CS 7490, Spring 2016

Homework 4: Procedural Textures

Due: 11:55pm, Friday, April 8, 2016

Objective

The topic of this project is procedural textures. You will create three different 3D procedural solid textures: wood, marble, and stones.

Scene Description Language Extensions

Below are the new commands that you will add to your ray tracer in order to give it new capabilities.

- **noise scale**
  This command replaces the diffuse component of the last material that was defined (diffuse or shiny) with a 3D noise texture. The scale factor should allow you to create noise at different scales. You should use the provided .pde code for Perlin noise for this. Note that the noise function provided returns values in the range [-1,1].

- **wood**
  This command replaces the diffuse component of the last material that was defined (diffuse or shiny) with a 3D procedural wood texture. Your wood should have alternating rings of dark and light brown. You should also be sure to add some randomness, so that it does not look too perfect. You don't have to match the colors and the look of the example wood in the images below -- be creative and see if you can come up with something you like.

- **marble**
  This command replaces the diffuse component of the last material that was defined (diffuse or shiny) with a 3D procedural marble texture. This should be similar to Ken Perlin's marble texture on the vase in his paper. You should implement turbulence (using 3D noise) in order to create convincing marble. You can be creative about the kinds of colors you use in your marble.

- **stone**
  This command replaces the diffuse component of the last material that was defined (diffuse or shiny) with a 3D procedural texture texture of stones that are separated by cement. This should be similar to Stephen Worley's figure of the stone torus. You should implement this using the cellular texture basis of Worley. Be sure to include some material in the cracks between the stones. As with the other textures, the colors you use are up to you.

