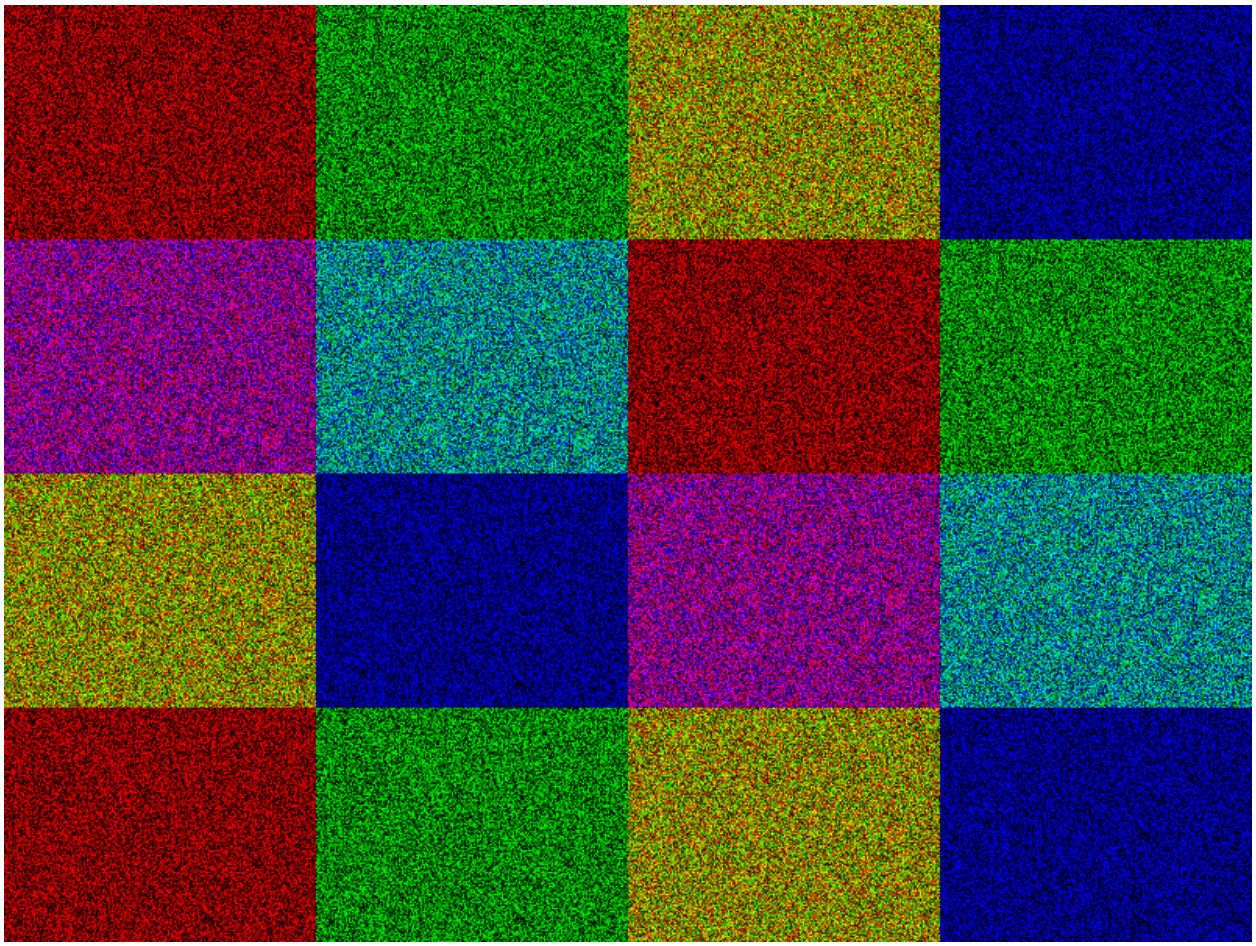
For the programming task you have to use C++
For questions and help refer to the course's discord server
Or the course's e-mail:
raytracingcourse@chaos.com

Slides: CRT 02 Fundamentals

Task 1.

Generate a .ppm file that represents an image with a resolution you specify. The image should be divided into rectangles (the number of which is your choice), and each rectangle should be distinct from its neighbors. For example, each rectangle could contain pixels with a random color based on one of the three components.

