

Materials and Textures

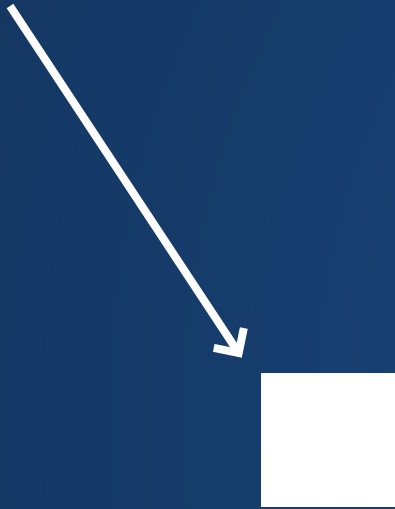
Lights only capture one half of shading.

Shining the same light at a mirror, a piece of paper, and a glow-in-the-dark toy will have very different effects.

How can we capture this idea?

Materials

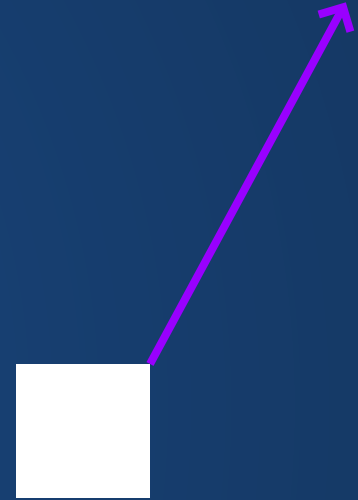
How does light interact with stuff?



Step 1: Light comes in



Step 2: ????



Step 3: Colors!

If only physics were this simple.



Why are we seeing red as opposed to blue or white?

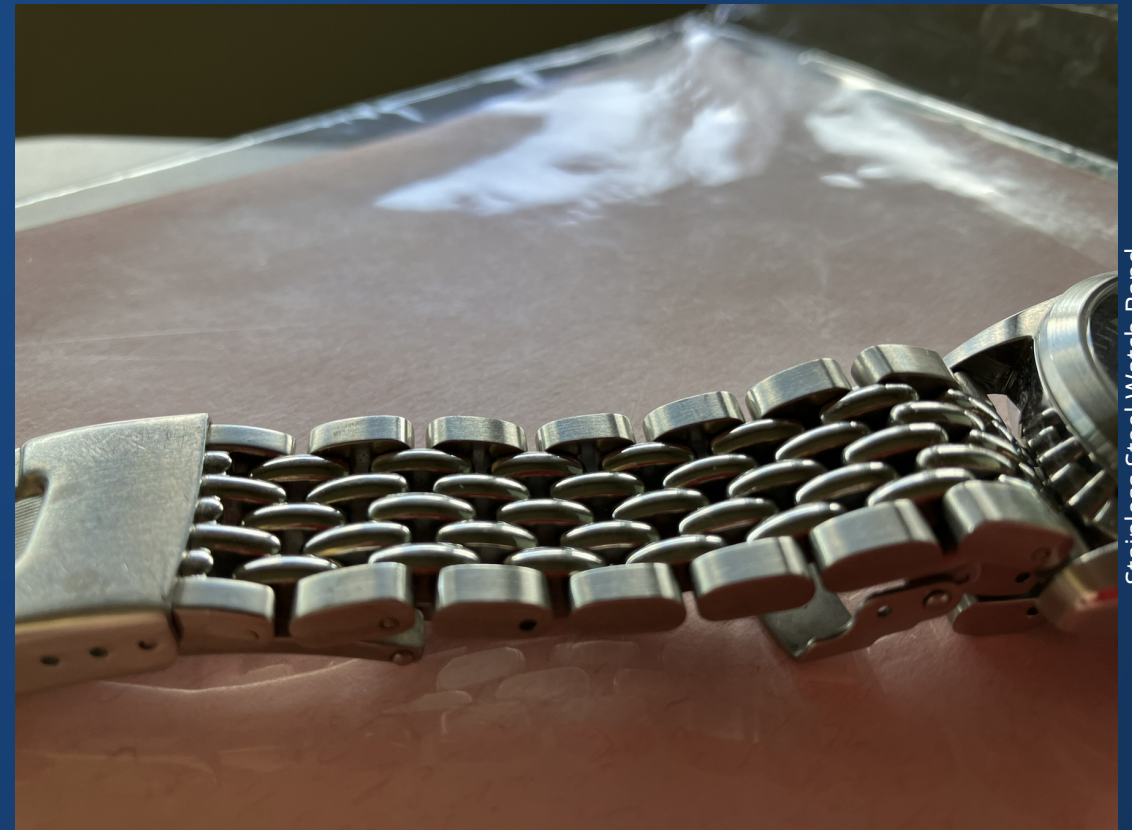
Why can we see the little dents/divots in the surface of the cover? (The material is the same everywhere).

