6.005 Elements of Software Construction
Fall 2008

6.005 Elements of Software Construction | Fall 2008
Exploration 1: A Midi Piano

The purpose of this exploration is to give you practice coding in Java, and to introduce you to the state machine case study that will be presented in lecture. Your solution will be judged by (1) correctness -- whether it meets the specification, and (2) clarity -- whether your code is well-organized, carefully commented, and makes proficient use of Java. You are *not* required to use any of the ideas or patterns that have not yet been taught in lecture.

Specification

Your task is to build a program that allows a standard computer keyboard to be used as a (rather crude) electronic piano with the following features:

- The notes of the octave above middle C are played on the numeric keys, starting with 1 on the left and ending with the = key on the right, with a semitone between each (so that each pair of adjacent keys corresponds to a pair of adjacent keys on a real piano, including the black ones).
- Holding a key down sustains the note, and several notes can be played at once.
- Different instruments can be chosen. Pressing the I key cycles through the available midi instruments.
- You can record and play back sequences of notes. Pressing the R key starts recording, and pressing it again stops recording. During recording, a red indicator is shown; this turns back to green when recording ends. Pressing P causes the last recorded sequence to be played back. Note that playback should preserve timing.

Hints

Your keyboard driver will probably generate repeat presses when a key is held down. You'll need to make sure these are filtered away so that the quality of the sound isn't damaged.

We recommend that you implement your program as a Java applet. We are providing you with skeletal code that shows you how to bind methods to key presses, and how to access the Java midi API. You may of course use this code in your solution.

To implement the color indicator, we recommend that you simply change the color of the applet's background, as shown in the sample code.

A slight annoyance: you'll need to click on the applet's panel with your mouse to give it keyboard focus before playing any notes.

Optional Challenge

If you've successfully completed the exploration, you may want to try this optional challenge. Allow the P key to be pressed during recording, so that recordings can be layered, with new notes played being added to notes that were previously recorded. (Warning: this is tricky!)

Addendum

Some Java installations do not have a default soundbank installed, causing the Midi code to throw the exception

```
java.lang.NullPointerException
    at com.sun.media.sound.AbstractPlayer.loadAllInstruments(Unknown Source)
    at midi.Midi.(Midi.java:43)
    at piano.PianoApplet.init(PianoApplet.java:24)
    at sun.applet.AppletPanel.run(Unknown Source)
    at java.lang.Thread.run(Unknown Source)
```

This can be corrected by following Sun's instructions here.