Τέχνη : Introduction and Overview

Mike Westall

Clemson University

westall@cs.clemson.edu

http://www.cs.clemson.edu/ westall/texnh/acmse09.pdf

Project Team

Co-PIs

- Tim Davis, Andrew Duchowski, Robert Geist, Pradip Srimani, James Wang, Mike Westall, Clemson University
- Sridhar Narayan, University of North Carolina at Wilmington
- Mark Holliday, Bill Kreahling, Western Carolina University

External evaluator

- Bob Schalkoff, Dept. of Electrical and Computer Engineering, Clemson University

What does Τέχνη mean??

- 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.
- One 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.
- We believe that design in the artistic sense promotes computational thinking in our problem domains.

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.

Present Τέχνη directions

In June 2007, the Τέχνη project received three years of additional 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
- Adoption of elements of the Τέχνη approach at partners WCU and UNC-W
- The development and application of more rigorous evaluation (assessment) mechanisms

Elements of Τέχνη have also been employed at Covenant College and

Bob Jones University

Underlying Theories of Learning

The Τέχνη approach is based upon a blending of

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

The Τέχνη Perspective on Constructivism

- 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 (as in piano, golf, and welding).
- Repetition (practice) is essential for reinforcement and the long-term retention of knowledge.
- Codelab from TuringsCraft is a fine tool to support this

Other Approaches to Constructivism

All constructivists don’t share a common world view.

- For example, the "new New Math" a.k.a. "Everyday Math" (http://www.lit.net/orschools/) tended to favor unguided (or self-guided/peer-guided) constructivism.
- From the Τέχνη perspective, self-guided or peer-guided construction of knowledge promotes (at best) VERY inefficient learning.
- Everyday Math proponents may consider Τέχνη "bogo" constructivism because the importance we place upon guidance interferes with the "construction" process.

The Τέχνη approach

- Problem based instruction with problems taken from the visual domain
- Other domains certainly possible – aural proposed in Τέχνη I
- The focus is upon understanding and computational modeling of physical systems
- NOT “TEACHING GRAPHICS”... no graphics API’s or libraries are used in any Τέχνη course
- Semester long open-ended projects that motivate the use language elements and programming techniques.
- The architecture of the solution is provided by the instructor.
- The students construct the details of the solution.
- This approach is consistent with the cognitive apprenticeship as described by J. R. Anderson, Bandura, Vygotsky and others.

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 and computational thinking

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 - Problem domains have included surface reconstruction, photon mapping, and advanced raytracing.
- CSn - Database management systems – The MeTube projects
- non-CS Introductory programming for non CS majors

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

- The C Language is used in CS-1 because it can be directly mapped to the von Neumann architecture
- Introductory problems in counting, searching, accumulating, and recurrences
- Writing and reading of ppm image files
- Basic parsing of small regular languages
- Basic image manipulations including: resizing, tiling, conversion to grayscale, gamma correction and monochrome conversion
- More difficult image manipulations including: procedural blending, convolution filters, compositing and transfer of color scheme from one image to another.