Why does the color change as we get close to the top of the cover?

Why are the top halves of the links bright, but not the bottom half?

Why is the plastic surface bright in some parts and not others?

How does the plastic interact with the light in the non-bright parts?



Material Model

We want some way to capture the interaction of light with matter.

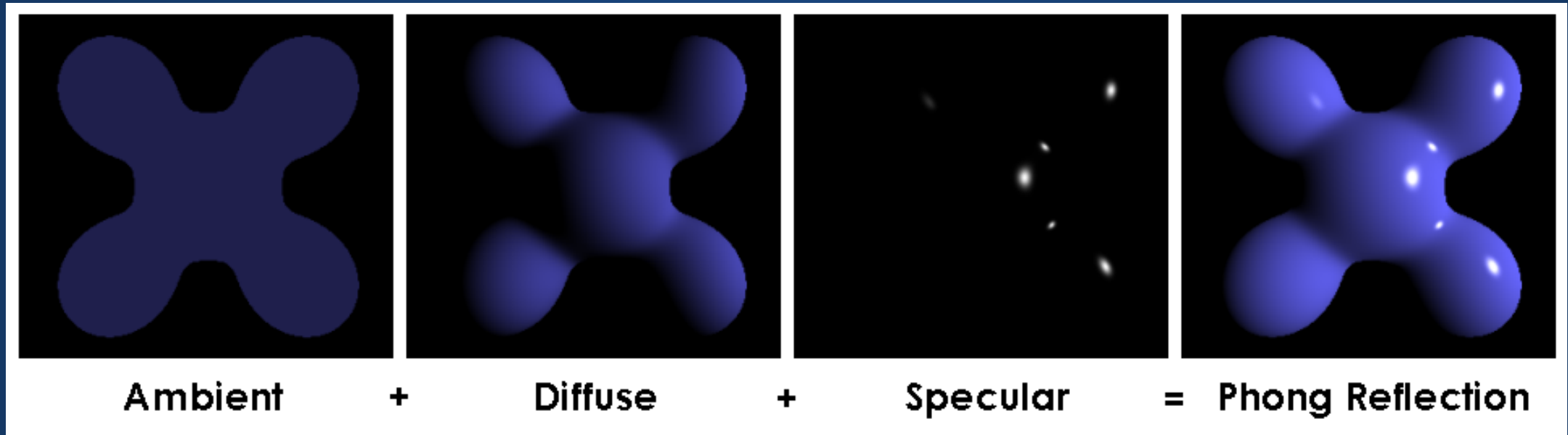


Need to balance different goals!

- Realism
- Speed
- Control

Phong

Non-physically-based material model that roughly captures material properties.



Ambient

Add some small component to the shading.



Light position and camera position are irrelevant, always add color.

Is this physically realistic? Of course not!

But can be used to approximate e.g. light that bounces off of other surfaces, or other light that comes from outside the scene.

