

Lighting

How much can you put on the screen in 3D before Processing starts to slow down?

```
1 void setup(){
2   size(800, 800, P3D);
3 }
4
5 float T = 0.01;
6
7 void keyPressed(){
8   beginCamera();
9   if(keyCode == LEFT){
10    rotateY(T);
11  } else if (keyCode == RIGHT){
12    rotateY(-T);
13  } else if (keyCode == UP){
14    rotateX(T);
15  } else if (keyCode == DOWN){
16    rotateX(-T);
17  }
18  endCamera();
19 }
20
21 void draw(){
22   noStroke();
23   background(50);
24   for(int i = 0; i < 800; i+=15){
25     for(int j = 0; j < 800; j+=15){
26       pushMatrix();
27       translate(i, j, 200);
28       box(5);
29       popMatrix();
30     }
31   }
```

Draw 2800 boxes to the screen.

Each box is 12 triangles.

Total: over 33,600 triangles.

Result: reasonable framerate.

Is 3D slower than 2D?

Yes, but not actually by that much.

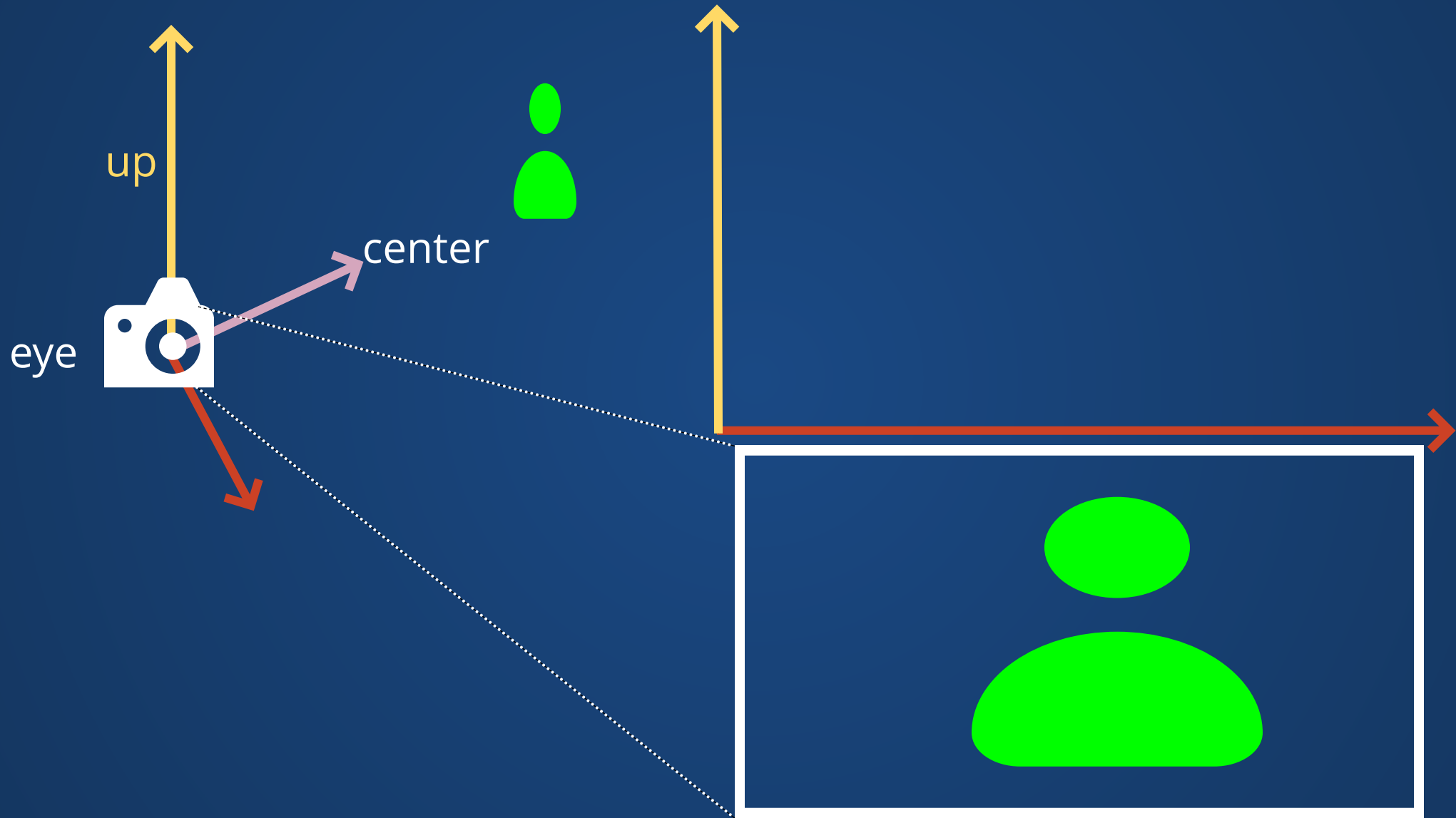
Can you make multiple cameras and switch between them?

Not built-in to Processing, but this isn't that hard to do yourself!

