CSCI 580 Discussion

HW6 Antialiasing

Lixing Liu

Slides modified from: Rongqi Qiu

Grading

- Regrading
  - HW 1 ~ 3 -> grades have been updated
- No regrading
  - HW 4 ~ 6

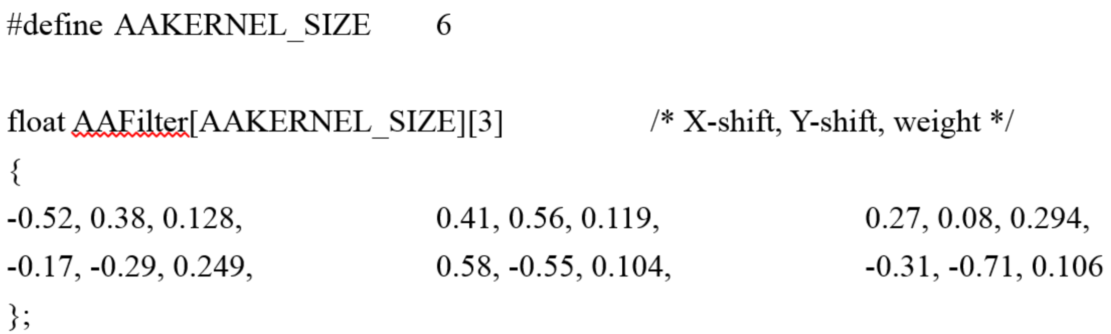
- Check your submission
  - Download the file you submitted
  - Try compiling and running in a new folder
  - Link your files to the correct path

HW6

- No starter code is given
- You should make changes to the HW5 **application** and **rend** to perform multi-render super-sample antialiasing.

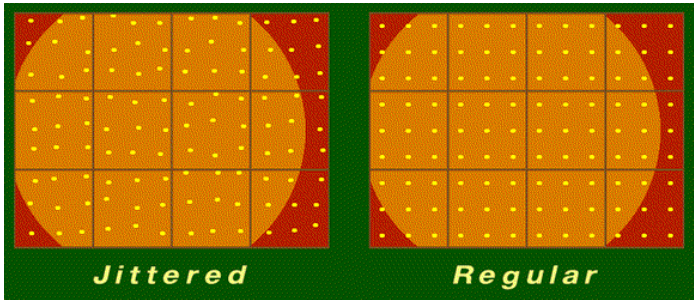
HW6

- Sub-sample pattern is irregular but the same for each pixel



```
#define  AAKERNEL_SIZE       6

float AAFilter[AAKERNEL_SIZE][3]               /* X-shift, Y-shift, weight */
{
-0.52, 0.38, 0.128,                    0.41, 0.56, 0.119,                    0.27, 0.08, 0.294,
-0.17, -0.29, 0.249,                   0.58, -0.55, 0.104,                   -0.31, -0.71, 0.106
};
```



- For each pixel, compute 6 supersamples
- Sum weighted supersamples to produce filtered FB pixels

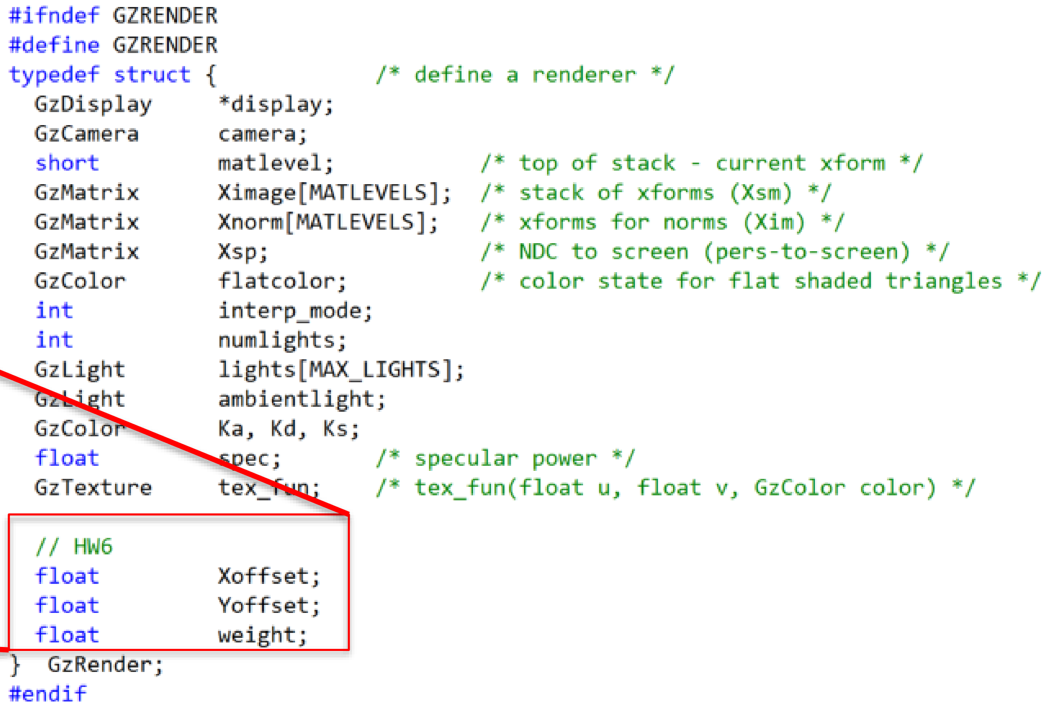
HW6

- Our render already samples the pixel center
- So we only need to shift the image by (-dx, -dy)