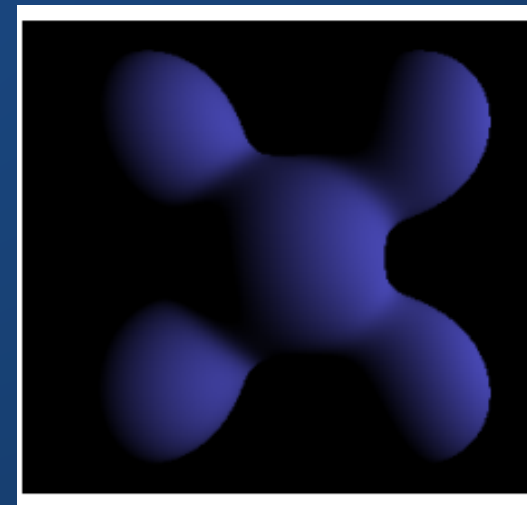
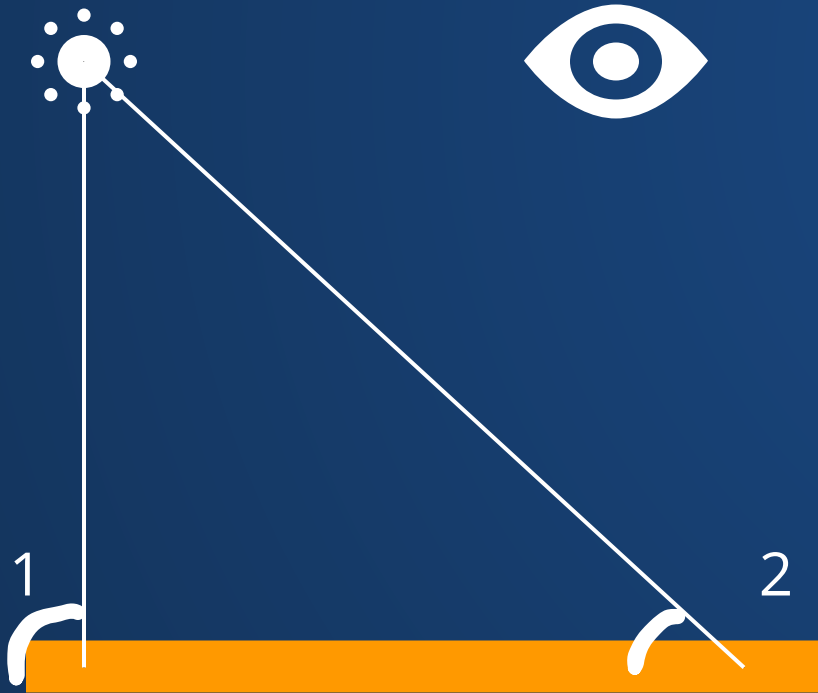
Can also be used *artistically*.



Diffuse

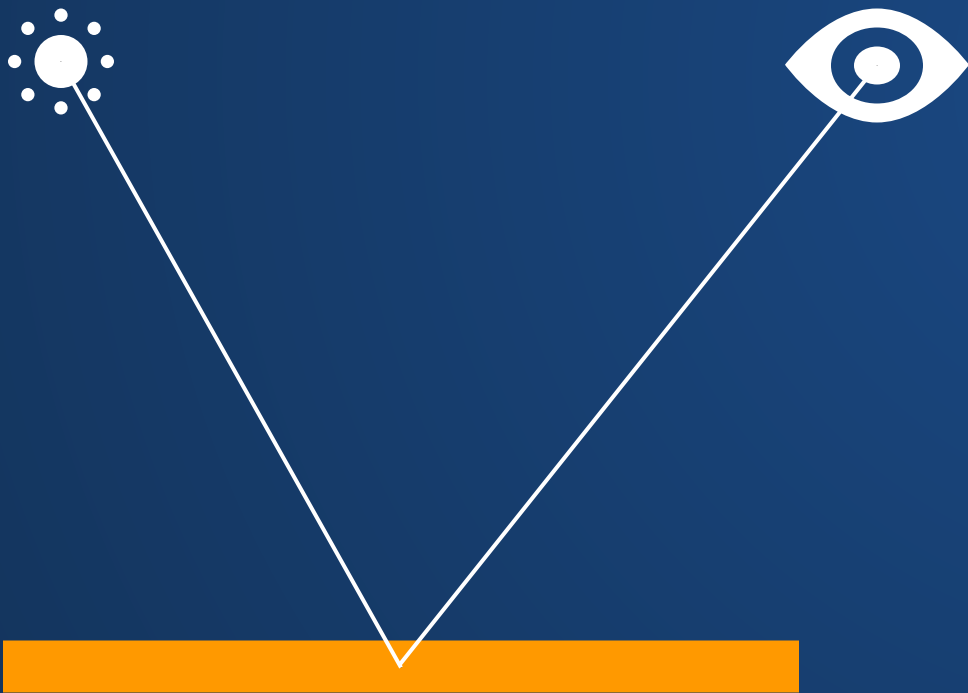
In fancy speak, "Lambertian Reflectance." Captures the light that is scattered off of the surface due to micro-irregularities in the surface.

Is based on the cosine of angle between the light and the surface.