What would we put in a class that represents a (fixed) camera position that we want to switch to?

```
1 class PresetCamera {
2   PresetCamera(float eyeX, float eyeY, float eyeZ,
3               float atX, float atY, float atZ,
4               float upX, float upY, float upZ){
5     // Stuff stuff stuff
6   }
7   void activate(){
8     camera(this.eyeX, this.eyeY, ...)
9   }
10 }
```

What does the `up` parameter do in the camera?



What happened to that grid drawing code?

Sorry! It's up on Canvas now, along with yesterday's demo.

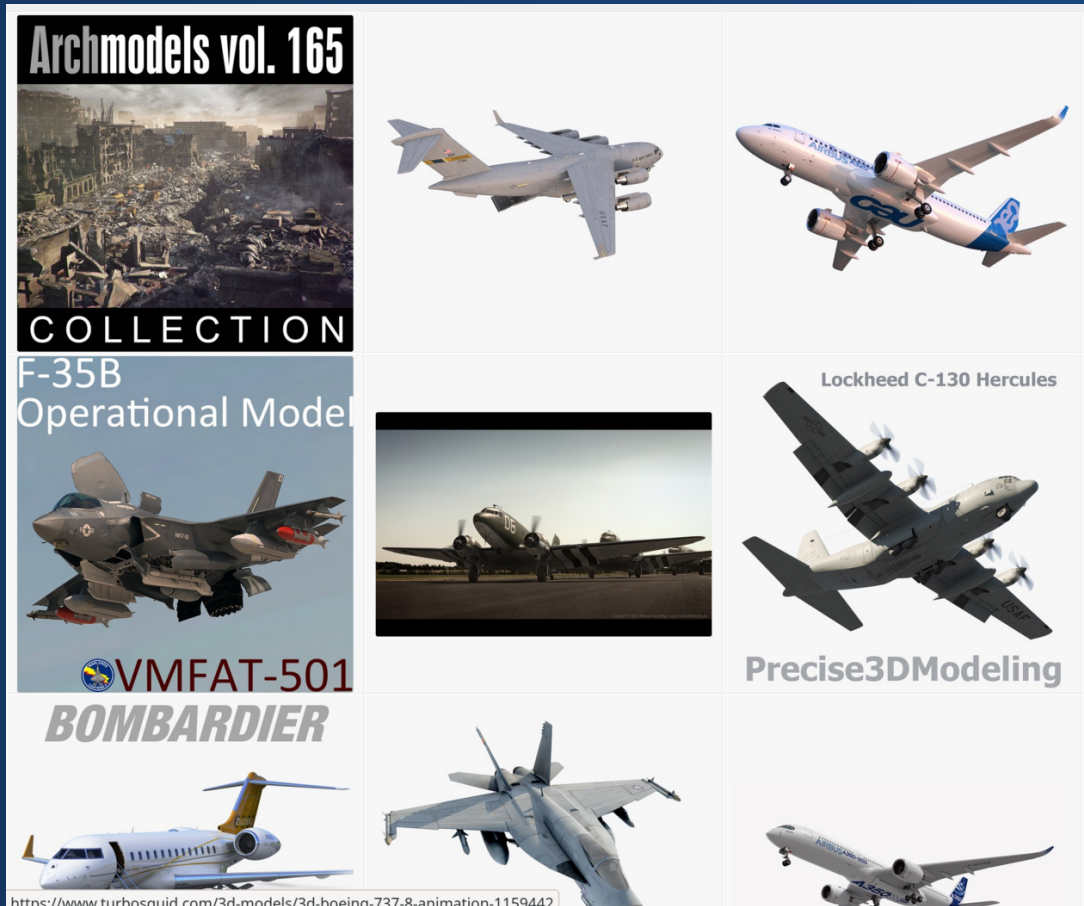
How can we visualize the camera position??

No real great ways to do this, unfortunately :\

Take your scene, draw it out on graph paper, and then try to plot where the camera goes. (But this is labor intensive!)

Where can we get custom 3D models?

Lots of options, just be aware that 3D file formats are not standardized (i.e. some files that work in some apps may not work in others).



For OBJ files (can load directly into Processing):

- TurboSquid
- Clara.io

For STL files (require conversion):
Thingiverse

How can we animate 3D objects?

In 2D

```
1 void setup(){
2   size(800,800);
3 }
4
5 void draw(){
6   background(50);
7   float ctrl = (frameCount % 100) / 100.0;
8   float x = lerp(0, 800, ctrl);
9   float y = lerp(0, 800, ctrl);
10  translate(x, y);
11  rect(0, 0, 50, 50);
12 }
```

In 3D

```
1 void setup(){
2   size(800,800, P3D);
3 }
4
5 void draw(){
6   background(50);
7   float ctrl = (frameCount % 100) / 100.0;
8   float x = lerp(0, 800, ctrl);
9   float y = lerp(0, 800, ctrl);
10  translate(x, y, 0);
11  box(50);
12 }
```

Is there ever a time when it's practical to make 3D shapes by explicitly specifying points?

Boxes and triangles.

Are there other built-in 3D shapes in Processing?

