**Description:** A child’s piano to help toddlers learn how to play their favorite songs on the piano.

**Requirements:**

1. The project should help with child sensory development and be an engaging learning activity for toddlers.
2. It should use at least 5 force sensor resistors (FSR), also known as pressure sensors, and not buttons.
3. Preloaded with at least 2 songs that can be played in full or a preview for the current position in the song by utilizing a passive buzzer. *Note: maybe include a button to cycle through the songs?*
4. LED lights will indicate which note to play. Color coordination would be helpful.
5. Must be coded in java.
6. *Optional*. Since this will be a child’s “toy”, a case and protective sleeves for the wiring would be great to have.
7. *Optional*. A GUI would be nice to display the keys in addition to the LED lights.

It has taken me a while to figure out what I was going to choose for my project. I wanted something that could be useful and beneficial in my life instead of just another gizmo that I created for a grade. I have two kids under the age of 3 who require a constant flow of educational toys or else boredom sets in. All the parents out there know that you should not let boredom set in with toddlers. There will be a vast amount of destructive behavior served with tantrum McFlurries covered in the shrilling sounds of “No!” and “I don’t want to!” With this in mind, I decided to create the *Pi.ano* (working name).

The Pi.ano is a toddler friendly piano that is both educational and engaging. It will teach toddlers how to play their favorite songs by stepping through each note and allowing them the time to press the corresponding key of that note. It will contain two preloaded songs that can be played in full or a snippet of the current position in the song.

As for the hardware, Pi.ano contains 5 force sensor resistors (FSR) strips that act as piano keys which will be color coded, a passive buzzer for playing a keynotes and playing preloaded songs, and an LED light color-coordinated with the keys that will indicate which key to press next. As for the software, it will be coded completely in Java. The reason for this is simply because I need to gain more experience with Java for my new software engineering job.

One challenge I foresee is loading songs from memory to play. I considered using JDBC but this seems like overkill for only playing a few songs. This might be best for the long term implementation when the song list grows. The next question is then how do I load songs? What file do I use? Do I load an audio file and figure out how to play that file through the buzzer? I think the best solution for this purpose is to simply write each note and delay time in a text file and read from that file. It might take a little effort up front, but since kid’s songs are not very long I do not anticipate this being much of a time sink.

The only issue I can see having is if Java has issues interfacing with any of the hardware components. I am hoping this should not be an issue, but before I begin coding I will make sure I do some initial research. Otherwise I may need to switch to Python or at the minimum call a Python script from the Java code if is required at that time.

Since the goal of this project is to make it kid friendly, I will be looking into safety features such as a case for the pi and wiring protections. I might place cardboard cutouts onto of the FSRs to act as keys so the kids are not pressing directly on the component. However, since this is a “prototype”, I do not anticipate it to be child safe and will require adult supervision.