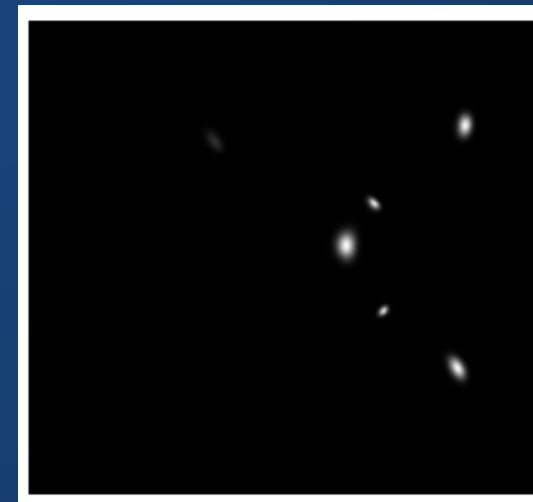


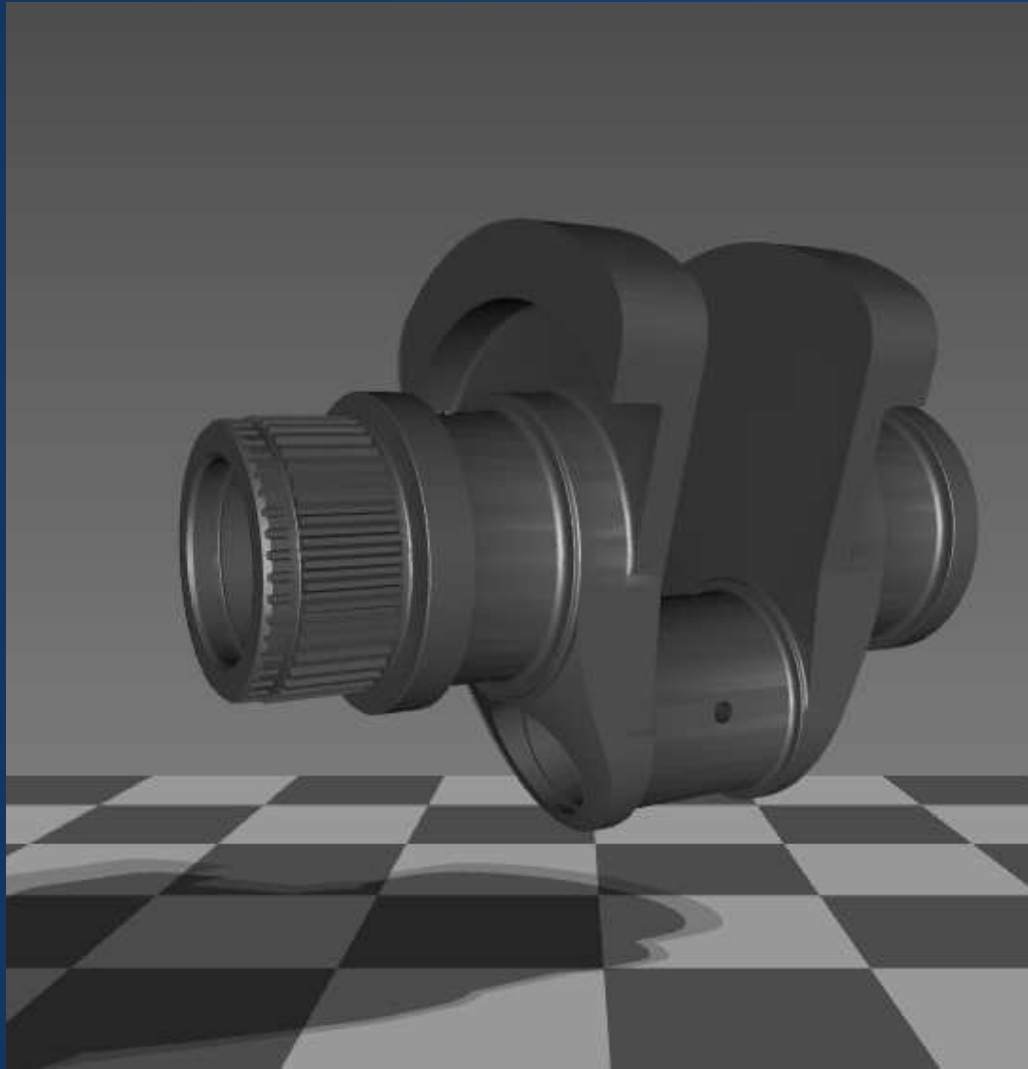
Specular

Responsible for object "shininess". Based on many things: material parameters, angle between light and surface, *and* angle between surface and camera.



Typically has a narrow angular region where it is very intense, but falls off quickly.





Depending on the exact parameters, Phong shading tends to create a plasticky, shiny look which is somewhat unrealistic.

But it's *very* fast (implemented in the early 1970s!) and simple to understand.

Materials in Processing

All material properties function like attributes: they set the material properties for any subsequent objects generated.