Just `box()` and `sphere()`

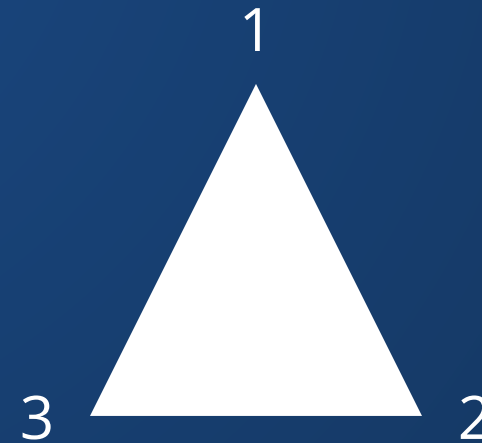
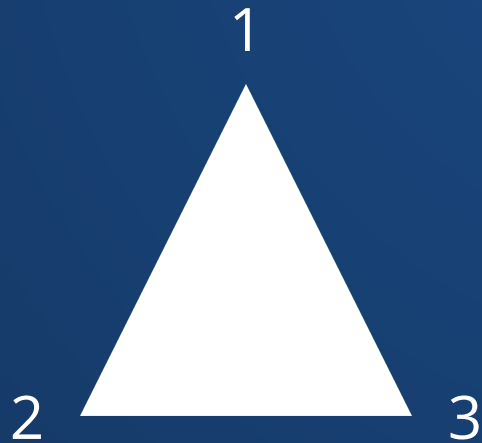
How do people create 3D shapes?

Lots of hard work. Programs like Blender exist, but word of mouth seems to be that programs like AutoDesk Maya and Houdini are much better for experienced users.

What are some recommended practices when working in 3D?

Take time to think about what your shapes are doing and why they're doing that. Drawing on paper is often helpful.

Order matters more in 3D than in 2D. Example: triangle directions.