- This shift can be easily accomplished by adding (-dx, -dy) to each vertex in **screen space**
  - After transformations
  - Before rasterization

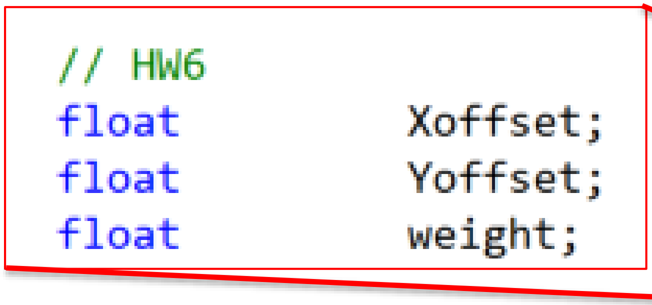
HW6

- Set up multiple renderers and displays.
- Each renderer is initialized with a different offset and its own display.



```
#ifndef GZRENDER
#define GZRENDER
typedef struct {                 /* define a renderer */
  GzDisplay     *display;
  GzCamera      camera;
  short         matlevel;             /* top of stack - current xform */
  GzMatrix      Ximage[MATLEVELS];    /* stack of xforms (Xsm) */
  GzMatrix      Xnorm[MATLEVELS];     /* xforms for norms (Xim) */
  GzMatrix      Xsp;                  /* NDC to screen (pers-to-screen) */
  GzColor       flatcolor;            /* color state for flat shaded triangles */
  int           interp_mode;
  int           numlights;
  GzLight       lights[MAX_LIGHTS];
  GzLight       ambientlight;
  GzColor       Ka, Kd, Ks;
  float         spec;            /* specular power */
  GzTexture     tex_fun;         /* tex_fun(float u, float v, GzColor color) */

  // HW6
  float         Xoffset;
  float         Yoffset;
  float         weight;
}  GzRender;
#endif
```



```
// HW6
float         Xoffset;
float         Yoffset;
float         weight;
```

HW6

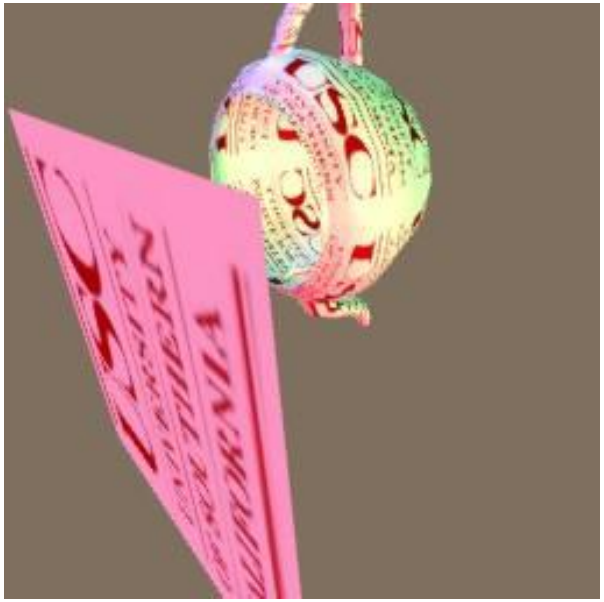
- Set new parameters (Xoffset, Yoffset, weight) in GzPutAttribute

- Apply offsets in GzPutTriangle

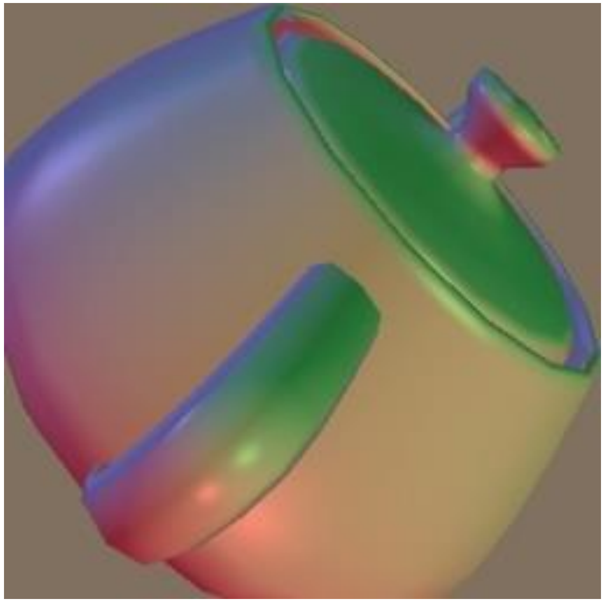
- Retrieve and compute a weighted sum of all 6 subimages (pixel by pixel)

HW6

- Example output #1



- Example output #2



Summary

“Application” part

- Set up multiple renderers
- Set up **offsets** for each renderer
- Rasterize with each renderer
- Compute weighted sum of all FBs

“Rend” part

- Modify GzPutAttribute to set up offsets
- Modify GzPutTriangle to apply offsets during rasterization

Q & A