Ambient: `ambient(r, g, b)`

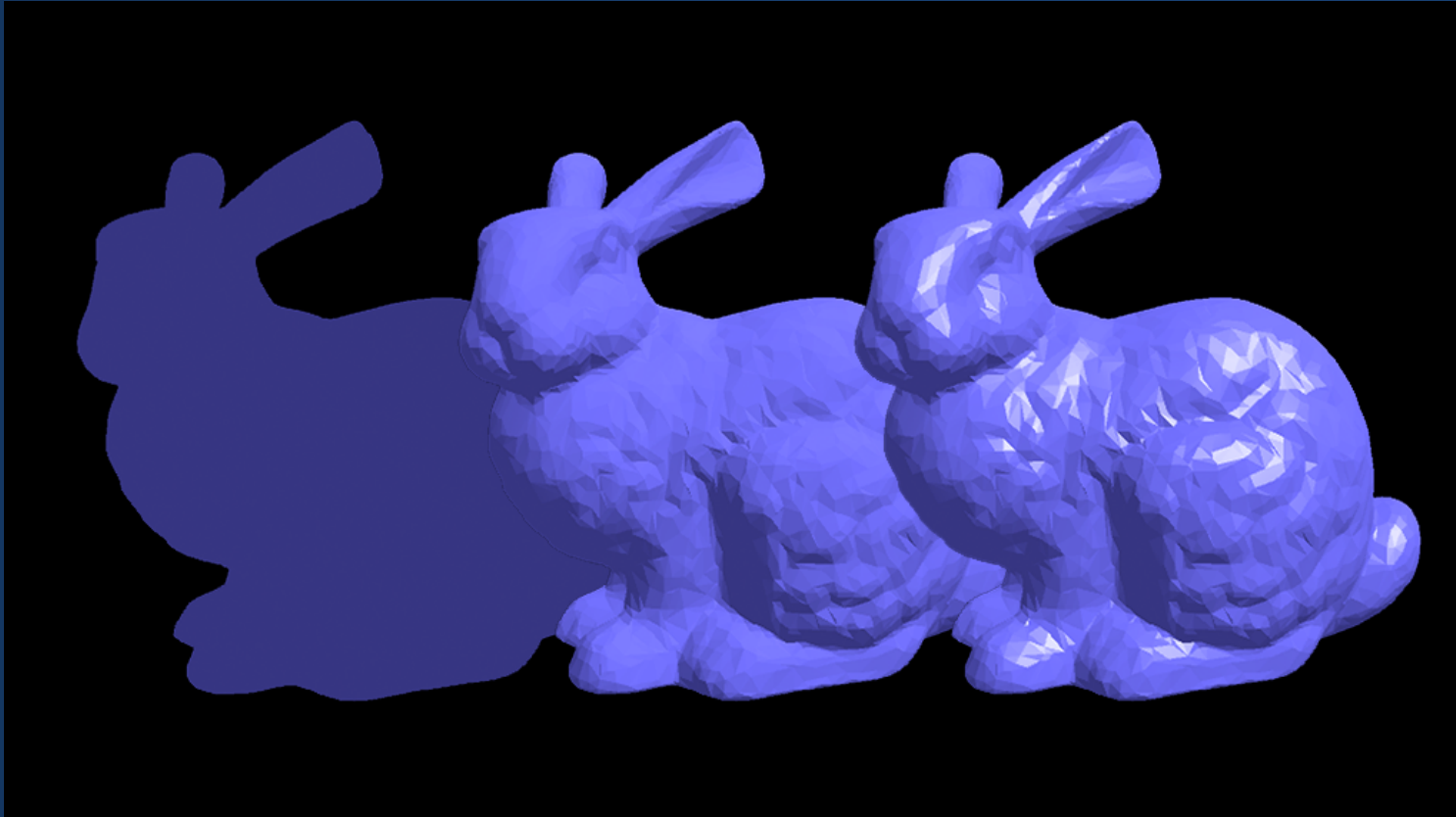
Diffuse: is built into the Processing lighting models, cannot be changed.

Specular:

- `specular(r, g, b)`
- `shininess(s)`
- `lightSpecular(r, g, b)`

Materials Demo

What are the material properties of



- A brick wall?
- The hood of a car?
- An unglazed clay pot?
- A glazed clay pot?

Advanced Materials

The Phong shading model can't capture everything!

Many interesting materials require us to think about light bouncing around within the material, not just at the surface.

But this is a lot harder to replicate.

Requires much more advanced mathematics and a lot more compute.

Fun fact: SSS on marble is one reason why so many statues are made of it.



Adding Details

Think back to the notebook cover shown earlier in class.