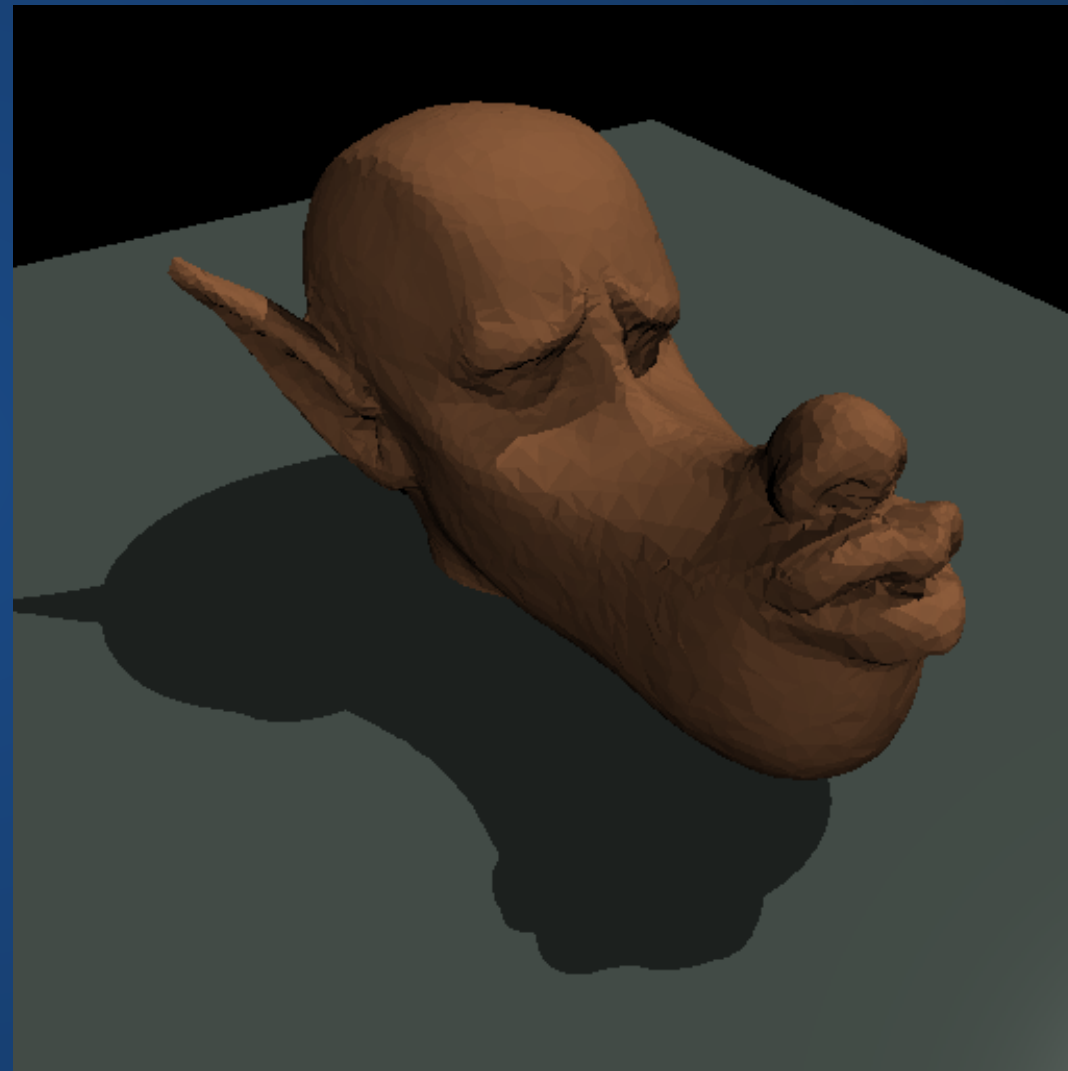
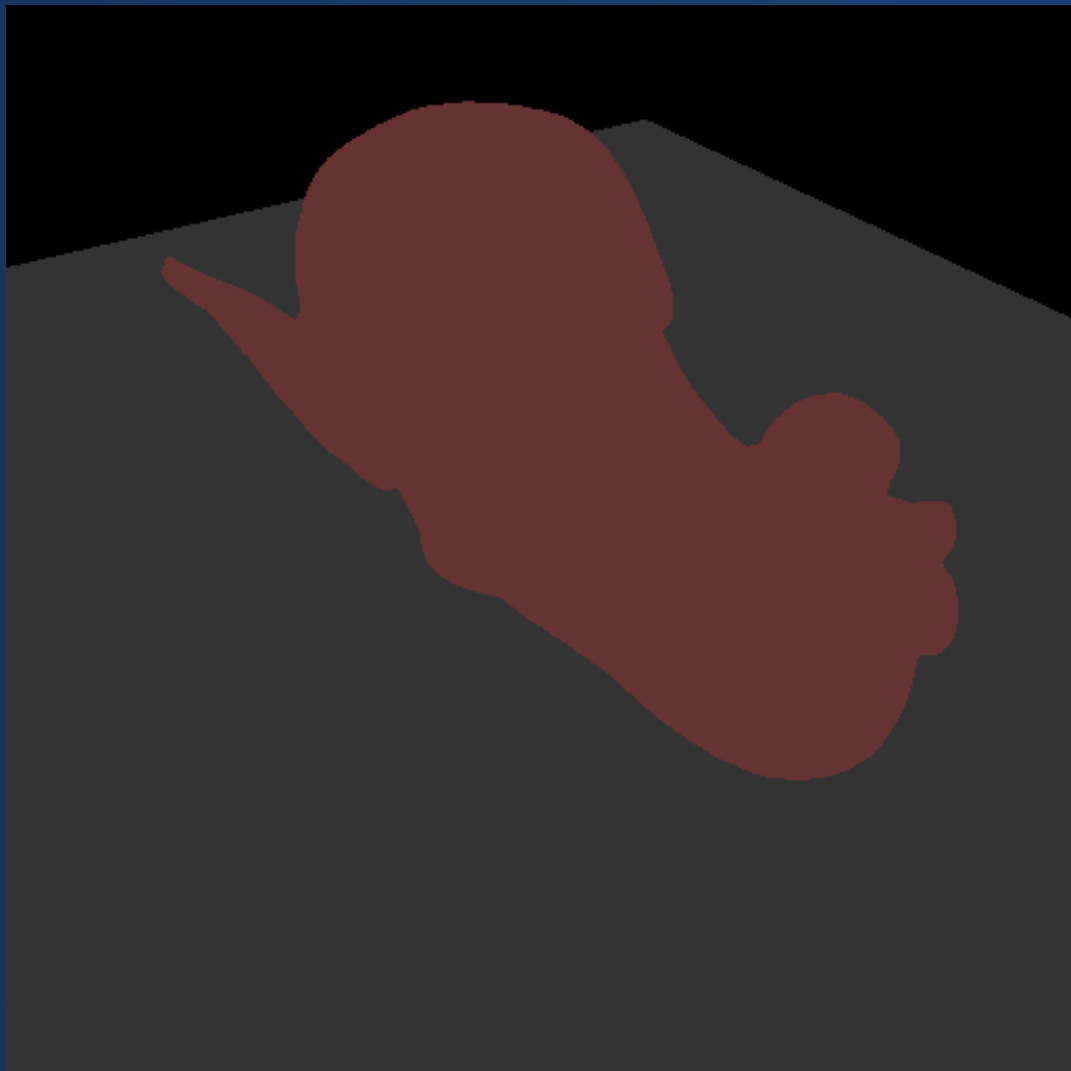


When we were working in 2D, we didn't really need to worry about lighting.

Give each shape a color, give the background a color, and we're done!

We can do this in 3D too, it just doesn't work nearly as well.

Light and shadow matter for being able to recognize objects!



Shading

Approximate the (real) interaction of light with matter.

- What is the object's material?
- How does light interact with this material?
- Where is the camera, and does it matter?



Toy Story (1995)