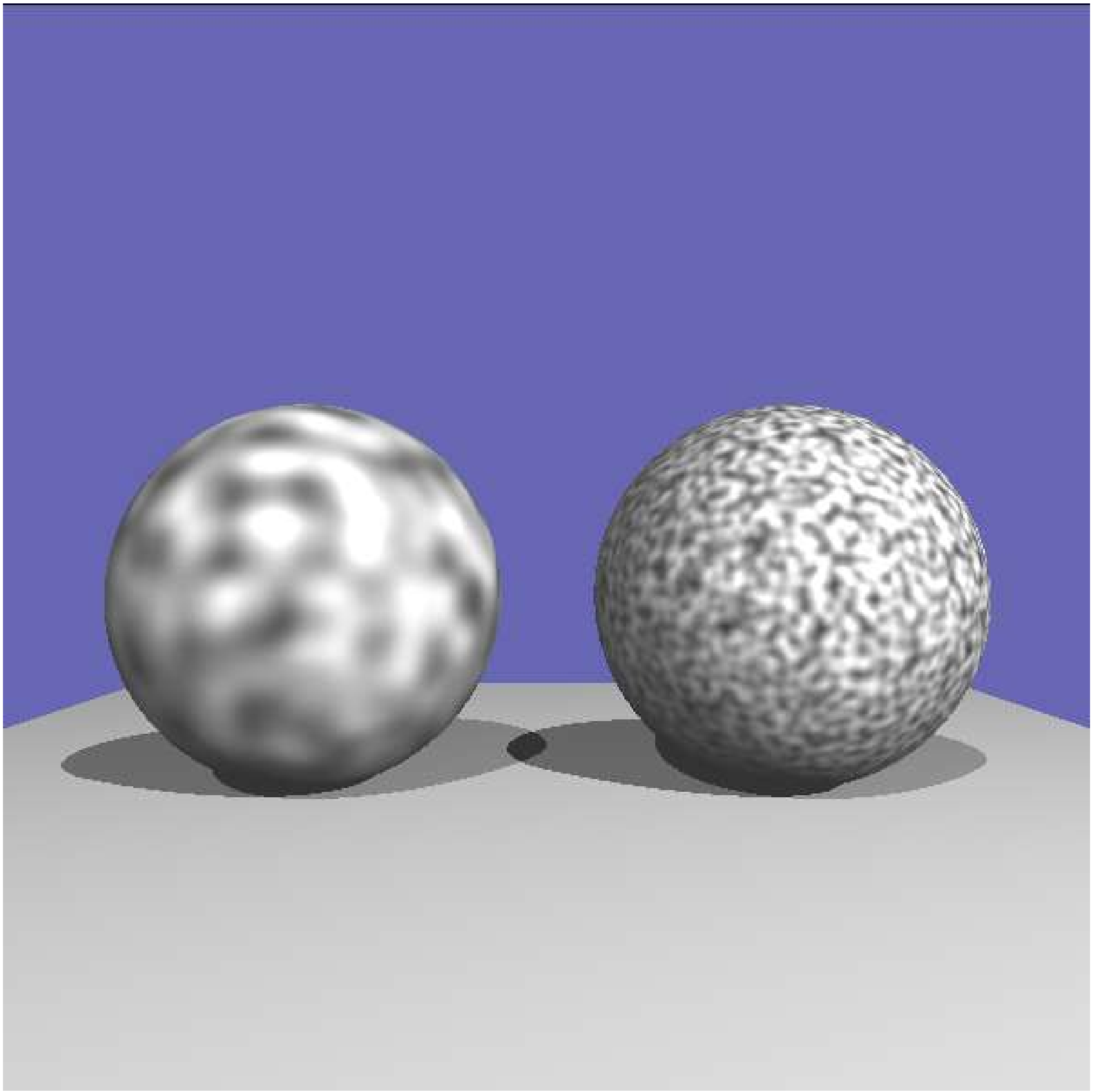
Provided Code

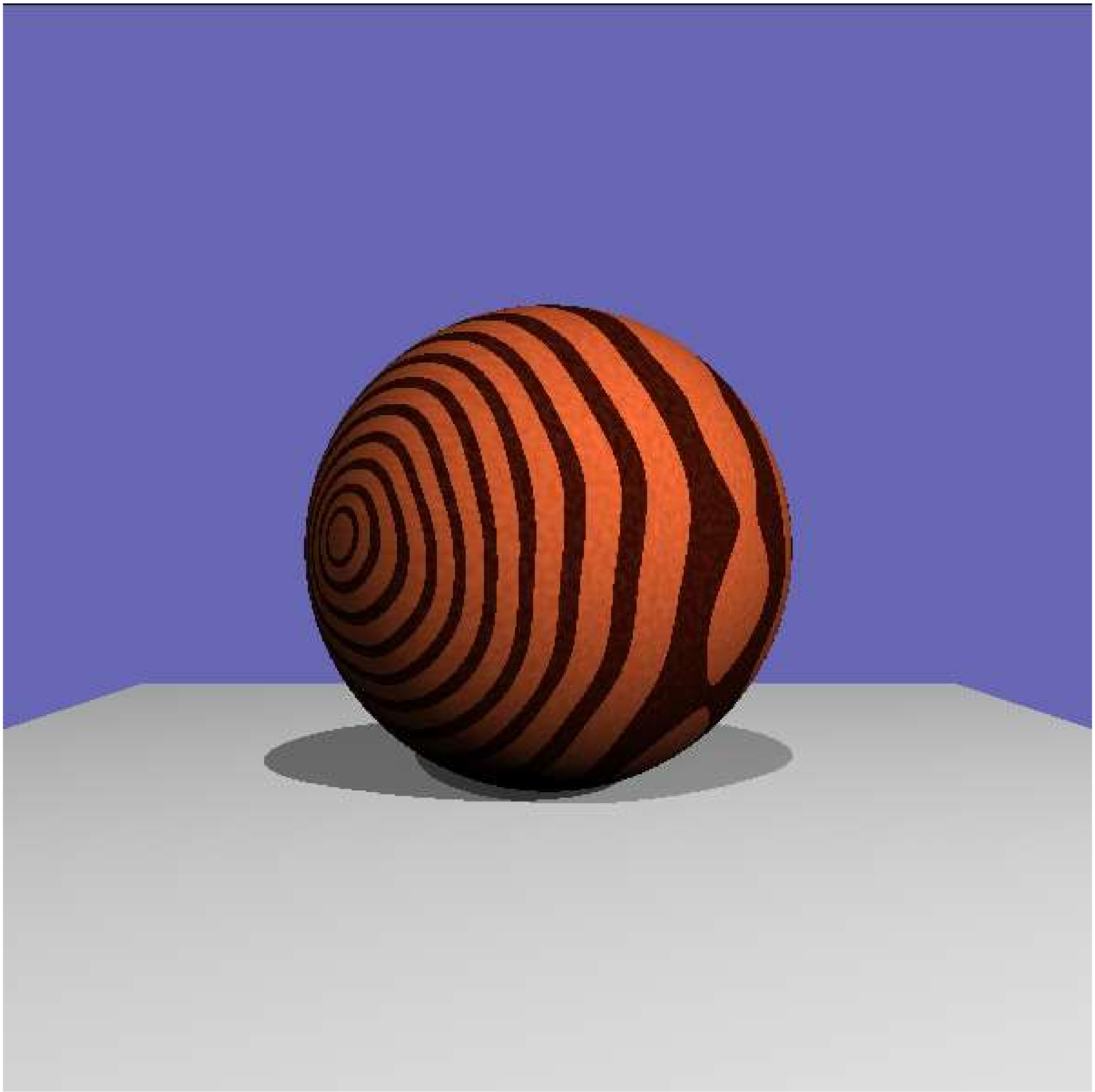
We provide source code (noise.pde) that implements Perlin's 3D band-limited noise. Please note that although Processing has a built-in noise function, it is has different characteristics than the provided noise code.