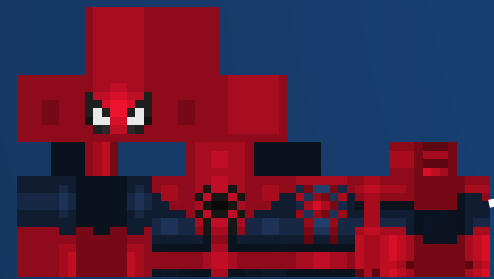
How can we start to replicate the patterned look on the surface of this notebook?



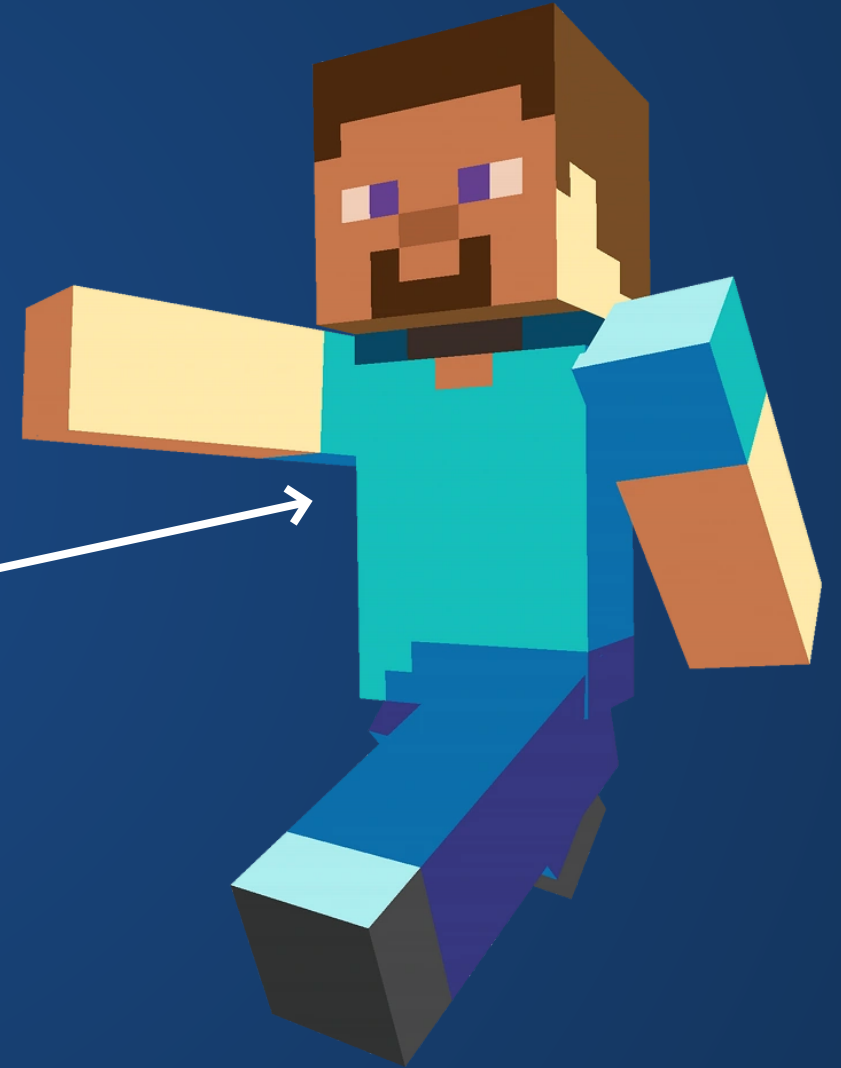
Textures

Provides details on surface of geometry,
deforming with the geometry.

Challenge: we need to map between the
surface of the geometry and the
coordinates of the texture.

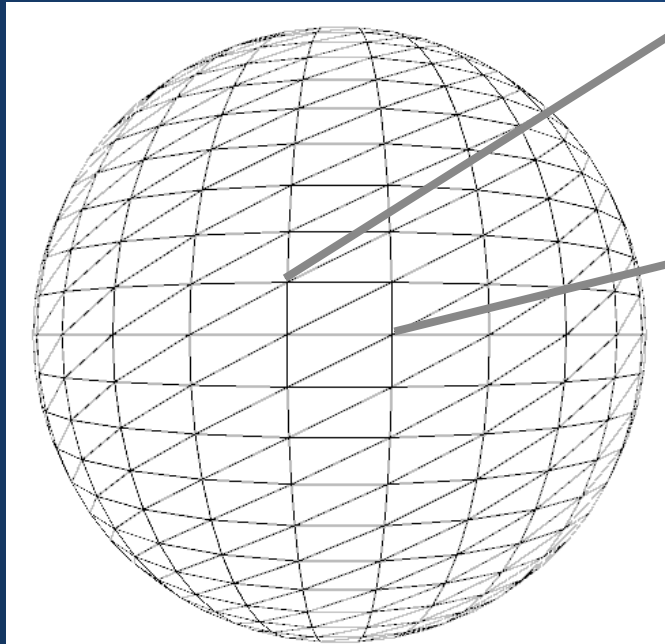


How do we make **this**
display on **that**?

