CS 580 – Discussion Setting up and HW 1

Week 1

Yijing Li ***yijingl@usc.edu***

Lixing Liu ***lixing.liu@usc.edu***

Discussion sessions

- Twice a week
  - Tuesday/Thursday 3:30-4:20PM SOS B44

- *Note*: Same sessions on Tuesday and Thursday!

- Taught by TAs

- Support/Discussion on assignments and class material
- Opportunity to ask questions!

Setting up the environment

HW environment:
MS Visual Studio 2015

- You can get visual studio from USC ITS:
  https://viterbiit.usc.edu/microsoft-imagine-downloads/

- Download the “Visual Studio Community 2015 with Update 2”

- *Note*: If you do not have a MS computer, you can use the computers in the SAL lab! They have VS 2015 installed. Also, you can set up virtual machine

Blackboard: How to download assignments

- Go to blackboard.usc.edu and enter your USC credentials.
- On BlackBoard
  - Select your class

    20183_csci_580_30142/30250:
    3-D Graphics and Rendering
  - In the left tab, pick "*Assignments*"
  - Choose your assignment, e.g. "HW1"
  - Download the zip file

How to compile assignments

- Unzip your assignment
- Open the DSW file In Visual Studio
  - For HW1: it is CS580HW1.DSW
- Click build (F7)

- *Warning*: If it does not compile, make sure you opened the DSW file and not the DSP one.

- *Note*: If you get the error "Command line error D8016: '/ZI' and '/Gy-' command-line options are incompatible"
  - *Solution 1*: Use "Release" mode
  - *Solution 2*: Go to "Project">"Properties">"C/C++">"General" and replace the "Debug Information Format" from "/ZI" to "/Zi"

Before submitting assignments

- Delete the "*Debug*"/"*Release*" folders
- Delete the "*ipch*" folder
- Delete the "*SDF*" file, e.g. CS580HW1.sdf
- Delete the "*PPM*" files
- Select everything and ZIP it
  - Right click>Send To>Compressed (zipped) folder
- **Rename your ZIP file as LASTNAME_FIRSTNAME_HW1.zip**

- *Warning*: If your ZIP is bigger than 1MB, there is something wrong!

How to submit assignments

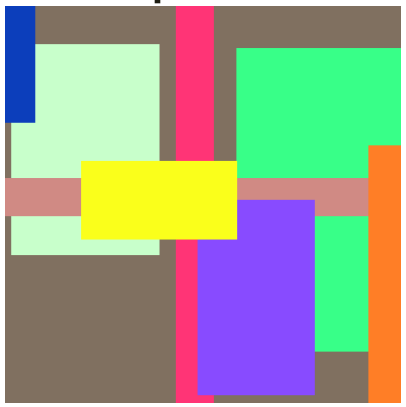
- In BlackBoard:
  - Go to the "*Assignments*" tab
  - Pick your homework, e.g. "*HW1*"
  - Click the "Browse My Computer" button
  - Select your ZIP file
  - Add relevant comment in the text box
  - Click the "Submit" button

- *Note*: You can submit **as many times as you want.** We will use your **latest submission before the deadline**.

HW 1

HW 1 - goal

- The primary goal is to get familiar with the set up.
- You will display an image on the screen. (see "*output1.ppm*")
- The image is made of rectangles (see "*rects*" file)
- You will save that image in a PPM format.

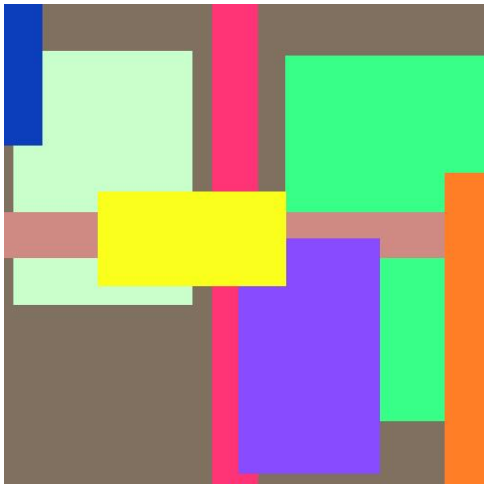


Note: the background color is your choice – it does not have to match this image!

