

Curriculum design for 'Explorations in Computing'

(a new GE course at USC)

Dr. Saty Raghavachary

saty@usc.edu

A new GE course..

- introduced in fall 2016, FROM CS :)
- can't be intro' to programming in disguise - this is for non-CS majors
- students grow up in a digital, media-saturated world, so the material needs to appeal to them
- course content needs to be RELEVANT
- course content needs to be ATTENTION-GRABBING

Relevant

The course topics were grouped into the following categories:

- Media Computing
- Recreational Math
- Algorithmic Art
- Social Media
- Data

Relevant computational principles were presented for each topic.

Engaging

One way to make the topics attractive:


1. do computations/trace algorithms/do design 'by hand'
2. INCORPORATE (TYPE IN) THE RESULTS INTO A TEXT-BASED FILE
FORMAT
3. use publicly available programs to view (or hear, or 3D-print..) the
file containing our data!

The approach above was a big hit!!

What was engaging - two examples


A Spirolateral curve, drawn using a turtle interface:

Question, 2 points: complete the 4₆₀ spirolateral.



Command (type code below)

Canvas (graphical output of our commands)

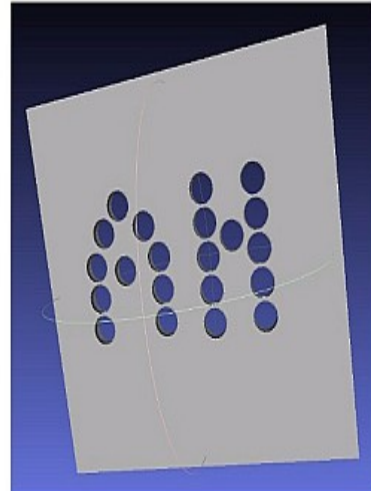


Definitions (type code below)

```
function spiro20(  
  forward(10);  
  right(120);  
  forward(20);  
  right(120);  
  forward(30);  
  right(120);  
  forward(40);  
  right(120);  
}
```

A student's initials, 'drilled' out in the ShapeJS interface:

```
function main(args) {  
  var box = new Box(20*MM,20*MM,1*MM); // our 'base plate' to cut holes into  
  
  var radius = 1*MM; // we will create 1mm holes  
  var sphere1 = new Sphere(5*MM,5*MM,0*MM, radius); // hole1, at x,y = 5,5  
  var sphere2 = new Sphere(-5*MM,5*MM,0*MM, radius); // hole2, at x,y = -5,5  
  // todos: change sphere1's and sphere2's x,y locations above; ADD MORE spheres  
  
  var cutout = new Union(); // overall negative volume generated from holes  
  cutout.add(sphere1); // sphere1 is the only negative volume at this point  
  cutout.add(sphere2); // our negative volume includes sphere2 as well..  
  //todo: ADD sphere3, sphere4 etc., similar to the above two lines  
  
  var myShape = new Subtraction(box, cutout); // subtract (cutout) holes from plate  
  
  var s = 12*MM; // overall bounds will be -12 to 12, ie 24 (MM)  
  return new Scene(myShape, new Bounds(-s,s,-s,s)); // generate printable  
}
```



Useful (ASCII) file formats

Following are text-oriented formats and software environments that can serve as containers for plain data, to help transform them into a variety of output:

- .svg: vector graphics (lines, shapes, fills..)
- .obj: polygonal meshes
- .stl: 3D-printable shapes
- .ppm: images
- .kml: points, lines, polygons over a map
- d3.js: code to create graphs (bar charts, bubble diagrams..)
- ShapeJS: code to output 3D-printables
- turtle graphics: code to output colored lines
-