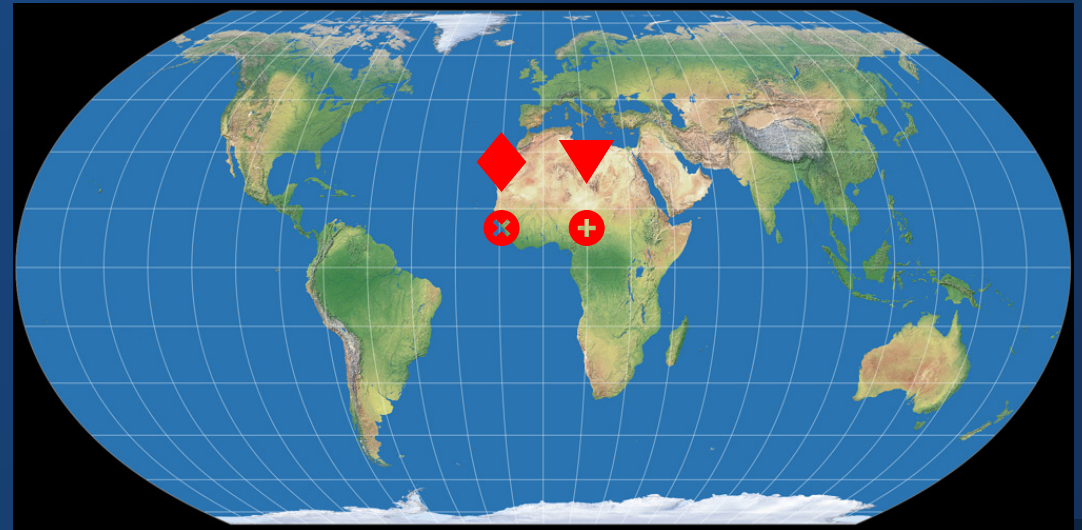
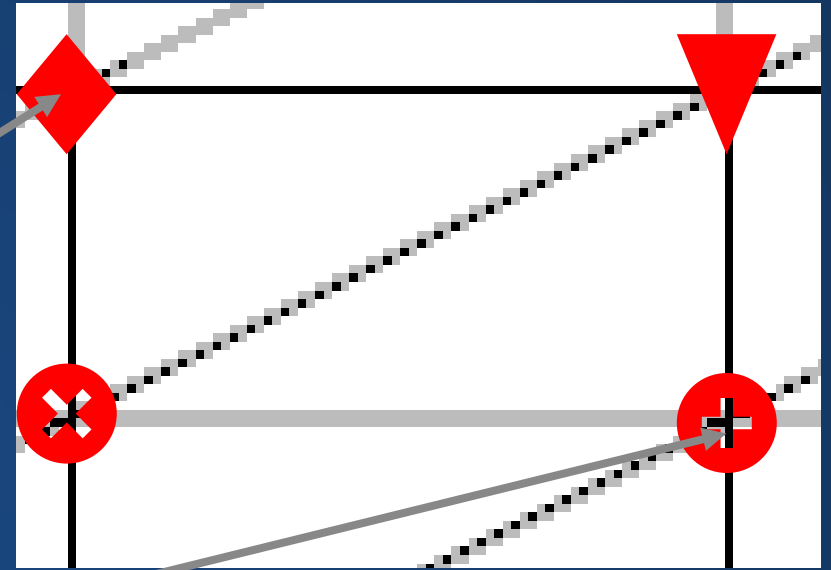


