

## Lab 3: PGraphics

Thursday, February 09, 2017

In this lab, we will be using Processing for intersection detection via PGraphics. Although simple Math is more superior than this technique, this lab enables us to explore new Processing features, and learn a few new concepts. We will be learning the basics of PGraphics, 3D canvas, and rotation and translation of objects.

### Basic requirements for this lab:

1. Implement in-class example:

```
PGraphics backbuffer;
void setup(){
  size(1000, 1000, P3D);
  backbuffer = createGraphics(width, height, P3D);
}

void draw(){
  background(102);

  //draw rect
  pushMatrix();
  translate(width*0.2, height*0.2, 0);
  rotateY(frameCount/ 15.0);
  rect(0, 0, 100, 100);
  popMatrix();

  color c1 = color(250, 0, 0);
  backbuffer.beginDraw();
  backbuffer.background(0);
  backbuffer.translate(width*0.2, height*0.2, 0);
  backbuffer.rotateY(frameCount / 15.0);
  backbuffer.fill(c1);
  backbuffer.rect(0, 0, 100, 100);
  backbuffer.endDraw();

  //image(backbuffer, 0, 400);
  color c2 = backbuffer.get(mouseX, mouseY);
  if (c1 == c2){
    println( "selected" );
  }else{
    println( "nope" );
  }
  println(red(c2));
}
```

2. Intersection detection of a moving target
  - a. Draw a shape of your choosing (e.g. sphere, box, star)
  - b. Your shape should be moving (e.g. spinning, bouncing, etc.) on your 3D canvas
  - c. Duplicate your moving target on a secondary canvas 3D canvas
  - d. Track the user's mouse clicks by sampling pixel colors from the secondary canvas
  - e. Should the user click on the shape, the shape should move a random location on the canvas.

**Submission:**

Submit your work via **Blackboard** at the end of class or by midnight. Use the naming convention: "FirstnameLastname\_l3.zip" (e.g., AlvittaOttley\_l3.zip). The labs are not graded but submission counts toward your participation grade. Add the name of your partner to the comments as the beginning of your .pde file.