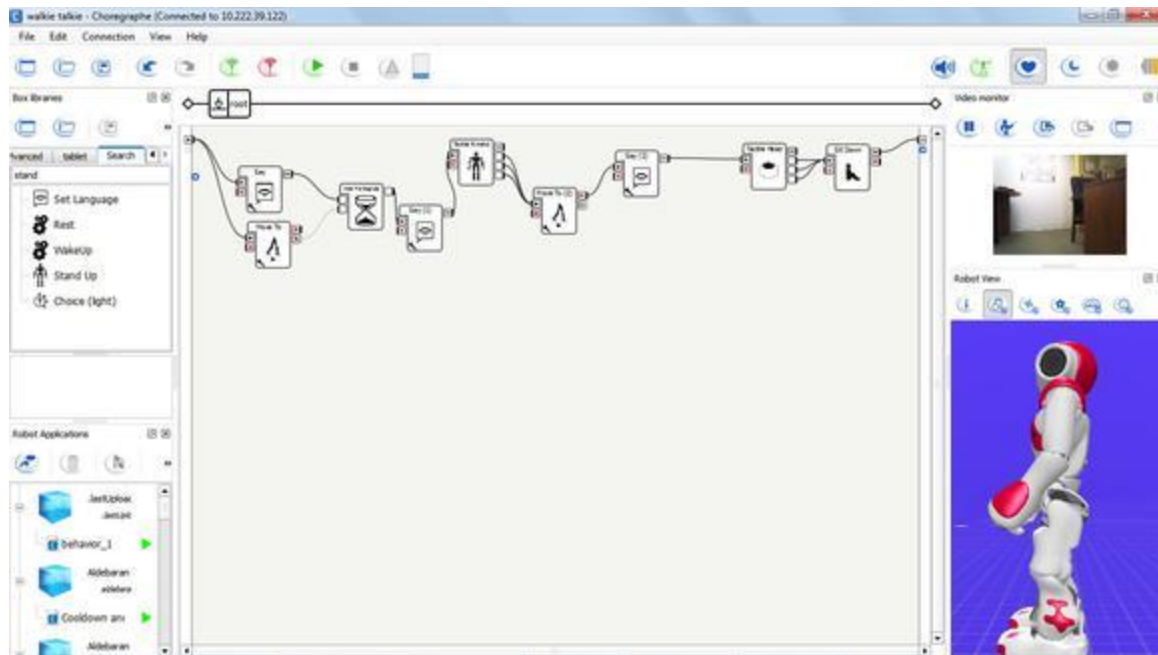
Here are some examples GUIs from professional robots, you will build something similar, but you don't have to copy these, you get to design your own:

NXT Software:



In this image the tools needed such as motors, sound, screen, loops and switch statements are represented by icons. When an icon is dragged on to the workflow line it becomes an event the robot will execute while this program runs. Each icon on the workflow line has properties that show up in the bottom left pane when the icon is selected. For instance, the motor icon has properties that will show up for the speed and direction the wheels should turn.

Here is another example:

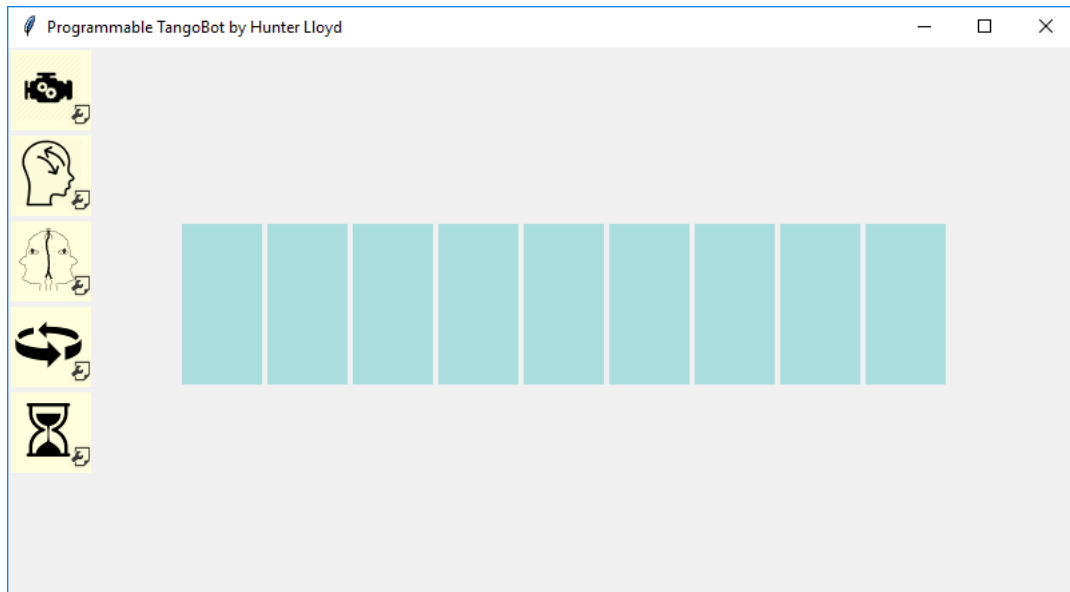


This is Choregraphe, I have shown in class several times. This is the GUI drag and drop that allows you to program a NAO robot or a Pepper robot.

