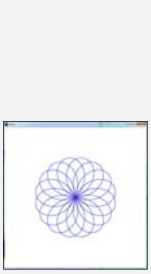
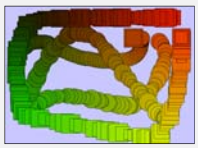
Three Dimensions





Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu

The 3D Processing Functions You Will Care About

```
rotateX( radians );
rotateY( radians );
rotateZ( radians );

scale( sx, sy, sz );

translate( tx, ty, tz );

box( sizex, sizey, sizez );

sphereDetail( slices, stacks );
sphere( radius );

beginShape( );
vertex( x, y, z );
endShape( );
```


Sample 3D Program – the Global Variables at the top of the program

```
int LastMouseX;
Int LastMouseY;
int Udetail = 20;
Int Vdetail = 20;
float Yangle = 0.;
float Xrot = 0., Yrot = 0.;

boolean FillSphere = false;
boolean StillPressed = false;
boolean Animate = false;
```


Sample 3D Program – the setup( ) Function

```
void
setup( )
{
  size( 800, 800, P3D );
  background( 200, 200, 255 );
  stroke( 0, 0, 0 );
  fill( 255, 255, 0 );
}
```

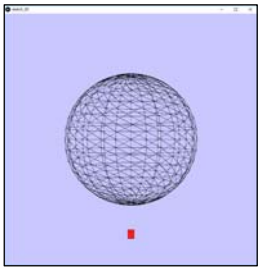
This third argument to size( ) tells Processing to allow it to do 3D

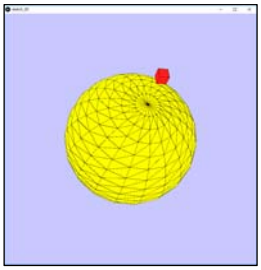
Sample 3D Program – the draw( ) Function, part I

```
void
draw( )
{
  background( 200, 200, 255 );
  if( keyPressed )
  {
    if( ! StillPressed )    // same as saying "if( StillPressed == false )"
    {
      switch( key )
      {
        case 'a':
          Animate = ! Animate;
          break;

        case 'f':
          FillSphere = ! FillSphere;
          break;

        case 'l':
          Udetail = Udetail - 1;
          Vdetail = Vdetail - 1;
          break;

        case 'm':
          Udetail = Udetail + 1;
          Vdetail = Vdetail + 1;
          break;
      }
    }
    StillPressed = true;
  }
  else
  {
    StillPressed = false;
  }
```


Sample 3D Program – the draw( ) Function, part II

```
  if( mousePressed )
  {
    int dx = mouseX - LastMouseX;
    int dy = mouseY - LastMouseY;
    Xrot = Xrot + dy;
    Yrot = Yrot + dx;
  }
  LastMouseX = mouseX;
  LastMouseY = mouseY;

  translate( width/2, height/2 );
  rotateY(  radians(Yrot) );
  rotateX(  radians(Xrot) );

  fill( 255, 30, 30 );
  pushMatrix( );
    rotateX(  radians(Yangle) );
    translate( 0, 300, 0 );
    box(20, 20, 20);
  popMatrix( );

  if( FillSphere )
    fill( 255, 255, 0 );
  else
    noFill( );

  pushMatrix( );
    rotateY( radians(Yangle) );
    sphereDetail( Udetail, Vdetail );
    sphere( 200. );
    if( Animate )
      Yangle = Yangle + 1.;
  popMatrix( );
}
```





Oregon State
University
Computer Graphics