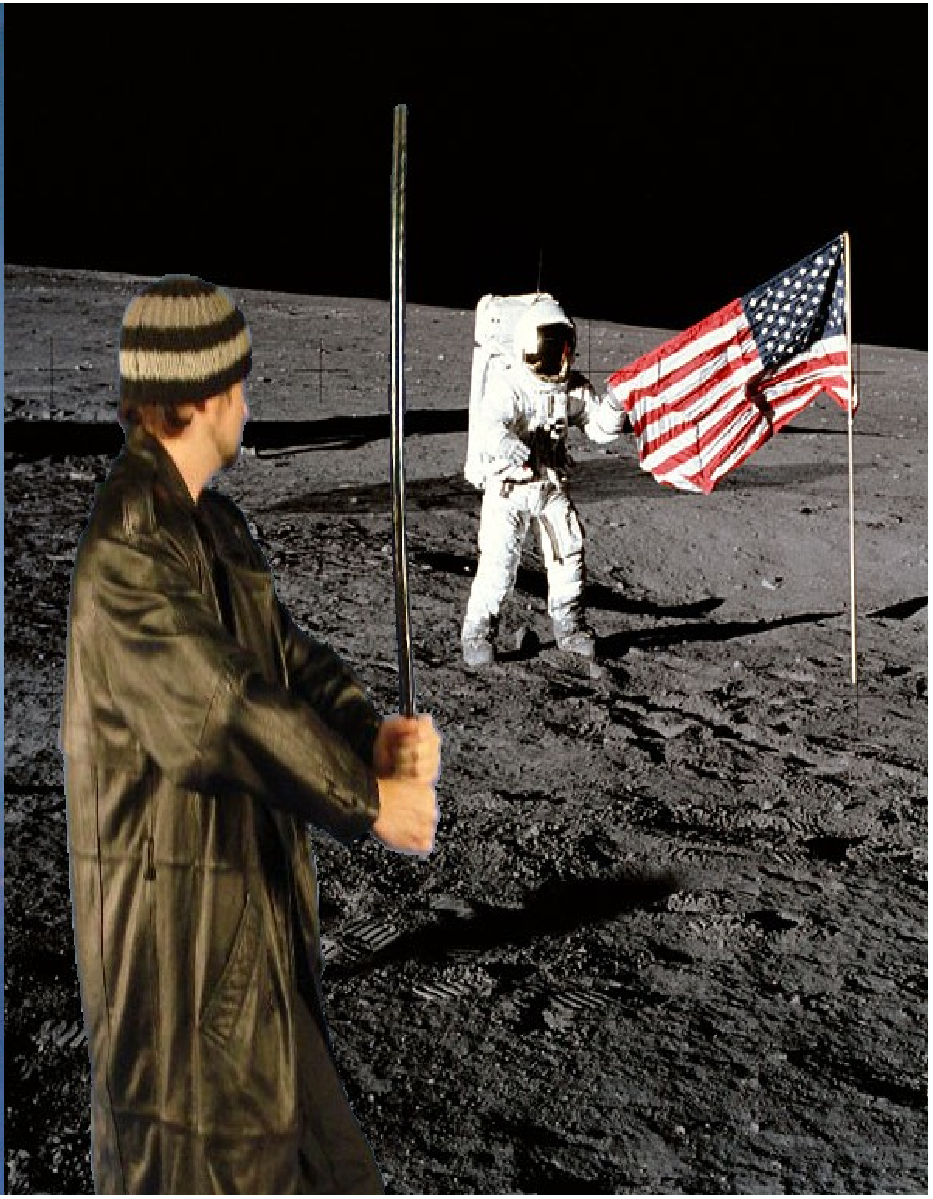
CS 1 image manipulation language

```
sobel
{
   in0 in0.ppm
   out tmunson.sobel.ppm
}

bright
{
   in0 tmunson.sobel.ppm
   out tmunson.sobel3.ppm
   factor 1.2
}
```

CS 1 Sobel filter - Thomas Munson

CS 1 Compositing - Michael Sautter



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 is 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.

