UCSD CSE 168 Assignment 1:

Ray Tracing

In our first homework, we are going to implement a simple ray tracer (almost) from scratch. We will provide some utility code such as the vector class and parallel threading. However, you will have to implement the rest yourself. Trust me, it will be fun!

Our homeworks are mostly inspired by the book Ray Tracing in One Weekend (click the link for free access of the e-book). You are expected to read the relevant chapters of the book before implementing your own version. The structure of the homeworks is also largely inspired by CS 87/287 at Dartmouth designed by Wojciech Jarosz (a UCSD alumni!).

0 Building Torrey

We are going to build our code on top of (a currently very barebone) renderer *torrey*. Torrey already includes all the third party libraries (pugixml, pcg, stb_image, tinyexr, miniz, tinyply) in its repo and all you need to do is to clone the repo and build it using CMake (assuming you are in a Unix-like system):

```
git clone https://github.com/BachiLi/torrey_public
mkdir build
cd build
cmake ..
make -j
```

After building, you should see an executable `torrey`. Try typing the following command:

```
torrey -hw 1_1
```

It will generate an image `hw_1_1.exr` that looks like the following:

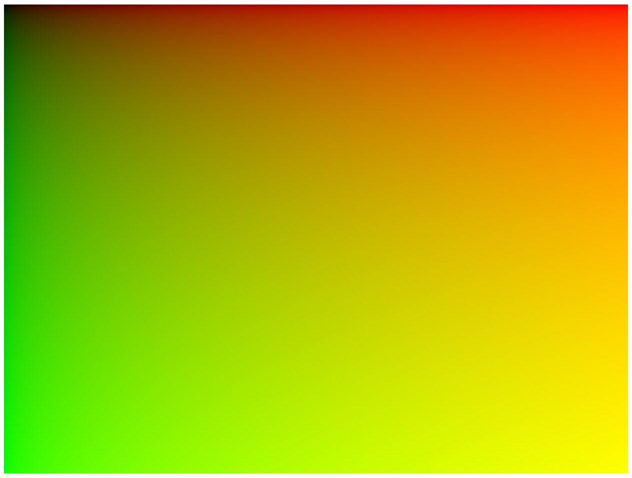

Figure 1: Image output by the homework 1.1 code before your modification.

`.exr` is an image format that is suitable for storing *high-dynamic range* images. Basically, instead of storing 8-bit (0-255) per color channel, we store a floating point number per channel. To view `.exr`, I recommend using HDRView or Tev.

Read `main.cpp` and `hw1.cpp` to understand the current code structure.