Τέχνη : Perspectives and Directions

Mike Westall

Professor of Computer Science

Clemson University

westall@cs.clemson.edu

http://www.cs.clemson.edu/ westall/homepage.html

Preliminaries

What is τέχνη ?

Preliminaries

What is τέχνη ?

- Τέχνη is the Greek word for *art*.
- It is also the root of the word τεχνολογία, the Greek word for *technology*.

Preliminaries

What is τέχνη ?

- Τέχνη is the Greek word for *art*.
- It is also the root of the word τεχνολογία, the Greek word for *technology*.

Τέχνη is also the name given to a multi-year endeavor that has been funded by the NSF since 2003. The goal of the Τέχνη project is to re-establish the connection between art and technology and in so doing to improve learning in undergraduate computing curricula.

The pre-Τέχνη curriculum

- CS1 - Java / programming with objects early
- CS2 - Java / object oriented design
- CS3 - Java / data structures
- CS4 - C/C++ language course

Assignments were typically topic driven.
For example:

Read the class roll inserting student names into an AVL tree in first name order and then print them.

Existing problems - possible causes

Definite problems

- High attrition rates at a time of declining enrollment
- Demonstrably poor programming skills in upper division courses

Possible causes

- Toy programs were uninspiring to students and faculty alike
- Faculty with expert programming skills were not involved in CS1 and CS2
- “Abstraction first” paradigm frustrated the average student and turned the best students into “experts” who actually understood almost nothing

The prototype course

Raytracing in the existing CS4

- “They” said it couldn’t be done
- But the problem is superbly suited to motivate O-O design and implementation
- The best students often went above and beyond the assignments

The success of the prototype motivated the submission of an NSF proposal that was funded in the summer of 2003.

Present directions

In June 2007, the Τέχνη project received three years of addtional funding from the EAE (Extension, Adoption, and Evaluation) component of the CPATH program of the NSF. This funding will support:

- Extension of the Τέχνη approach to upper division courses at Clemson
- Refinement of existing Τέχνη courses
- Adoption of elements of the Τέχνη approach at partners WCU and UNC-W
- The development and application of more rigorous evaluation (assessment) mechanisms

The Τέχνη Vision

We envision a curriculum that

- attracts, retains, and challenges the very best students,
- motivates average students and provides them the skills they need to contribute to the rapidly changing computing field and,
- inspires faculty to stay abreast of leading-edge applications of computing.

Underlying Theories of Learning

The Τέχνη approach is based upon a melding of

- Piaget's constructivism with
- the cognitive apprenticeship as described by J. R. Anderson, Bandura, Vygotsky and others.

Constructivism as seen by Piaget

Piaget’s views as described by Edith Ackerman:

- Teaching is always indirect. Kids don’t just take in what’s being said. Instead, they interpret what they hear in the light of their own knowledge and experience. They transform the input.
- The transmission model of human communication won’t do. To Piaget, knowledge is not information to be delivered at one end, and encoded, memorized, retrieved, and applied at the other end. Instead, knowledge is experience that is acquired through interaction with the world, people and things.
- A theory of learning that ignores resistances to learning misses the point. Piaget shows that indeed kids have good reasons not to abandon their views in the light of external perturbations.

The Τέχνη Perspective on Constructivism

Westall's view: *"knowing stuff"* is fundamentally different from *"knowing how to do stuff"*.

- Useful knowledge must be constructed by the learner
- Software development skills, like piano, golf, or welding skills must be learned and truly can't be taught
- But an appropriate level of "directional guidance" and support in problem resolution is essential for correctness and efficiency of learning.
- Repetition (practice) is essential for reinforcement and the long-term retention of knowledge.

Another Perspective on Constructivism

The "new New Math" a.k.a. "Everyday Math"
http://www.lit.net/orschools/everydaymath2.htm

- Encourages:
  - working in groups
  - discovery of problem solving techniques (how to multiply 2 digit numbers)
  - familiarity with (but not mastery of) of multiple techniques for solving a problem.
- Discourages:
  - direct instruction (lecturing) by the teacher
  - memorization of algorithms (multiplication/long division)
  - memorization of multiplication tables

At the college level a variant of this approach has been called scale up

Unconstrained constructivism

The student may construct:

- incorrect “knowledge”
- inefficient solution techniques
- correct solutions that are not appropriate for harder problems
- correct solutions... but take an excessive amount of time to do so.

The Cognitive Apprenticeship

Serves as a counterforce against constructivism run amok. It is based upon a continuing feedback loop:

- Craftsman demonstrates and apprentice observes
- Apprentice attempts the task and craftsman corrects errors
- Craftsman maintains control over project direction as apprentice constructs knowledge
- As skill level improves, the apprentice is assigned tasks of increasing difficulty
- This approach is consistent with Vygotsky’s zones of proximal development
- It is applicable not only to aspiring welders, but also to aspiring golfers, pianists, and software developers.

Implementation in Τέχνη

- Problem based instruction with problems taken from the visual domain
- Other domains certainly possible – aural proposed in Τέχνη I
- The focus is upon computational modeling of physical systems
- NOT “TEACHING GRAPHICS”
- Use of semester long open-ended projects that motivate the need for particular programming techniques.
- Use of the C Language in CS-1 because it can be directly mapped to the (extended) von Neumann architecture
- The architecture of the solution is provided by the instructor.
- The students construct the details of the solution.