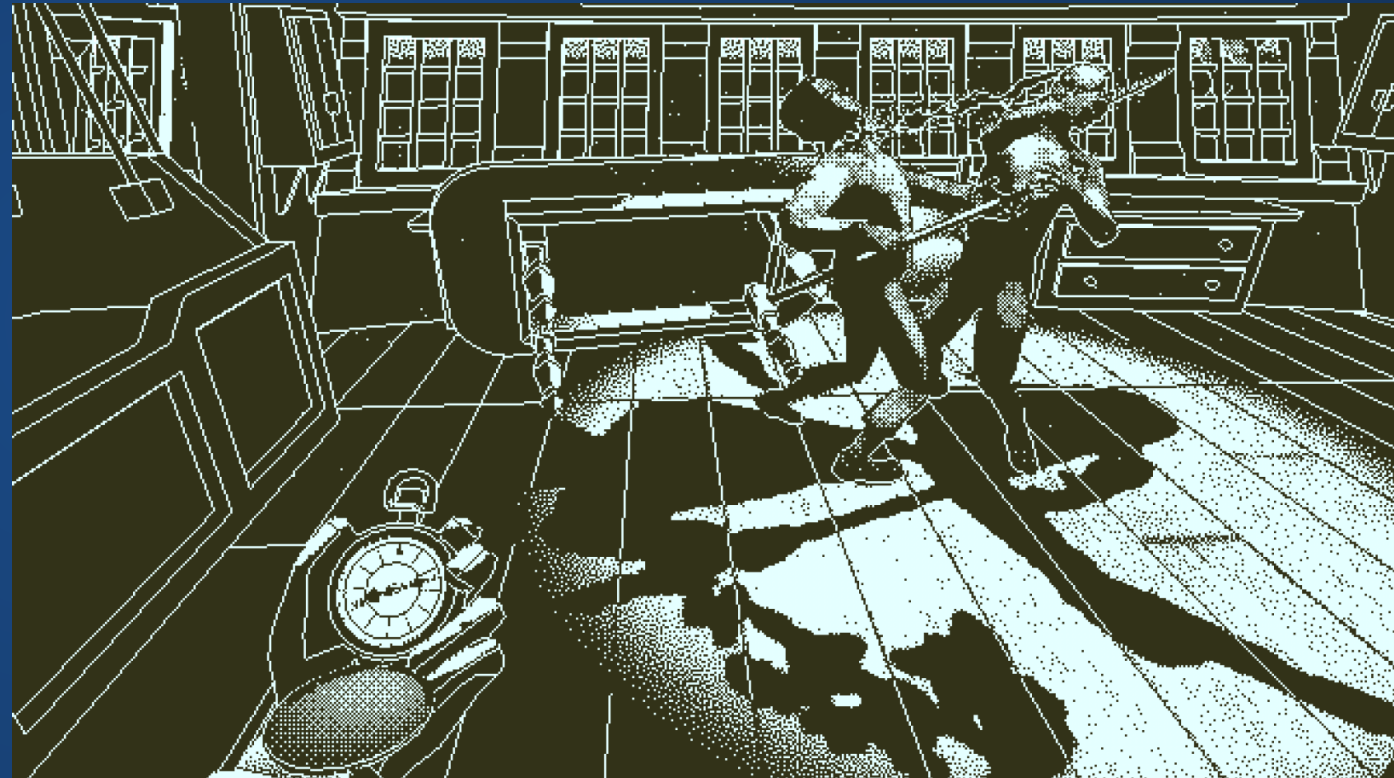
Big Hero 6 (2014)

The answers do not have to be realistic!

Sometimes we'll deliberately make shading unrealistic, for performance or stylistic reasons.

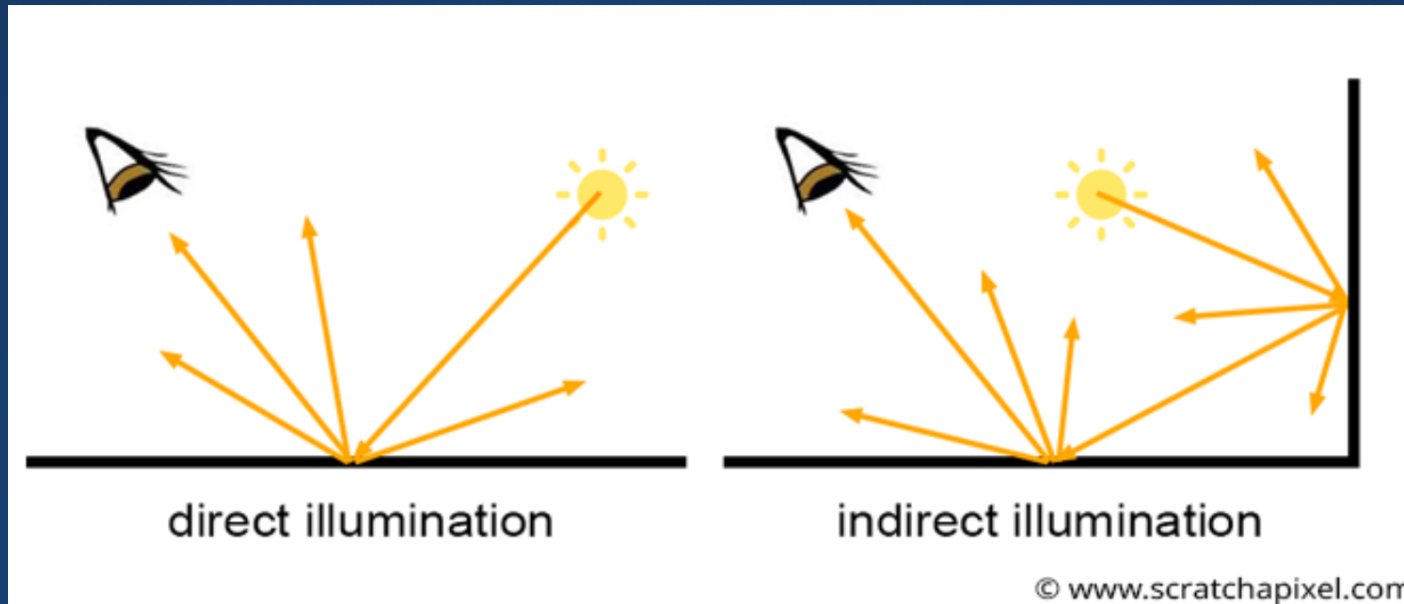


TLoZ: Wind Waker (2002)



Return of the Obra Dinn (2018)

Types of Lighting



We will focus on direct illumination.
(Indirect is challenging even for graduate students!)

