Song Generator Nifty Assignment
SIGCSE 2010 Nifty Assignments Panel

Dan Zingaro
University of Toronto

March 13, 2010

Sound Processing

- We frequently have nifty graphics-related assignments in this session (image manipulation, animation, video games . . .)
- But, trust me: sounds are way, **WAY** cooler than images
- Media computation approach should use sounds as well as images
  - Appeal to students with different strengths

Filters on Sounds

- Digital sound: one-dimensional list of integers ("samples")
- We have a Python sound library that makes sounds accessible as lists of samples
- Students use `get_sample(i)` to get a sample object at index `i`
- Sample objects have `get_value` and `set_value` methods
- With this, several filters can be written using loops
  - Change volume of a sound
  - Mix sounds together
  - Add echo to a sound ...

Why Filters?

- Students write those filters so that
  - They practice writing code with loops
  - They get familiarized with our media library
  - They understand how digital sounds are stored and manipulated
  - They can use them in a larger program to generate songs

Song Generator

```
[1150]4e4g8g8g4g8g4a8f12f4p4e4g8g8g4g8g4a8f
12f4p4e8e12d8d4f8c12c4p4c8e8e8d8d4f8c12c
```

- What's that thing?
- It's a notestring for really popular song. Ready? ...
- Notestrings specify a language for songs: rests, notes, octaves, beats per minute, volume, channels
- The assignment handout assumes no knowledge of music: it's "just" a string-processing exercise

Multiple Channels

- So far we've heard one note at a time
  - Like QBasic `play` or Nokia RTTL
- Channels are the coolest feature of notestrings
- We separate channels in a notestring with a |
- Channels let us play multiple "hands" simultaneously
- This gives us just enough flexibility to painstakingly craft real songs . . .

Variations

- Different sound filters
  - Fade in/fade out,
  - Echo with "number of echoes"
- Support superset of RTTL
  - RTTL: ringtone text transfer language
  - e.g. `d=4,o=5:2c,2e,2g,2c6`
  - Complicated by default note length and default octave parameters, dotted notes, sharps/flats

Why Nifty?

- Students learn about the representation of digital sounds
- They make use of filters in their song generator — compelling example of function reuse
- They enjoy incrementally developing their song generator to support increasingly complex songs
- Musically-inclined students can develop and share songs
- No images!