- Warning: To visualize PPM files, you can download the free software Irfanview http://www.irfanview.com/

Input: rects

10	50	200	320	3200	4320	3254
300	55	511	444	900	4200	2189
-100	222	600	270	3333	2212	2121
222	-50	270	588	4321	834	1898
250	250	400	500	2180	1209	5333
100	200	300	300	4000	5000	444
470	180	999	999	4100	2030	620
-100	-100	40	150	200	1000	3000
Top-left corner		Bottom-right corner		Color		
X	Y	X	Y	Red	Green	Blue

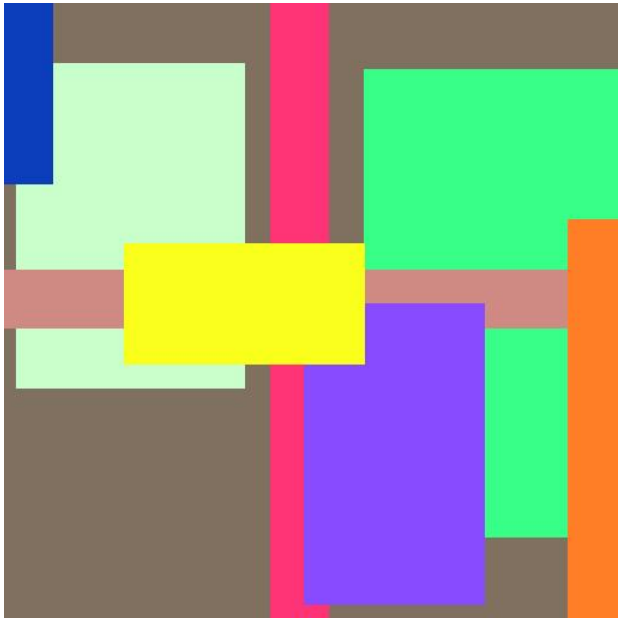


Output: output.ppm

- PPM file format has an ascii header followed by 8-bit binary pixel color values in raster order (Top-Left to Bottom-Right)

For example:
P6 256 256 255\nRGBRGBRGB….
Produces a 256x256 image

Output image



HW 1 - files

- Important files to understand the code:
  - Application1.cpp
  - gz.h
  - rend.h
  - rects
- **Files to modify:**
  - **rend.cpp**

- There are 7 bullet points that need to be coded (from HW1.1 to HW1.7)

rend.h

```
/* define general display pixel-type */
typedef     struct {
  GzIntensity    red;
  GzIntensity    green;
  GzIntensity    blue;
  GzIntensity    alpha;
  GzDepth        z;
} GzPixel;

/* define a display type */
typedef struct {
  unsigned short xres;
  unsigned short yres;
  GzPixel     *fbuf;        /* frame buffer array */
} GzDisplay;

/* put some bounds on size in case of error */
#define     MAXXRES   1024
#define     MAXYRES   1024

/* access pixel (x,y) in a buffer */
#define ARRAY(x,y) (x+(y*display->xres))
```

Notes:

- Pixel structure holds anything we will need in the frame buffer
- Display structure is complete data for the display (an object) – The App should be able to create and use many Displays if needed.

- Do bounds checking and logical correction or error management of
  - xres, yres
  - GzIntensity (RGB)

HW1 pitfalls

- Bounds check the parameters passed to the display functions
  - Pixel coords – ignore off-screen coordinate commands
    => pixels are between 0 and Xres-1, 0 and Yres-1
  - pixel GzIntensity values
    => clamp to 0-4095 within 16-bit short;
- Flush command requires conversion of GzIntensity to 8-bit rgb component
  - Drop LS 4-bits by right-shifting and then use low byte of GzIntensity value
  - (C command for right-shifting by X bits: >> X)
- For the display, the buffer order is BGR!
- For the PPM file, the buffer order is RGB!

- *Before submitting*: check that **both** your output.PPM file and the on-screen result are correct!!!