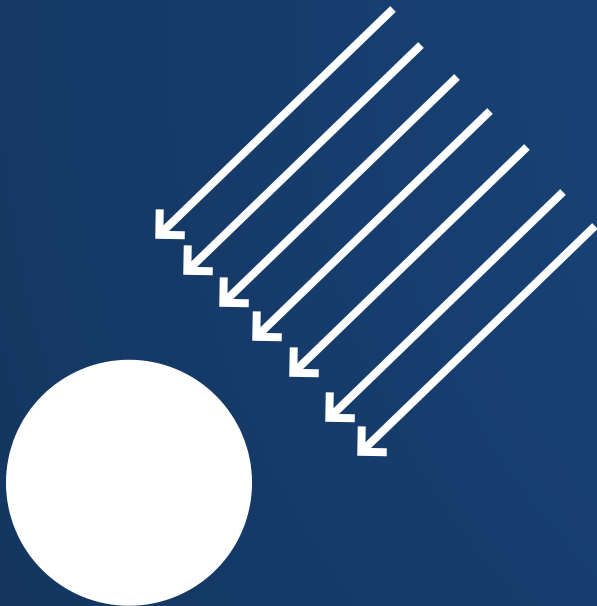
Types of Lights

Intensity and direction of light sources change what surfaces are affected, and how.

What kinds of light sources might we have?

Directional Lights

Light source with no position, but a direction. Can think of this like sunshine: a light located at infinity so that there's no "source".



In Processing: `directionalLight(r, g, b, dir_x, dir_y, dir_z)`

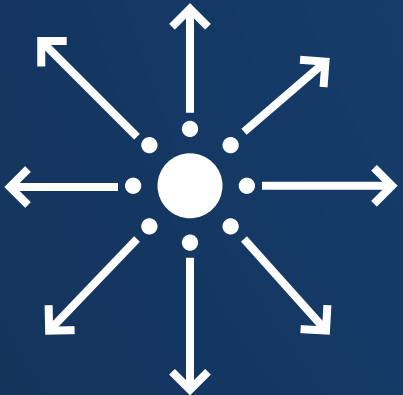
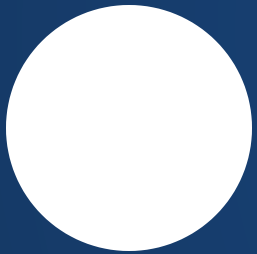
- First three parameters define the color (RGB or HSV depending on the color mode)
- Next three parameters define the direction.

```
1 void setup() {
2   size(500, 500, P3D);
3   noStroke();
4 }
5 void draw() {
6   background(0);
7   //This shines a light in the -y direction (note:
8   // Processing defaults to a left-handed coordinate system)
9   directionalLight(255, 255, 255, 0, -1, 0);
10  translate(width/2, height/2, 0);
11  sphere(50);
12 }
```

Point Light

Light source with no direction, but position. Can think of this like an idealized lightbulb: shines in all directions.

```
pointLight(r, g, b, pos_x, pos_y, pos_z)
```

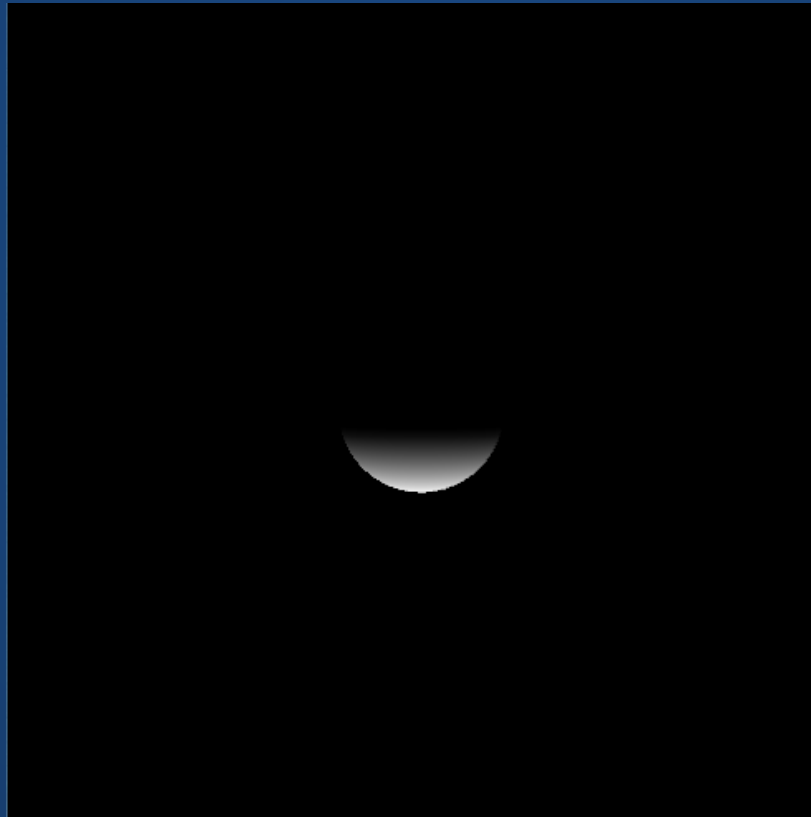


- What do the first three parameters represent?
- What do the second three parameters represent?