Basic Idea

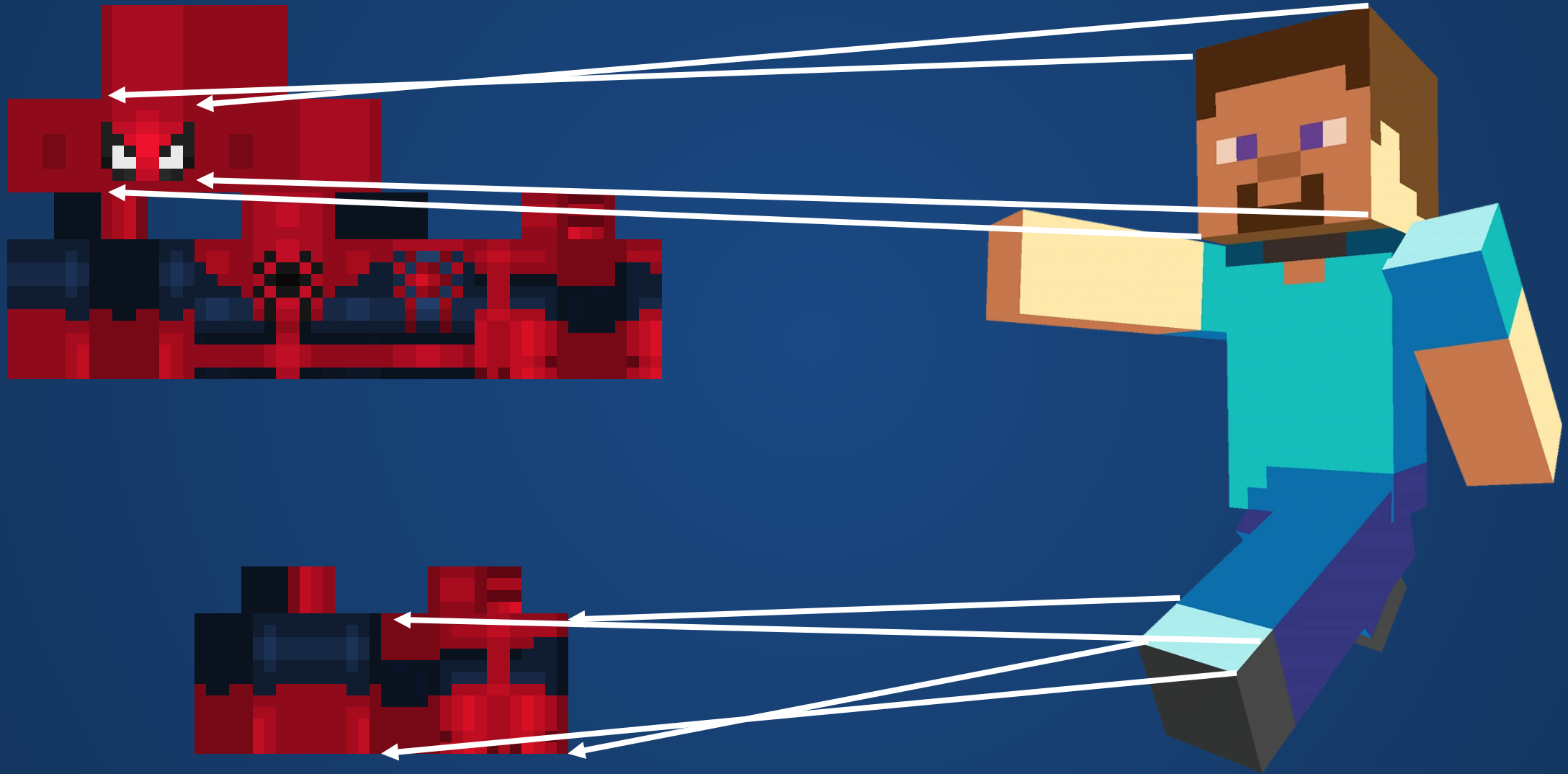
Assign each part of the geometry some coordinate in the texture, then move between them smoothly.



Zoom



Basic Idea



Texture Demo

How would the following code change the texture?

```
vertex(0, 0, 0, 0, 0);  
vertex(350, 0, 0, .5, 0);  
vertex(350, 200, 0, .5, 1);  
vertex(0, 200, 0, 0, 1);
```

textureMode and textureWrap

Sometimes you don't know how large the image is, so specifying coordinates in pixels is annoying. `textureMode(NORMAL)` sets coordinates to be in normalized image coordinate (0.0-1.0)

The default behavior is given by `textureMode(IMAGE)`.

Sometimes you want to wrap the texture by repeating it over the shape (e.g. if you have a texture of bricks and are texturing a brick wall). You can achieve this with `textureWrap(REPEAT)`.

The default behavior is given by `textureWrap(CLAMP)`

Texturing Meshes

We can manually apply textures to meshes with processing.



But it's a lot easier to use dedicated programs like Maya or Blender, and then import their meshes!

Hands-On: Textures

1. Use a scene you built for one of the previous hands-on activities
2. Change the material properties of the 3D objects, modifying their color (and optionally, specularities).
3. Create a simple Shape and apply a texture to it. You can download an image from Canvas if you don't have one you'd like to use.
4. Experiment with texture mode and texture wrapping options

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.