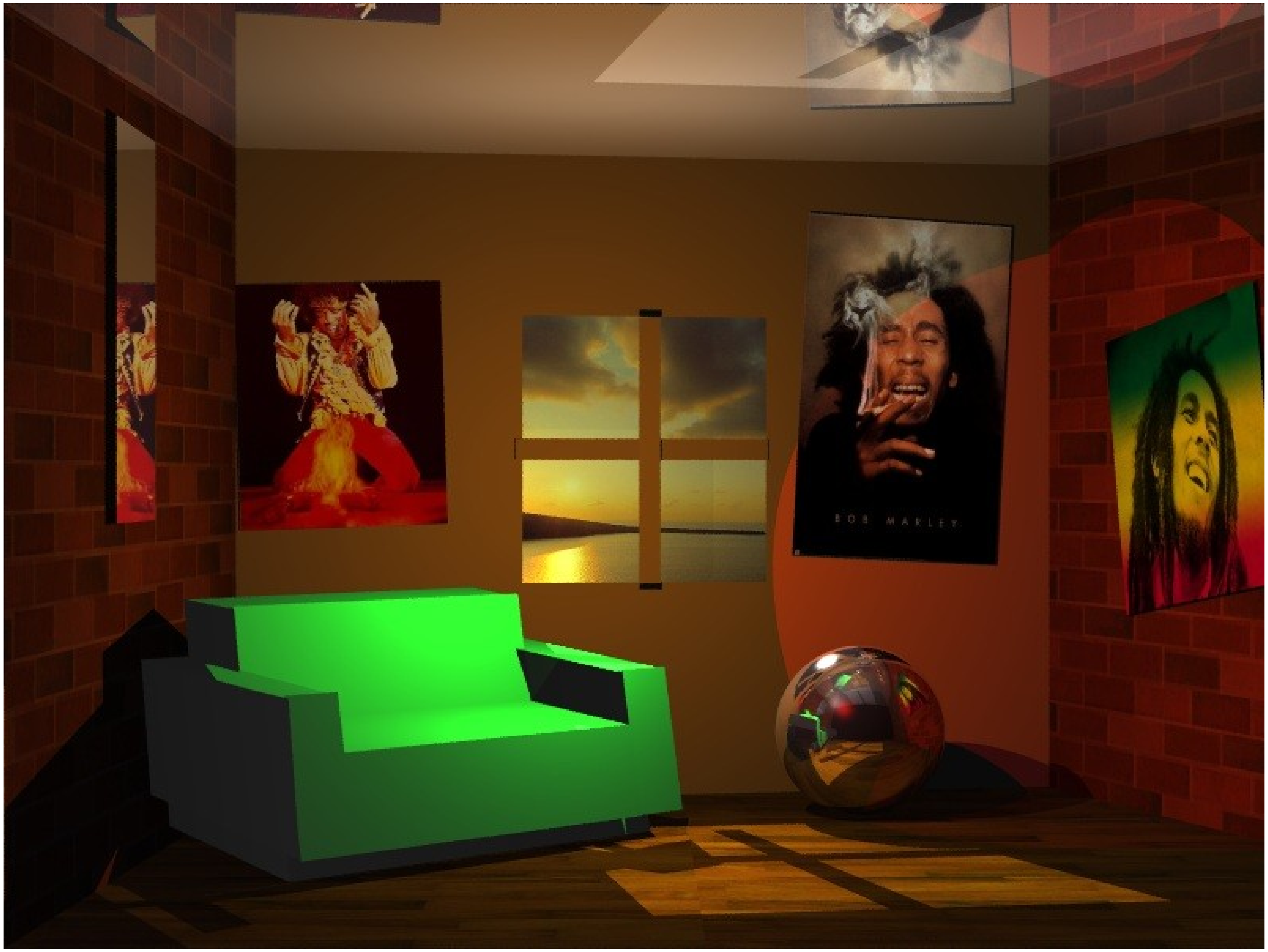
CS 2 problem domain

- 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.