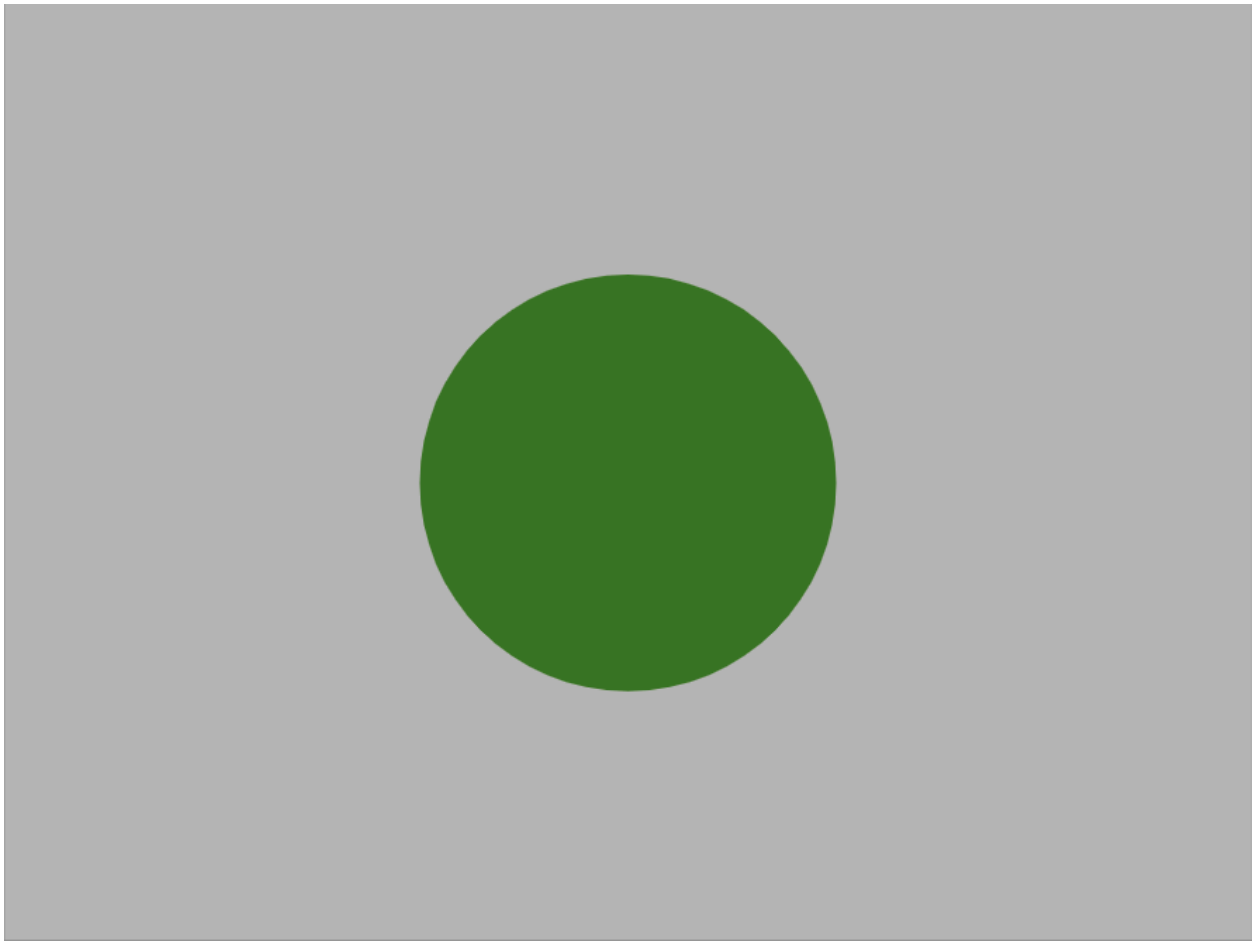
Example:



Task 2.

Generate a .ppm file that represents an image where the pixels within a 2D shape of your choice have a color different from the rest.

Example:



Example C++ code, with which a .ppm file could be generated:

```
#include <fstream>

/// Output image resolution
static const int imageWidth = 1920;
static const int imageHeight = 1080;

static const int maxColorComponent = 255;

int main() {
	std::ofstream ppmFileStream("crt_output_image.ppm", std::ios::out | std::ios::binary);
	ppmFileStream << "P3\n";
	ppmFileStream << imageWidth << " " << imageHeight << "\n";
	ppmFileStream << maxColorComponent << "\n";
```

```
	for (int rowIdx = 0; rowIdx < imageHeight; ++rowIdx) {
		for (int colIdx = 0; colIdx < imageWidth; ++colIdx) {
			ppmFileStream << "0 0 255\t";
		}
		ppmFileStream << "\n";
	}

	ppmFileStream.close();

	return 0;
}
```