How does intensity of light change if we change position relative to it?

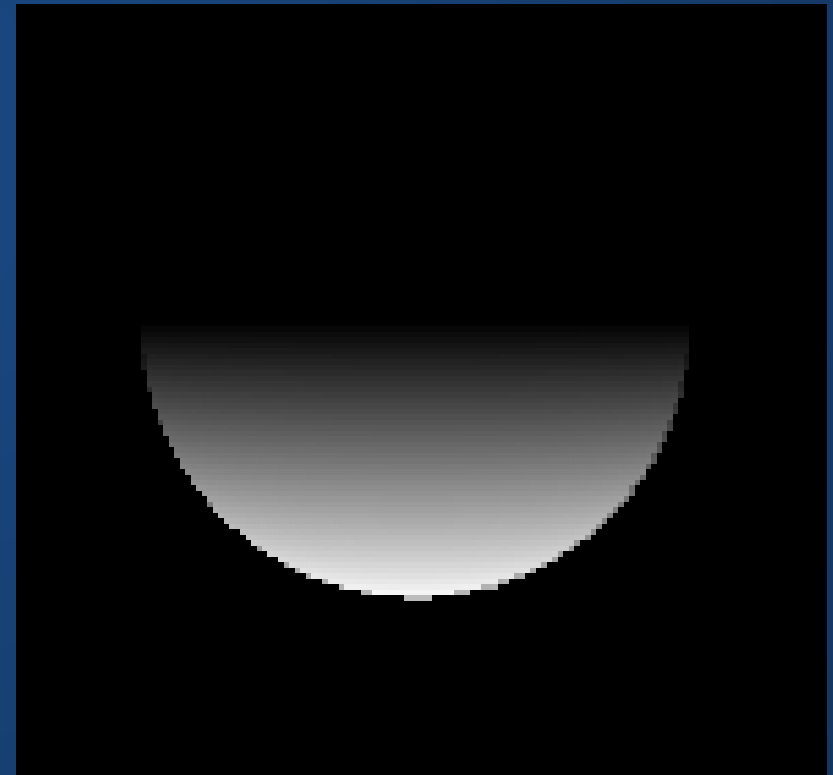
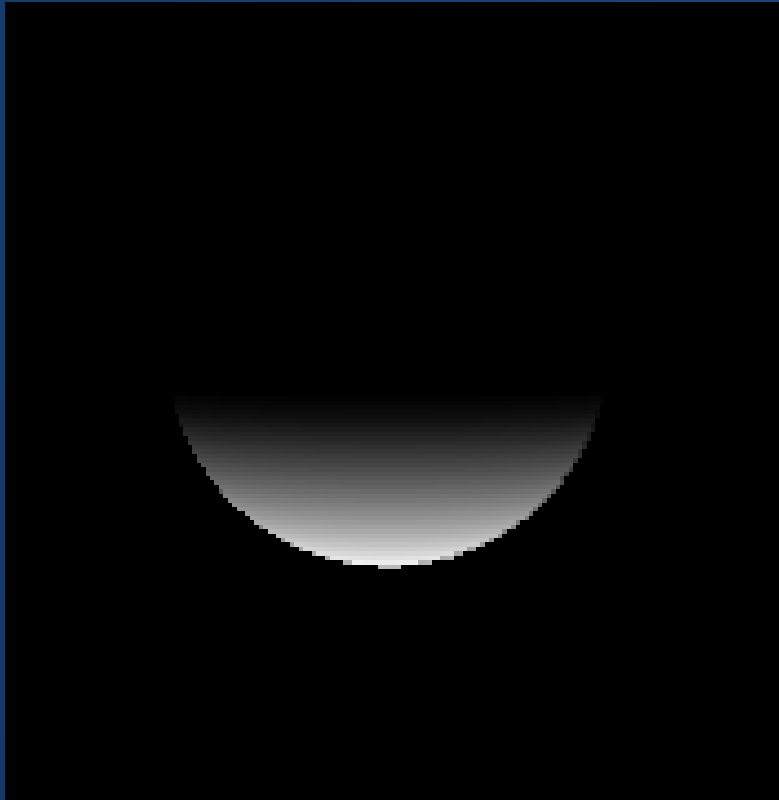
Question

Where is the point light in this image?



Question

Which image has the point light, and which has the directional light?

