Multimedia Construction Projects for CS1/2

Mark Guzdial
College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332-0280
guzdial@cc.gatech.edu

Abstract

Media computation can be a motivating and creative domain for examples and assignments in first CS classes. Simple and obvious algorithms run in reasonable time on modern computers. In this tutorial, we will cover both algorithms and working code for creation and manipulation of sound, image, and video data. Example code will be presented in Smalltalk, Python, and Java.

Tutorial Category: Half-day

1 Tutorial Description

Multimedia construction can be a motivating and creative domain for examples and assignments in computer science classes. Because of the speed and capacity of modern computers, simple and obvious algorithms that fit even at the CS1 and CS2 levels run in reasonable time. In this workshop, we will cover both algorithms and working code for creation and manipulation of sound, image, and video data. Example techniques will include sound synthesis, chroma key ("blue screen") image manipulation, animation, and Photoshop-like effects. Example code will be presented in Smalltalk (Squeak), Python (Jython), and Java.

Participants will see many examples of multimedia construction and manipulation that can be directly used in examples and assignments in courses including CS1 and CS2. The advantage of the multimedia domain over other kinds of projects is that it has concrete and interesting results that are easily related to CS concepts. For example, sound is just an array of samples (numbers), so concatenating arrays or moving around sequences within an array leads to results students can hear. (Bugs literally "shout out" to you.) The order of media covered in the tutorial is arranged to correspond to an increasing level of complexity in data structures.

- A sound is an *array* of *samples.*
- A picture is a *matrix* (two-dimensional array) of *pixels.*
- A directory structure (of media files, to process many files with a single recipe) is a *tree* of files.
- A movie is an *array* of *matrices* (frames, as pictures).

The media thus serve as a way of visualizing and making concrete (and interesting, we believe) the programs that the students are writing.

All the examples are based on coursework developed at Georgia Tech. The Squeak examples will be based on a recent *Introduction to Computer Music* course taught at Georgia Tech recently (e.g., [2]) and on follow-up work exploring additional media [4]. We have developed a Java API for media computation usable in both Java and Jython. We are developing the Jython examples for a new CS1 course for non-majors, *Introduction to Media Computation.*

Our API allows for access to the samples that make up sounds and the pixels that make up pictures.

- Figure 1 is an example program using our API that converts a picture object to greyscale. It computes the intensity of a given pixel by averaging the red, green, and blue components, and then replaces the color of that pixel with a gray pixel (red, green, and blue components the same) with the same intensity.
- Figure 2 is a program that normalizes sounds to a maximum volume, by searching for the largest sample, computing a multiplier so that that sample would reach the maximum amplitude, and then multiplies all samples in the sound to raise the amplitude of the overall sound.

```
def greyScale(picture):
  for p in getPixels(picture):
    intensity = (getRed(p)+getGreen(p)+getBlue(p))/3
    setColor(p,makeColor(intensity,intensity,intensity))
```

Figure 1: An example Jython program using our API to convert a picture to greyscale

```
def normalize(sound):
    largest = 0
    for s in getSamples(sound):
        largest = max(largest,getSample(s) )
    multiplier = 32767.0 / largest

    print "Largest sample value in original sound was",  largest
    print "Multiplier is", multiplier

    for s in getSamples(sound):
        louder =  multiplier * getSample(s)
        setSample(s,louder)
```

Figure 2: An example Jython program using our API to normalize sounds to a maximum volume

Participants will receive a CD containing:

- All the media and code used in the presentation
- Jython and Squeak implementations for common platforms (assuming that those interested in Java will already have Java implementations)
- Powerpoint slides from the presentation
- Other Powerpoint slides used in our classes using these materials which can be reused in the participants' classes

The workshop could be offered hands-on, but given the scope of media, techniques, and languages proposed, a presentation format will be more efficient.

2 Audio/Visual/Computer requirements

I will need a projector and speakers (built in to the projector or not). I will provide a laptop. (If no projector or speakers is available, I can bring them.

3 Background of the Presenter

Mark Guzdial is an associate professor in the College of Computing at Georgia Institute of Technology (*Georgia Tech*). His research area is in computer-supported collaborative learning and computer science education. His interest is in creating *collaborative Dynabooks*, in reference to Alan Kay's vision of computation for learning through multimedia creation and exploration. He is the author and co-editor of two recent books on using Squeak for multimedia [2][3]. His dissertation work was on an environment for high-school students to learn programming through programming multimedia and simulation [1]. He is currently developing a large-scale (400-600 students/semester), non-majors CS1 course on *Introduction to Media Computation.*

References

[1] Guzdial, M. Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments 4*, 1 (1995), 1–44.

[2] Guzdial, M. *Squeak: Object-oriented design with Multimedia Applications*. Prentice-Hall, Englewood, NJ, 2001.

[3] Guzdial, M., and Rose, K., Eds. *Squeak, Open Personal Computing for Multimedia*. Prentice-Hall, Englewood, NJ, 2001.

[4] Guzdial, M., and Soloway, E. Teaching the nintendo generation to program. *Communications of the ACM 45*, 4 (2002), 17–21.