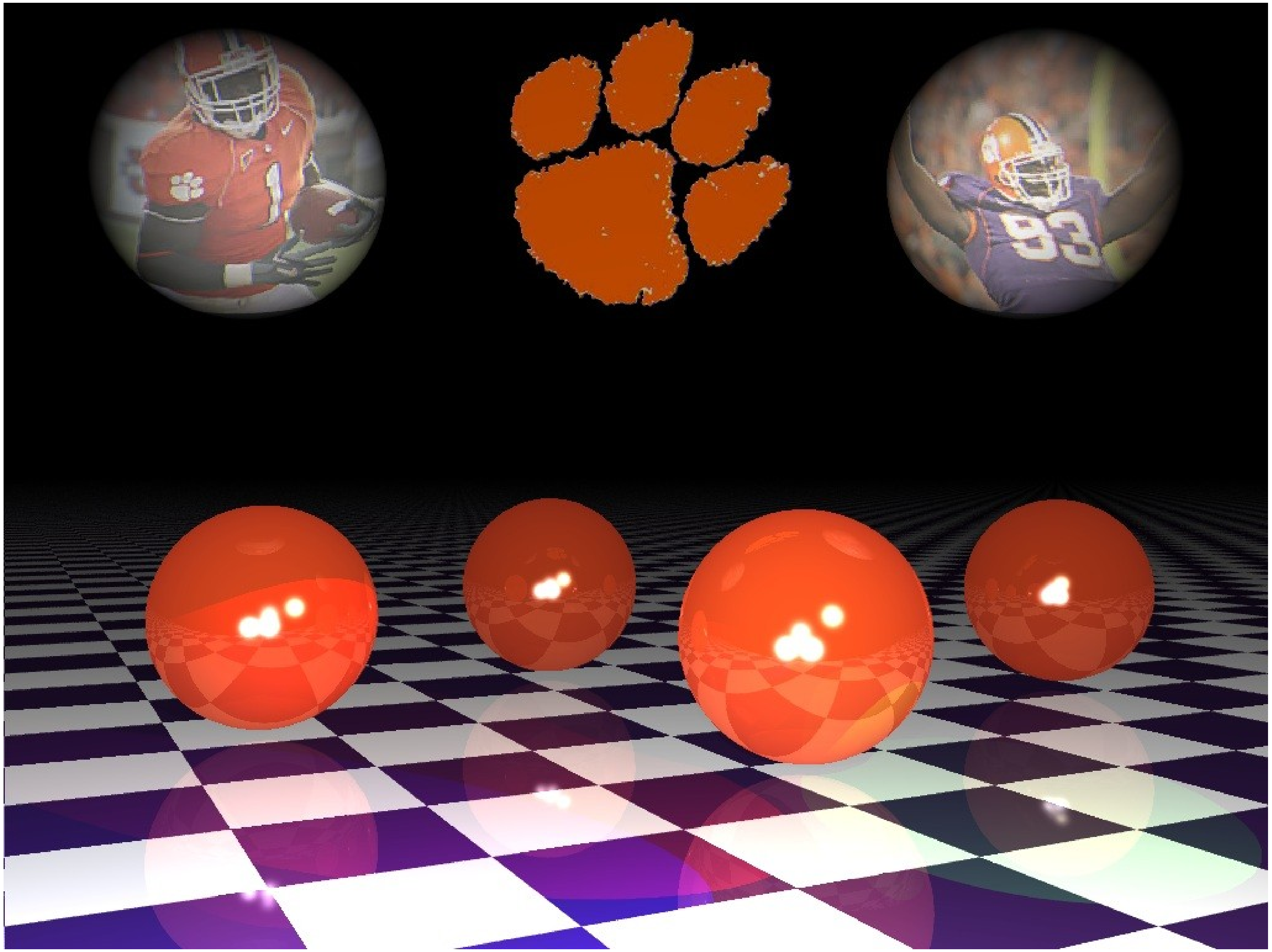
CS 2 example

```
camera cam1
{
  pixeldim  800 600
  worlddim  8 6
  viewpoint 4 3 16
}
material green
{
  diffuse 1 9 1
  ambient 2 2 2
  shininess 100
}
light sun
{
   location 1 12  -18
   emissivity 100 100 100
}
spotlight red
{
  location 2 6 1
```