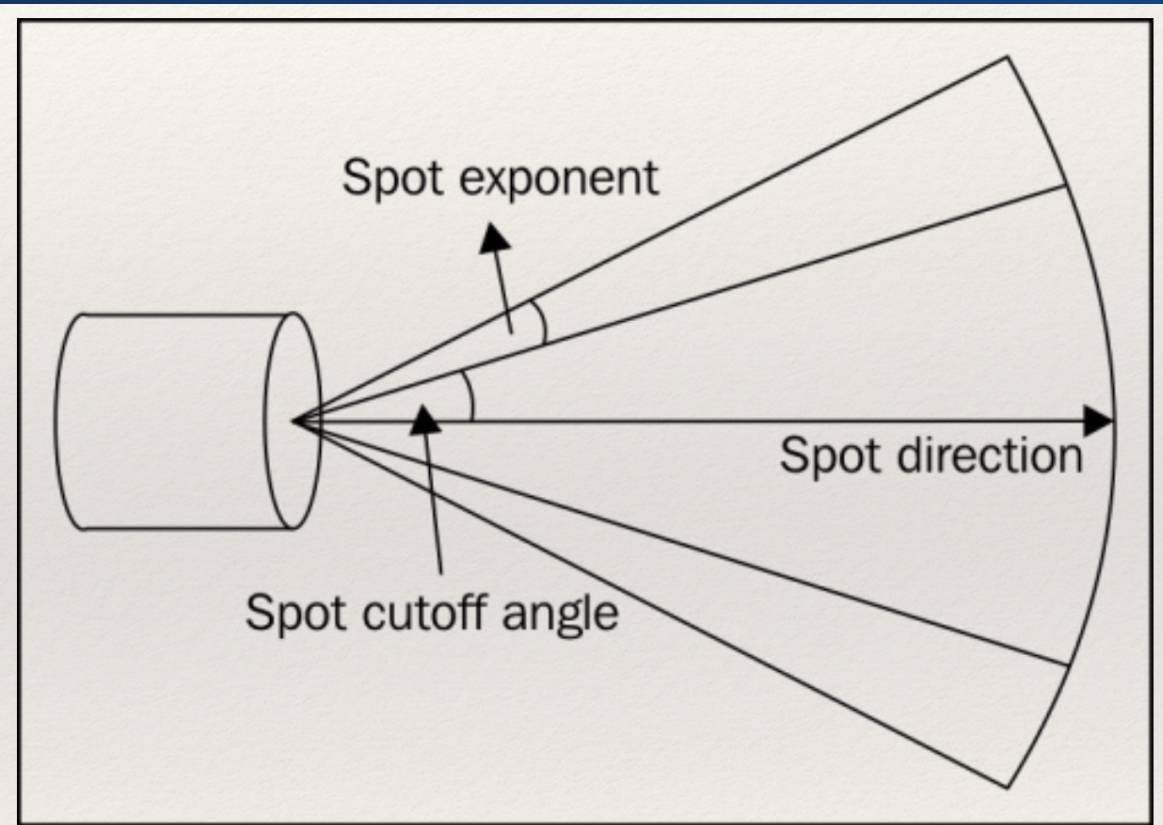


Spot Light

Has position and direction. Approximates a single source of light with direction and fall off. Like a spotlight!



```
spotLight(r, g, b,  
          pos_x, pos_y, pos_z,  
          dir_x, dir_y, dir_z,  
          angle, concentration)
```



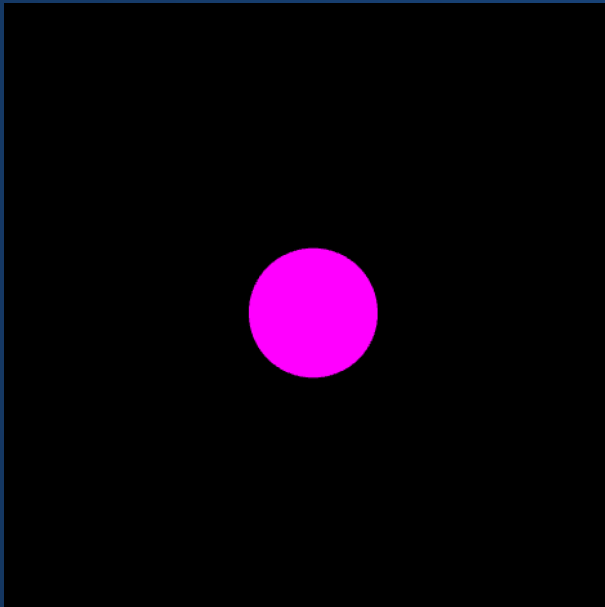
(Safari Books, OpenGL cookbook)

- Angle is the cutoff angle of the light. The smaller this value, the more laser-like the light is.
- Concentration (exponent) is how quickly the light falls off as you move away from the direction.

Ambient Light

A non-directional "general" source of light. Approximates many light bounces within a scene.

Similar to indirect lighting on a cloudy day, or (to a lesser extent) during magic hour.



```
ambientLight(r, g, b)
```


Hands-On: Lighting

1. Create 3D objects in a scene along with a camera
2. Create one of each:
 1. A directional light
 2. A point light
 3. A spot light
 4. An ambient light
3. Modify everything you've created until the scene looks reasonably nice to you.

Index Cards!

1. Your name and EID.
2. One thing that you learned from class today. You are allowed to say "nothing" if you didn't learn anything.
3. One question you have about something covered in class today. You *may not* respond "nothing".
4. (Optional) Any other comments/questions/thoughts about today's class.