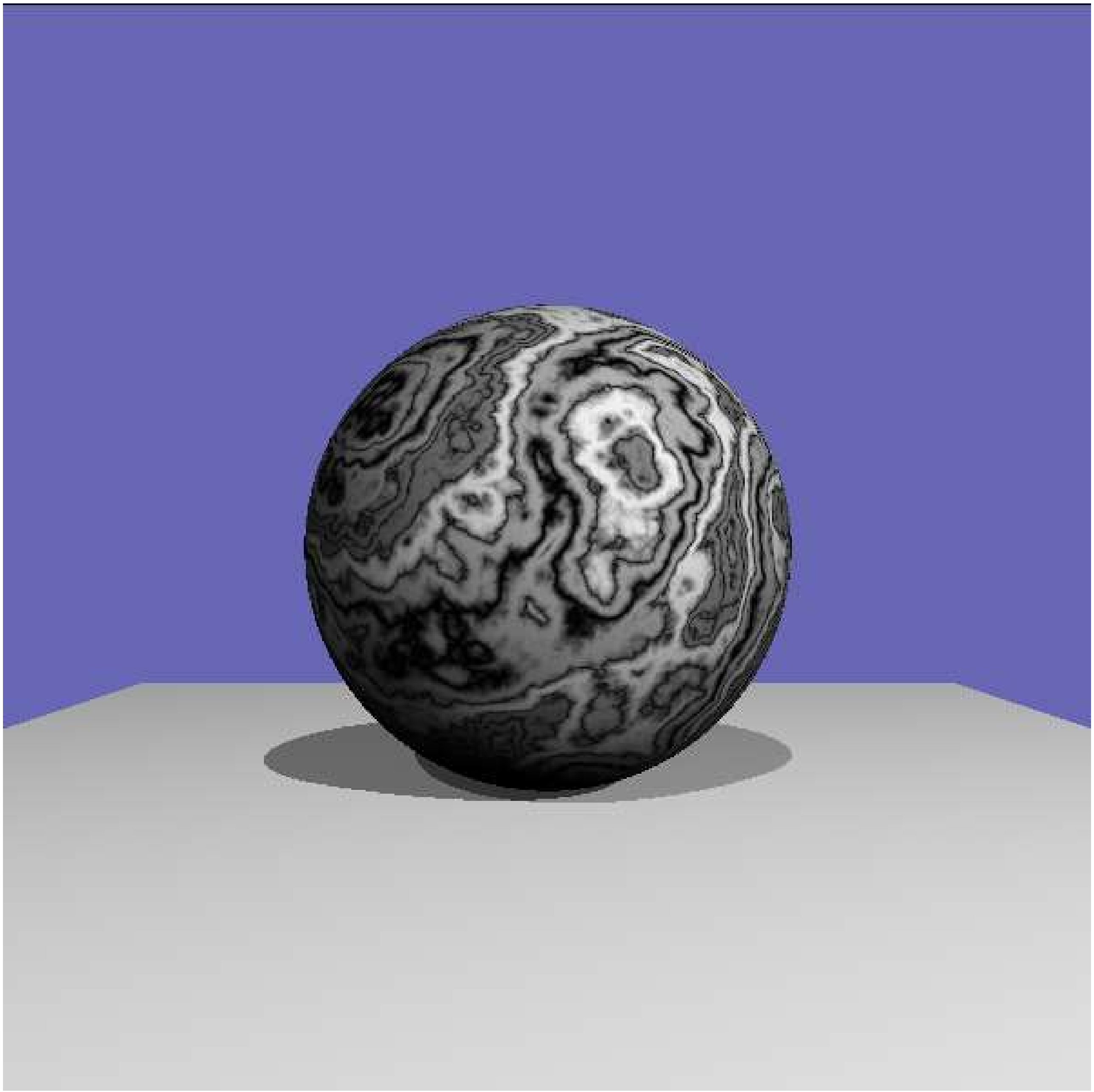
Scene Files

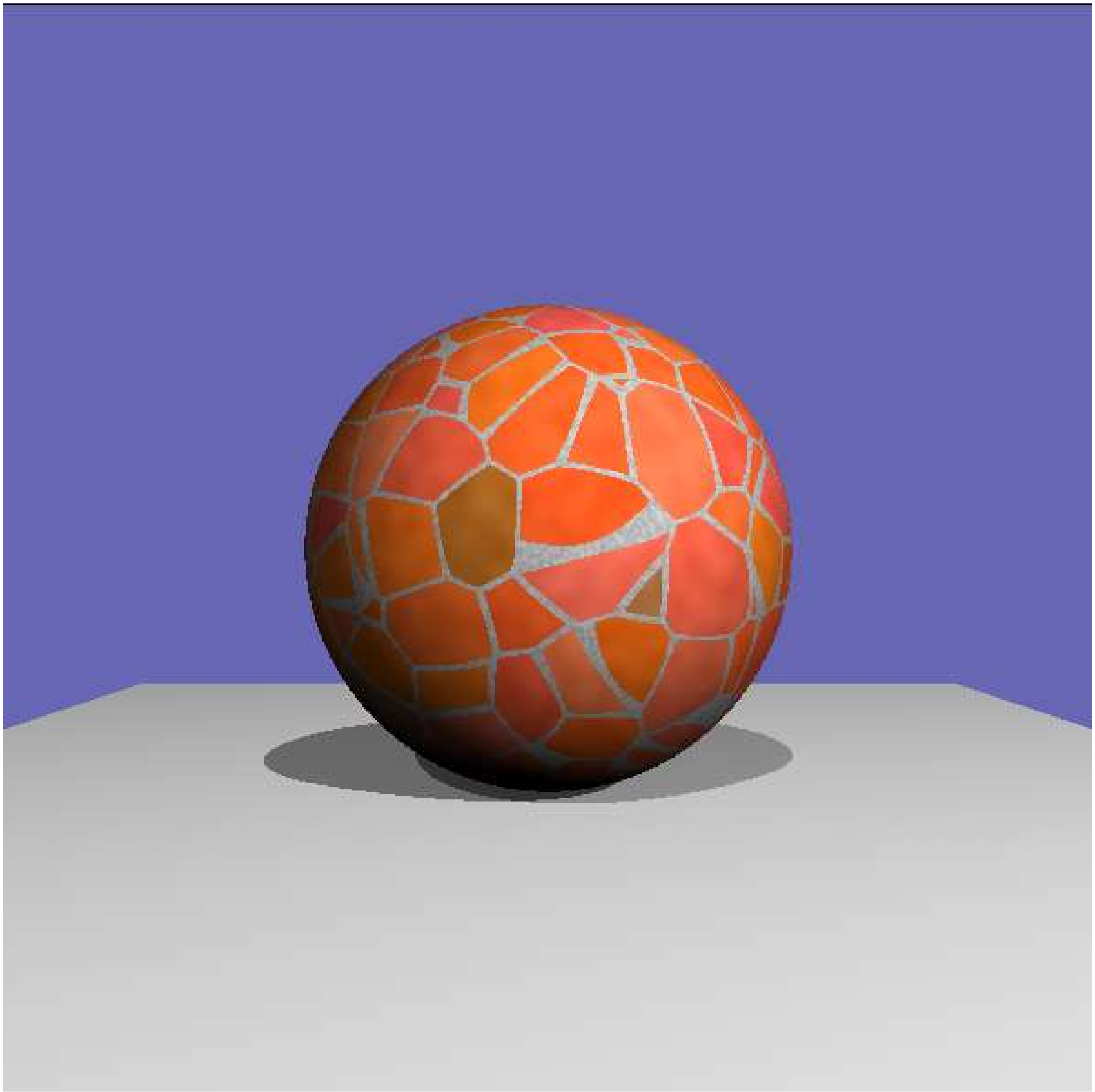
In the directory "**data**" are several test scenes that are described by .cli files. Also in that directory are the images that should be created by these scene files. The provided source code, CLI and image files are in a zip file. Pressing keyboard keys 1-4 calls these test CLI files.

Sample Results:









Suggested Approach

First, you should modify your ray tracer to include the noise command (using the provided code). After getting noise to work, the wood texture is probably the best one to work on next.

Authorship Rules

The code that you turn in must be entirely your own. You are allowed to talk to other members of the class and to the instructor about high-level questions about the assignment. You may not, however, use code that anyone other than yourself has written. Code that is explicitly **not** allowed includes code taken from the Web, from books, or from any source other than yourself. The only exception to this rule is that you should use the parser routines that we provide. You should not show your code to other students. If you need help with the assignment, seek the help of the instructor.

Development Environment

You must use the Processing language which is built on Java. Be sure that you are using Processing version 2.0 or higher. The best resource for Processing language questions is the online or offline Processing language API (found in the "reference" subdirectory of the Processing release).

What To Turn In

Compress the whole folder for this project (not merely the files within the folder) into a zip archive submit them to T-square. The zip archive should be included as an attachment. When unzipped, this should produce a folder containing all of your .pde files and a directory "data".