CS 2 - Textured planes - John Laplante



CS 2 - Projected textures - Tram Bethea



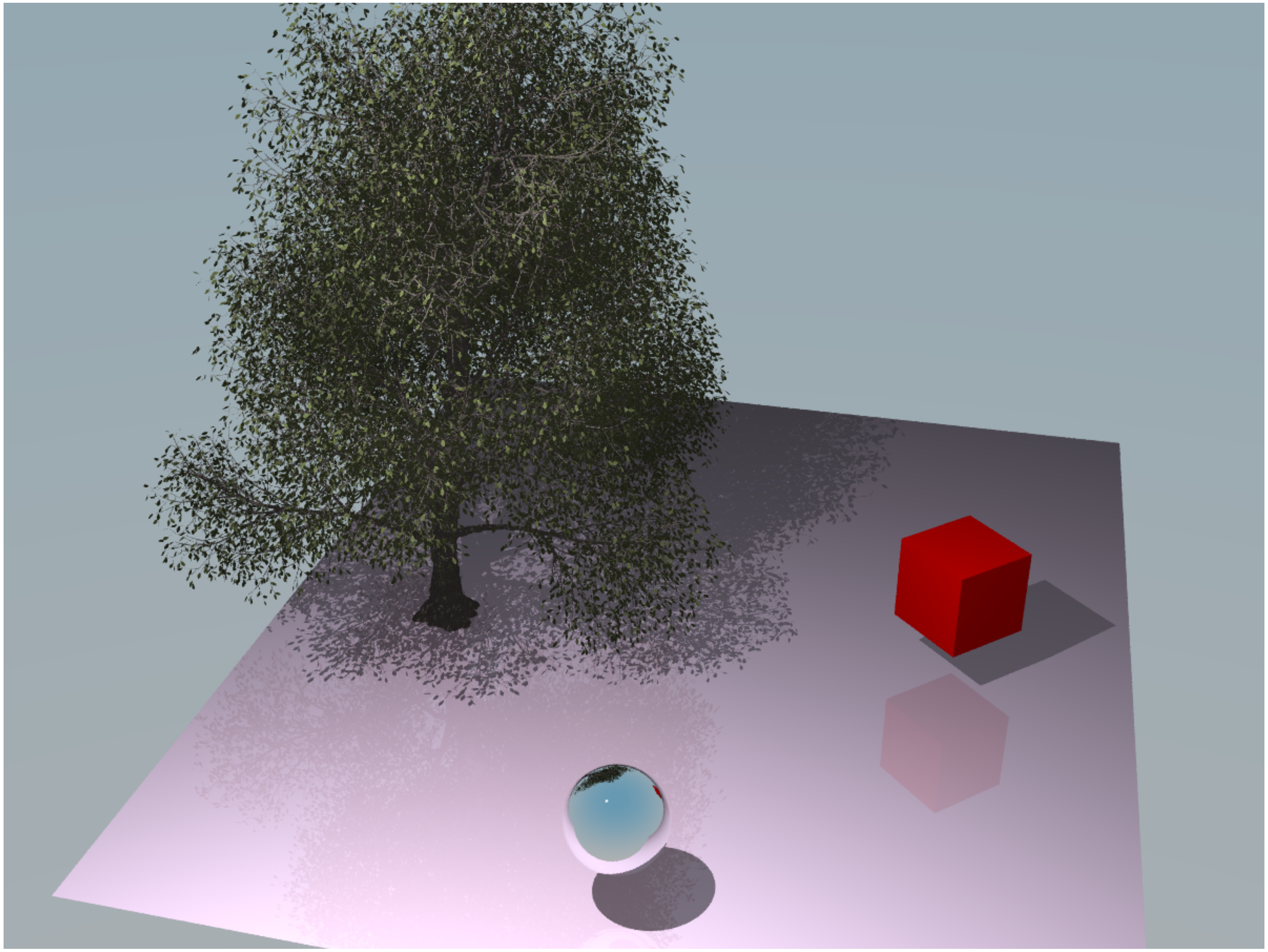
CS3 - Data structures

- Solve problems that are intractible with naive structures
- More depth (but less breadth) than traditional courses

Problem domains have included

- Surface reconstruction from point clouds
- Photon mapping
- Raytracing of scenes with millions of objects