Open-ended assignments in Τέχνη

Open-ended assignments provide a mechanism that

- Challenges best students
- Provides a path to success to the average student
- Encourages creativity

The present Τέχνη curriculum

- CS1 - C / programming with digital image transformations
- CS2 - C/C++ / advanced C and basic C++ with raytracing
- CS3 - C++ / data structures with raytracing, photon mapping and surface reconstruction
- DBMS - / development of a multimedia database.

CS 1 objectives

Learning objectives are not radical. At the end of the end of the course the student should understand:

- how the (extended) von Neumann machine operates
- how to create instances of basic data types
- how to set and reference values
- the implications of stack and heap residence
- the use of arithmetic and logical operators
- the use of looping and conditional control flow mechanisms
- how to create and manipulate arrays
- how to use formatted, byte, line/string, and block I/O facilities
- how to create functions and use them to partition a problem
- how to use dynamic memory allocation and pointers
- how to create and use hierarchically structured data types

CS 1 problem domains

- Introductory problems in counting, searching, accumulating, and recurrences
- Writing and reading of ppm image files
- Basic parsing of strongly constrained input languages
- Basic image manipulations including: resizing, tiling, conversion to grayscale, gamma correction and monochrome conversion
- More difficult image manipulations including: procedural blending, convolution filters, and transfer of color tone from one image to another.

CS 1 example

```
edge
{
   in0 trump.ppm
   out trumpedge.ppm
}
bright
{
   in0 trumpedge.ppm
   out trumpedbr.ppm
   factor 2.0
}
```

```
sobel
{
   in0 trump.ppm
   out trumpsob.ppm
}
bright
{
   in0 trumpsob.ppm
   out trumpsobbr.ppm
   factor 2.0
}
```

CS 1 example

CS-1 Final Project

Image copy	50
Mirror	5
Grayscale	5
Brightness	5
Monochrome	5
Color fading	5
Resize	5
Smoothing filter	5
X-Y Sobel filter	5
Caption	5
Blended composite	5
Color transfer	5
Total	105

CS 2 objectives - C language

The student should understand

- use of function pointers
- use of tables of function pointers to eliminate switch()
- use of function pointers in C "objects" to provide polymorphism
- use of hierarchical structures linked with void * pointers to simulate inheritance

CS 2 language objectives - C++

The student should understand

- that objects are a natural generalization of C constructs
- how to create and use instances of C++ classes
- how to make use of polymorphism and inheritance to minimize code duplication and facilitate reuse
- how to use C++ I/O facilities
- how to override operators in C++
- the use reference parameters in reducing the risk of pointer problems.

CS 2 problem domain

- Basic linear algebra operations: vector add, subtract, scaling, length; dot product, cross product, projection
- Use of rotation matrices as linear transforms in 3-D space
- Construction of generalized parsing routines
- A basic C language ray-tracer including sphere, infinite plane, and finite-plane objects with ambient lighting
- A C++ raytracer with support for tiled, textured, and procedural planes, boxes, surfaces of revolution sky, diffuse and specular point light sources, projector light sources, partial transparency, and refraction.

CS 2 example

```
camera cam1
{
  pixeldim  800 600
  worlddim  8 6
  viewpoint 4 3 6
}
material white
{
   ambient 0 0 0
   diffuse 9 9 9
}
sphere earth
{
   material white
   center 4 3 -8
   radius 5
}
```

```
projector front
{
   emissivity 17 17 17
   location    4 3 8
   material white
   point       2 1.5 3
   normal      0 0 1
   xdir        1 0 0
   dimensions 4 3
   texname ../images/sky.ppm
   mode 0
}
```

CS 2 example

CS-2 Final Project

```
Ambient lighting with planes & spheres  50  50
Diffuse lighting / shadows              10  60
Finite plane                             5  65
Textured planes                          5  70
Spotlights                               5  75
Texture based projectors                 5  80
Specular reflection                      5  85
Specular glints                          5  90
Anti-aliasing                            5  95
Tiled infinite plane of arb orientation  5 100
Partial transparency                     5 105
A non-quadric revsurf                    5 110
                                       ---
Total                                  110
```

Conclusions

Rule 1: to help, or at least to do no harm.

I strongly believe that the Τέχνη project:

- Does not violate rule 1.
- Has inspired many of our best students to achieve at level FAR BEYOND that which was previously the case.

Conclusions

I tend to believe that the Τέχνη project

- has helped retain the students who should be retained
- has helped many students who should be looking for another major to do so after only one semester.
- has given the students a more positive view of the instruction they are receiving
- has given the students a more positive view that they are learning useful skills

Conclusions

On the not so positive side...

- Unmotivated students still exist in significant quantity.
- The problem determination and resolution skills of many students remain very poor.
- Faculty "buy in" has been marginal. I question the sustainability of the approach.
- Rigorous assessment is extremely difficult because of inadequate sample sizes and presence of uncontrolled variables..

(Some) quantitative measures of are available in the papers on my texnh web page.

Acknowledgements

Thanks to the NSF for their continuing support of the project.