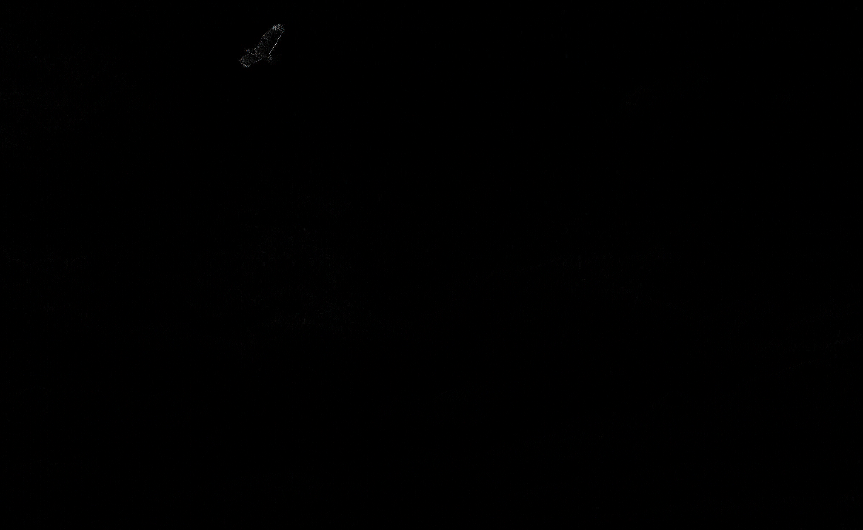
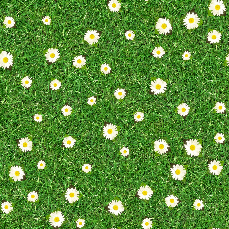
The only file in sunflow that was altered was bump\_demo.sc. Everything that was created can be found in the Blending class.

 The goal of the project was to synthesize textures using image composition, and then render a scene in sunflow using the newly synthesized textures. First in order to create textures that would need to be corrected, we used Poisson disk sampling on a texture to map an image onto that texture multiple times, with the size and rotation of the image randomly changed every time it is mapped onto the texture. Originally we decided to try to implement the image compositing as was discussed in class, where first the gradient of each pixel in the image is found by comparing the red, green, and blue values of the pixel with those of its neighbors. From there the gradient is used to calculate the laplacian, which if converted into an image would create an edge detector of the original image. Unfortunately the next step, the Poisson equation, was too difficult for us to implement, and we believed that we should implement the everything ourselves. Next we tried using laplacian pyramids to composite the image. Laplacian pyramids are built by first building a gaussian pyramid. The gaussian pyramid involves taking an image, performing a gaussian blur on it, then shrinking the image to half size. This process is continued until the image is 1x1. Each successive image makes up a level of the pyramid. To make the laplacian pyramid, each level of the gaussian pyramid is subtracted from the previous level. This leaves us with a laplacian pyramid of size N-1. Finally, the laplacian pyramids of each image are combined by a blending function, in this case a gaussian pyramid built from a blending image. The output pyramid is then collapsed and gives the final image. Our program was almost able to successfully composite an image, instead it seems to do it too well. The image composited looks transparent. Another problem is the rendered image is very blurry, with some strange seams in the image.

Contributions:

Matt Mahoney: Poisson equation rendering, sunflow image rendering, finding textures

Michael Meli: Poisson disk sampling, problem research, final program implementation