Report: Computer Graphics **Xucheng Chen**
COMS 4160 xc2360
Prof. Zheng Date: 02/23/2017

Running the code

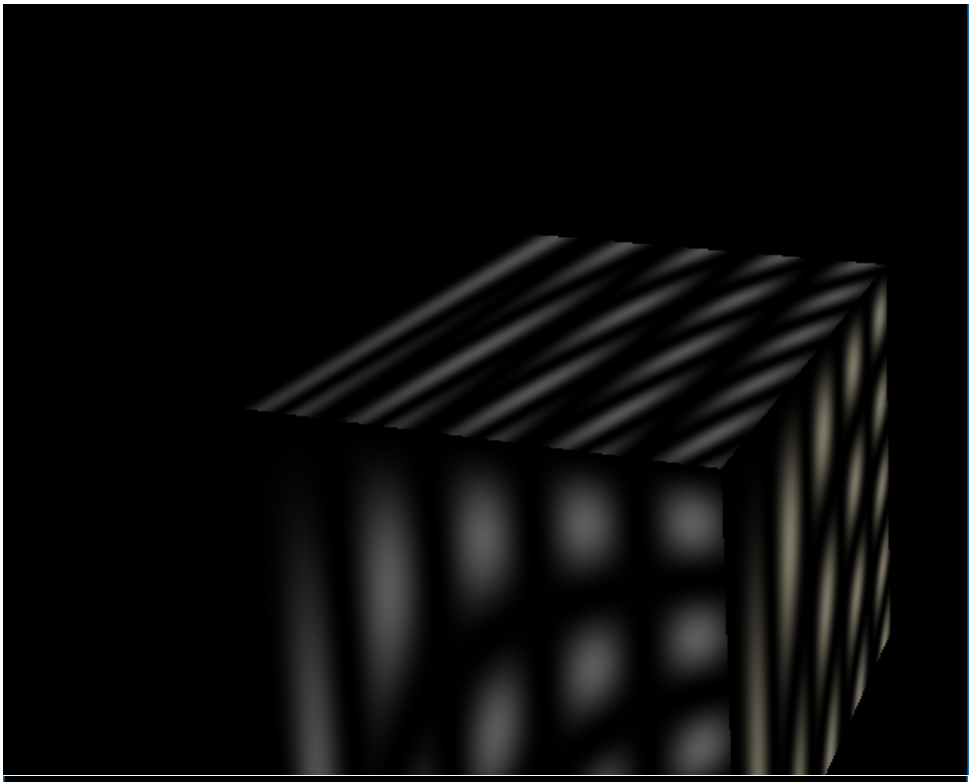
I basically finished all coding in eclipse, so the easiest way to compile and run the code would be just import the code in eclipse and then configure the build path to include the lwjgl library. After importing the project, just run Main.java:). If you have any problems running the sample code, please contact me and I will personally demo it in office hours.

