

Twinkle, Twinkle
Assignment due 11 February 2008

0. Reading

a. Types, values, and variables

<http://chuck.cs.princeton.edu/doc/language/type.html>

b. Unit Generators in ChuckK

<http://chuck.cs.princeton.edu/doc/language/ugen.html>

c. Manipulating Time in ChuckK

<http://chuck.cs.princeton.edu/doc/language/time.html>

1. Download the “Twinkle, twinkle” skeleton code. As is, the code plays the first note of the song. Add code to play the rest of the melody, with the right notes and rhythm.

(Are you rusty on your Twinkle skills? See

http://schools.tdsb.on.ca/joyce/main/pathfinder/download/chart_scoretwinkle.pdf
for a score.)

2. Spice up your twinkle performance by replacing the SinOsc with another oscillator and/or STK instrument. See the ChuckK UGen specification at <http://chuck.cs.princeton.edu/doc/program/ugen.html> to see what parameters are available for the different oscillators and instruments, and don't forget to use `.noteOn` and `.noteOff` for STK instruments.

3. Inheritance

Many of the UGens **extend** other UGens. For example, the STK instrument **PercFlute** extends **FM**. This means that all **PercFlute** objects are *also* **FM** objects, and they *inherit* all of the parameters and functions of **FM**. **FM** is therefore a “parent” and a “superclass” of **PercFlute**, and **PercFlute** is a “child” and “subclass” of **FM**. So, if you want to know all the parameters and functions available to a ChuckK object, be sure to look at the documentation for that object *and* its parent (and its parent's parent, and so on...)

N00bs

Inheritance is a core feature of object-oriented languages like C++ and Java. Think about practical and conceptual reasons for inheritance.

1337 Haxors

ChuckK currently does not support multiple inheritance. Is this a moral outrage?

What to hand in:

- ChuckK code for your Twinkle, Twinkle masterpiece for Question 2.