E06: Shadow

Since you can sample a texture more than once, you can do all sorts of simple effects. Keep some things in mind though. The default blending mode for sprites is premultiplied alpha blending, and this should be your go to blending mode. It will allow you to composite multiple additive or alpha blended layers in the shader and output a single premultiplied value for all of them. If you need to mix other kinds of blending together, you might need to render using multiple sprites stacked on top of one another. Also, because you cannot render outside of a sprite's bounds, make sure you have enough transparent padding around your sprite for any effects you plan to add.

Here is an example shader that uses the texture's alpha to composite a shadow layer underneath the sprite. As long as the shadow is black, it can use alpha blending to composite it.

```
vec4 composite(vec4 over, vec4 under){
  return over + (1.0 - over.a)*under;
}

void main(){
  vec4 textureColor = cc_FragColor*texture2D(cc_MainTexture, cc_FragTexCoord1);

  // Offset of the shadow in texture coordinates.
  vec2 shadowOffset = vec2(-0.03, -0.03);
  float shadowMask = texture2D(cc_MainTexture, cc_FragTexCoord1 + shadowOffset).a;

  const float shadowOpacity = 0.5;
  vec4 shadowColor = vec4(vec3(0.0), shadowMask*shadowOpacity);

  gl_FragColor = composite(textureColor, shadowColor);
}
```

The `composite()` function is simple enough. It should look familiar as the linear interpolation formula, but with one of the multiplies removed. Since the color's are premultiplied already, there is no need to do it again. The output color is even properly premultiplied. Handy!

The `main()` function starts off simple enough by reading the texture normally and again at a constant offset. It only keeps the alpha value of the second sample since it will only be used for a mask.

Next it calculates the shadow color. Since the shadow is black, it's already properly premultiplied no matter what the alpha is (0 times anything is still zero).

Exercises:

- Can you figure out how add a colored layer instead of a shadow underneath? You'll need to calculate a premultiplied color.
- Can you figure out how to make the shadow respect the sprite's alpha?

- Can you figure out how to apply an inner shadow instead?