# cs805 Assignment 2

Ray Shulang Lei
200253624
Department of Computer Science
University of Regina

October 18, 2012

**Abstract**

This assignment is written in literate programming style, generated by noweb, rendered by LaTex, and compiled by clang++ with c++ 11 standard.
assignment paper is at latex/as2.pdf
c++ programs are at src/*
binary executable for OS X 10.8 is inside bin

# 1 The foreach pixel function

First I will define a function, named foreach_pixel_exec, that executes a function on each pixel of a 2D image panel.

```
<<src/util.cpp>>=
#include "util.h"
#include <math.h>

ImagePanel foreach_pixel_exec(ImagePanel empty_img, std::function<int(int)> ray_fu
  for (auto& pixel: empty_img) { //foreach pixel in empty_img
    //std::cout<<"before: "<<pixel<<std::endl;
    pixel = ray_func(pixel);
    //std::cout<<"after: "<<pixel<<std::endl;
  }
  return empty_img;
}

ImagePanel init_img_panel(ImagePanel img) {
  for (auto& pixel: img) { //foreach pixel in empty_img
    pixel = 0;
  }
  return img;
}

//helpers
void print_img_panel(ImagePanel img) {
  std::cout<<std::endl;
  for (auto& pixel : img) {
    std::cout<<pixel<<", ";
  }
  std::cout<<std::endl<<"Array size: "<<img.size()<<std::endl;
}

@
```

Here is an header file for typedefs and function declarations.

```
<<src/util.h>>=
```

```
#ifndef UTIL_H
#define UTIL_H

//define global vars
#define IMG_X 320
#define IMG_Y 240
#define IMG_LEN ( IMG_X * IMG_Y )

#include <array>
#include <functional>
#include <iostream>
typedef std::array<int, IMG_LEN> ImagePanel;
typedef std::array<float, 3> Ray;//assuming there are 3 parameters for ray equatio

ImagePanel foreach_pixel_exec(ImagePanel, std::function<int(int)>);
ImagePanel init_img_panel(ImagePanel);

//helpers
void print_img_panel(ImagePanel);
#endif
@
```

```
<<src/main.cpp>>=
#include <iostream>
#include <typeinfo>//debugging only
#include "util.h"

int main () {
  ImagePanel resultImg;
  resultImg = init_img_panel(resultImg);
  resultImg = foreach_pixel_exec(resultImg, [](int x){return x+2;});
  print_img_panel(resultImg);
  return 0;
}
@
```

Furthermore, this is the command to link these files. Notice that I am using -std=c++11 flag to enable c++ 11 features. The output binary executable is bin/run

```
<<compile.sh>>=
clang++ -std=c++11 -stdlib=libc++ -o bin/run src/main.cpp src/util.cpp
@
```