The library of icons is vast, over 80 in my library (they can be custom created and programmed by user). You can drag them out and drop them on the stage and then link them by wires. This program allows you to run two icons at the same time, multithreaded, you will not have to do that for our assignment.

The command library is in the top left of the screen, under the tabbed pane that is currently on "search". You can drag and drop the icons to the screen and then the properties are setup by clicking the little wrench in the bottom left corner.

Now I have combined these two GUI's and created my own that will fit in the touch screen for the Tangobots.....here's a quick look at mine.....much simpler, but still pretty complex:



*Yes, I know I need to work on the head turn graphic, but the rest look decent.*

I have five icons that can be dragged to the nine spots on the timeline. From top to bottom MOTORS, HEADTILT, HEADTURN, BODYTURN and PAUSE. I use binds to listen for mouse controls to drag, drop and click.

These are the first five I decided to work with at this point.

I did this example in Tkinter, but I am now going to switch it to Kivy, I like Kivy better. I showed numerous examples of both in the lectures this week. You can use whatever GUI package you want as long as it works on the touchscreen of the TangoBots. Some people prefer HTML graphics, which I think is ridiculous, but I'm not going to stop it.

Each has a wrench icon in the corner which will open the properties dialog for that icon. The properties dialog will not open until the icon has been dragged (copied) to the timeline. Once the icon is copied to the timeline it becomes its own instance for that icon.

For instance, the motor icon has speed, direction or it allows turns. Right now I'm just using a variable for the amount of seconds you want this task to execute.....but as some of you already found out the distance travelled in a particular time is based on the battery charge. Two seconds on a low battery might be two feet, but on a full battery might go across the building. We don't have any encoders for determining turns of the wheels, but that would be a more accurate determination of distance. For this class "time" is fine (assume batteries will be full).

The tilt and turn icons you can use a determining value so it moves in the correct direction and distance.

The sleep icon is nice for pauses and dramatic effect.

I didn't think about it until now, but another icon for graphics on the screen while running would be nice too.

## YOUR EXPECTED DUTIES:

Each team will need to create your own GUI to program the robot with the touch screen. You will need to be able to implement an 8 instructions on their timeline for full credit. The instructions (icons you will place on your timeline) you will need are

1. Motors with speed, time and direction.
2. Motors turn robot left, or right for x amount of seconds.
3. Head tilt both directions
4. Head pan both directions
5. Waist turn both directions
6. A wait for human speech input
7. Talking, be able to type in what sentence you want to say and the robot says it.

That is seven icons and your timeline must take 8 instructions.

No multithreading (multiple instructions at once) need to be done.

Full credit will be 100 points.

1. Being able to complete my set of commands 40 points, each command you can pull off is 5 points ..... Example: Go forward 5 feet, turn right, go forward 7 feet, stop, look right, look left, turn 90 degrees left, say hello, go forward 2 feet, back up 2 feet, twist body full to right.....etc. etc.
  - a. I'll set up an obstacle course for you to attempt to complete.
2. 30 points is for your GUI look and feel, ease of use and no errors on just using the GUI. There must be a play/start button of some kind that will run through the commands that have been programmed in order. For full credit you must be able to delete and clear programs or pieces of programs the user has developed.
3. 10 points is for an animation you develop while the bot is running the program. It can be whatever makes sense that a user might like, it can be letting the user know where you are in the program, or just something that makes sense.....has to be an animation, and gives your robot a little personality.

This is for the people that just can't get the 8 steps working in the step one above.....

4. The last 12 points is if you can get the robot to dynamically any set of two commands you have.....this is not the real dynamic plans I give you, but just any commands that can vary slightly....."a single icon that allows you to move forward x amount of seconds, and then back up y amount of seconds" as long as the values for x and y can dynamically be input.

Because batteries will be different values you can try it a few times to get five feet down to your timing.....but the timing has to be something dynamically input at program time.

These next two rules are so you get creative with your touch screen input.....

- If you must use a mouse to program the robot you lose 10 points.
- If you must use a keyboard to program the robot you lose 20 points.

All executable commands must be activated by one push of the start button. One push moves the robot through all 1 to 8 commands in the order the user programs.

#### **HELPFUL HINTS**

- Use Object Oriented Programming techniques. Keep all the properties for each command in instances which will allow you to easily go through them 'in order' to execute the commands. All my icons and locations on screen are instances of objects for which ever element they are. It's easy to keep track of things and reuse code that way.
- Using OO will need plenty of up front planning, do not just start typing, draw out your GUI, what needs what classes, ect. ect. Use Software Engineering skills.
- DON'T WAIT, start immediately. I did most of my GUI programming on my laptop not anywhere near a robot. You don't have to be in lab to do a lot of this programming.

There is no right click on the touch screen without a mouse.

Demos Friday, April 22<sup>nd</sup>.