CS 3 - Raytracing a "real" tree - Greg Stabler



Database management systems

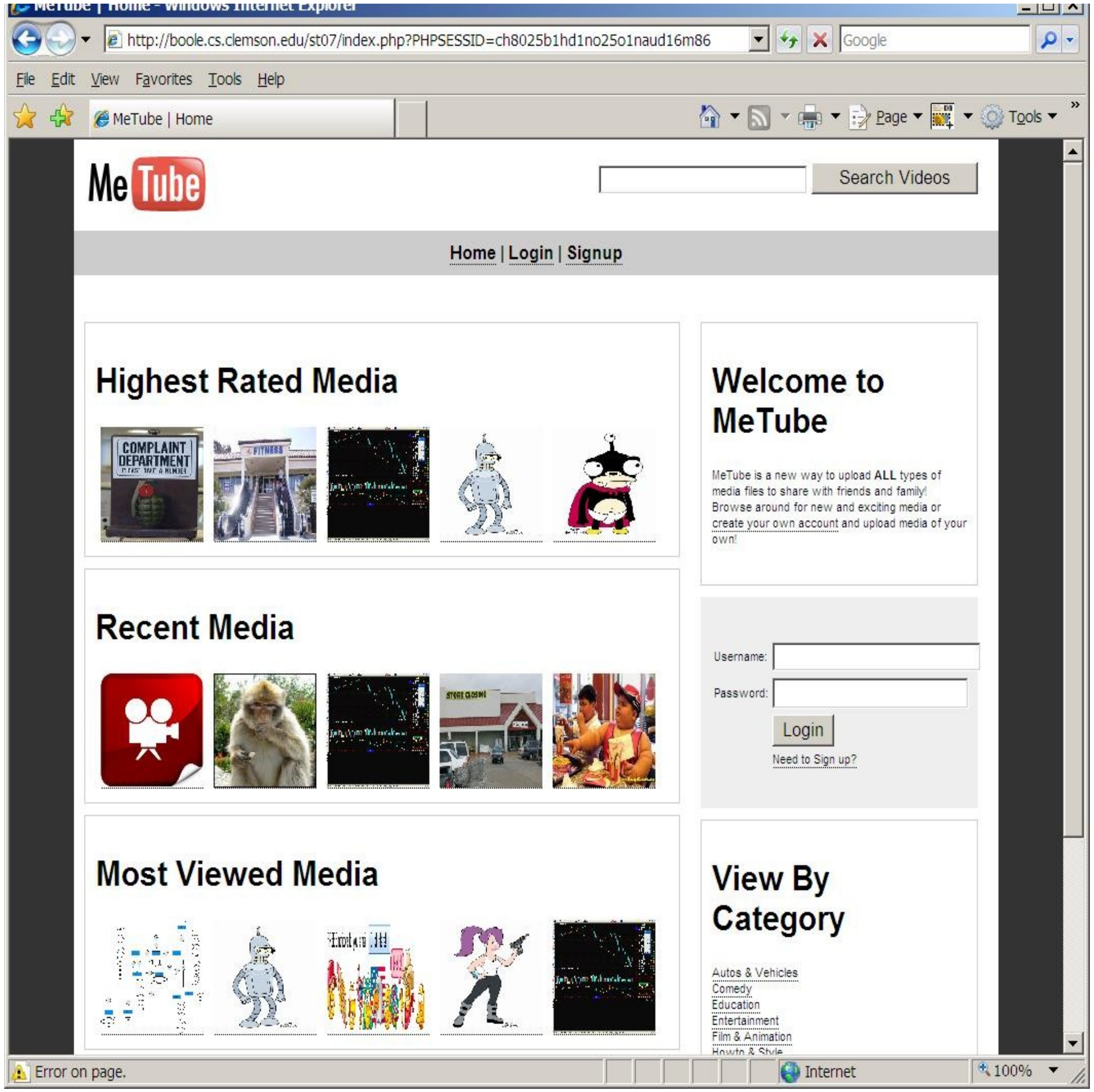
Development of a multimedia database system modeled on the popular YouTube system built on:

- Apache web server
- PHP for server side scripting
- MySQL relational DBMS

Project milestones include

- Entity-Relationship Diagram (5th week)
- Schema and SQL statements for creating the DBMS and populating tables
- Functional requirements and SQL queries
- Working website with PHP scripts
- Live demonstration

MeTube screen shot



http://boole.cs.clemson.edu/st07/index.php?PHPSESSID=ch8025b1hd1no25o1naud16m86

MeTube is a new way to upload **ALL** types of media files to share with friends and family! Browse around for new and exciting media or create your own account and upload media of your own!

Username:
Password:

Personal perspectives

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

I strongly believe that the Τέχνη approach:

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

I tend to believe that the Τέχνη project

- has helped motivate and retain capable students
- has helped students who should have chosen another major do so after only one semester!

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

Acknowledgement

- The first phase of the Τέχνη project was supported by the National Science Foundation under grant TEXNH: A New Approach to the B.A. Degree in Computer Science, award number 0305318.
- The project is presently supported by the National Science Foundation under grant CPATH EAE: TEXNH - Evaluation, Adoption and Extension, award number 0722313.
- The investigators gratefully acknowledge this support without which the project not have been possible.