

**Drum Machine 1**  
Assignment due 18 February 2008

**0. Reading**

- a. <http://chuck.cs.princeton.edu/doc/language/> :  
Read the sections on Arrays, Operators + Operations, and Control Structures
- b. Simple “on-the-fly” synchronization  
Look at (and run!) the otf\_01 to otf\_08 examples at  
<http://chuck.cs.princeton.edu/doc/examples/>
- c. SndBuf: A UGen that lets you use your own sounds.  
Look at [http://chuck.cs.princeton.edu/doc/program/ugen\\_full.html#sndbuf](http://chuck.cs.princeton.edu/doc/program/ugen_full.html#sndbuf)  
and <http://chuck.cs.princeton.edu/doc/examples/basic/sndbuf.ck>

1. Download the drum machine skeleton code.
2. At the “\*\*\*”, add a loop control structure to play the rhythm forever.
3. Add another loop control structure to iterate through beatArray. Within the loop, you should be able to play a note by  
`beatArray[i] : quarter => now;`

Hint: Pay attention to the legal values of “i” in this statement. Note that `beatArray.size()` will tell you the number of elements in the array.

4. Use one or more of your own sound files instead of the ModalBar (see the SndBuf reading).
5. Now make it interesting. Start with a more interesting rhythm. Vary the dynamics, “instrumentation”, pitch, tempo, and whatever else you can think of. Feel free to add other arrays for these other properties.
6. Now make it rock. Play different versions of your code from Question 5 concurrently in miniAudicle. These could be different versions saved in different files, or you could live-code and edit the file on-the-fly. Make sure to set each shred’s synchronization appropriately (by changing from 1-second intervals to something else).

Q: Explain how the synchronization code below works when running multiple shreds concurrently.

```
x::second => dur T;  
T - (now % T) => now;
```

7. Think about how you could use arrays and loops to beautify and generalize your “twinkle, twinkle” code. Implement this if you think it would be helpful. Aren’t loops and arrays nice?

**What to hand in:**

- Your ChuckK file solution to Question 4, along with any sound sample files you’ve used.
- Your musically compelling ChuckK file(s) solution to Question 6 (includes work for Question 5), again along with any extra files.
- Your written component for Question 6.
- Your modified twinkle code (not required, but we’ll give you feedback if you did this part).
- Submit the above via Blackboard.

Be ready to perform your drum machine from Question 6 in class!