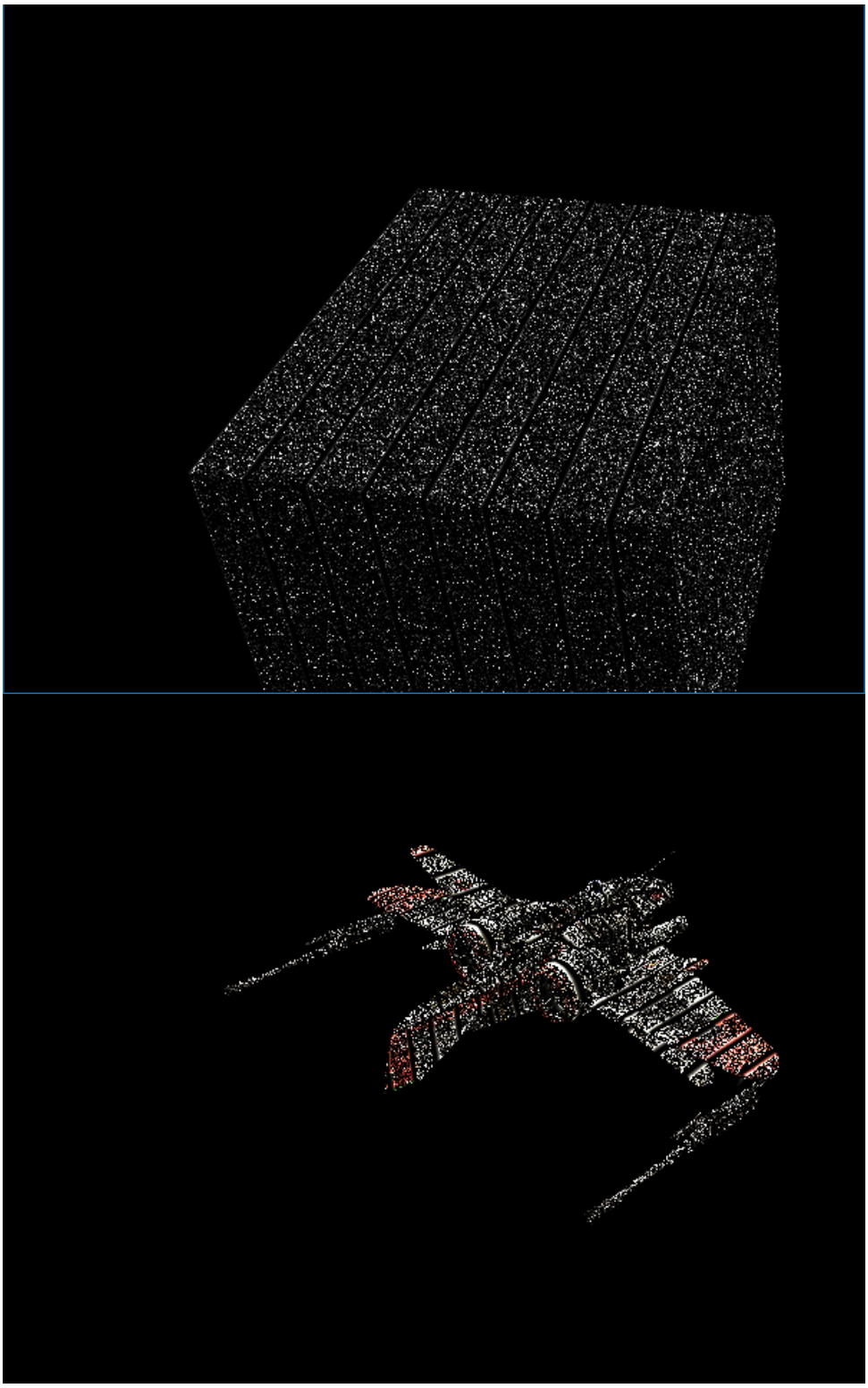
External resources

The normal mapping shader is using the code from official tutorial from lwjgl, and checkerboard shader is inspired by tutorial: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/procedural-texturing

Creative scene

The initial idea is to simulate something like fractal geometry, but later I realized it is kind of hard to generate such pattern. Instead I've generated several special patterns using the iterative algorithm. In the video I provided, I rendered the checkerboard pattern on my Arc170 and there are several other patterns which can be found in checkerboard fragment.fs. Just uncomment some of them so you can see other patterns. Also, I have taken some screenshots of these patterns.





Installation

The following installation guide is copied from the official website of lwjgl: Eclipse supports Gradle/Maven projects and it is highly recommended to use them instead of a manual Eclipse project configuration. However, if you prefer setting up a native Eclipse project, follow these

instructions (works with Eclipse Neon):

- Download the ZIP bundle from https://www.lwjgl.org/download
- When the download is complete, extract its contents to some file system directory, henceforth referred to as <lwjgl3>.
- In Eclipse go to menu "Window" > "Preferences" and in the tree view to the left search for 'Java' > 'Build Path' > 'User Libraries' and Click "New..." in the 'User Libraries' dialog. In the opened modal dialog "New User Library" write "LWJGL3" in the 'User library name:' text field and click 'OK'. The newly created library "LWJGL3" should show up in the list 'Defined user libraries:'.
- Now, select this item in the list and click 'Add External JARs...'. This opens a standard OS file selection dialog allowing you to select *.jar files which get added to the classpath/buildpath of all projects using the LWJGL3 User Library. Go to <lwjgl3> and select all *.jar files which do not have -javadoc or -sources in their names. Make sure you don't forget the lwjgl-natives-<os>.jar file, and click 'Open'. This will populate the LWJGL3 user library in the list with respective entries for all selected jar files. You could leave it at that now in order to use LWJGL 3.
- However, if you want to have Sources and JavaDocs, you will have to select each of the entries, click on 'Source attachment: (None)' and on 'Edit...'. This will open the "Source Attachment Configuration" dialog. Here you could select 'External location' and 'External File...' to select the appropriate *-sources.jar file.
- In order to actually use the LWJGL3 User Library in one of your projects, go to the Build Path settings of your project and select the 'Libraries' tab. Here, click 'Add Library...', select 'User Library' and mark the "LWJGL3" User Library.
- Now you are set up to use LWJGL 3 in your project.

Additional files

The video is include in the video file.