

RenderMan

CMSC 435/634

Project 1

Use RenderMan/Pixie (pixie.sourceforge.com) to create a C program to model a simple scene

- ▶ Grading
 - ▶ Follow directions
 - ▶ Accuracy
 - ▶ Creativity/Effort

Pixie Environment

- ▶ Development Cycle
 - ▶ write C code
 - ▶ make, linking with libri → executable
 - ▶ run program → RIB
 - ▶ rndr → TIFF file
 - ▶ display imagefile

Minimal Program

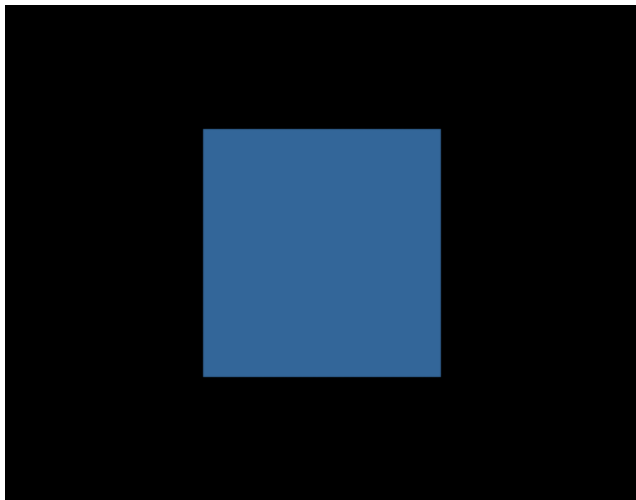
```
#include "ri.h"
RtPoint Square[4]={{.5,.5,.5},{.5,-.5,.5},
                  {-.5,.5,.5},{-.5,-.5,.5}};
static RtColor Color = {.2, .4, .6};
int main ()
{
    RiBegin("square.rib"); /* start the renderer */
    RiDisplay("square.tif", "file", "rgb", RI_NULL);
    RiWorldBegin();
        RiSurface("constant", RI_NULL);
        RiColor(Color); /* declare color */
        RiPatch(RI_BILINEAR, /* declare the square */
               RI_P, (RtPointer)Square, RI_NULL);
    RiWorldEnd();
    RiEnd();
    return 0;
}
```

Square RIB

```
##RenderMan RIB-Structure 1.0
##Creator 3Delight 2.1 (Jun 5 2004)
##CreationDate Tue Aug 31 18:32:38 2004
Display "square.tif" "file" "rgb"
```

```
WorldBegin # {
  Surface "constant"
  Color [ 0.2 0.4 0.6 ]
  Patch "bilinear"
  "P" [ 0.5 0.5 0.5 0.5 -0.5 0.5
       -0.5 0.5 0.5 -0.5 -0.5 0.5 ]
WorldEnd # }
```

Results



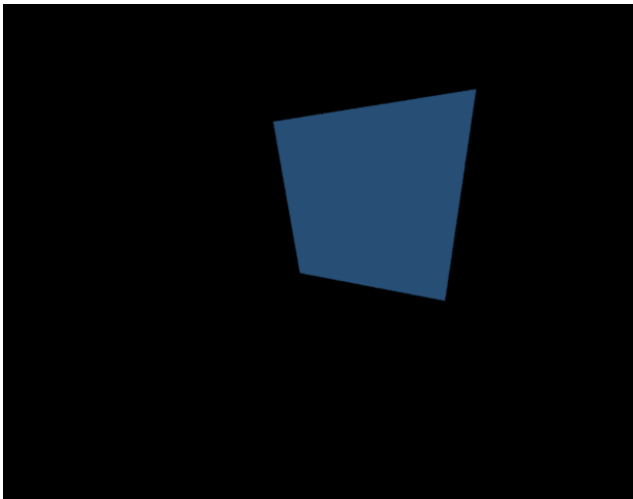
Refined Program (lights, perspective)

```
RiBegin("persp.rib"); /* start the renderer */
    RiDisplay("persp.tif", "file", "rgb", RI_NULL);

    RiLightSource("distantlight", RI_NULL);
    RiProjection("perspective", RI_NULL);
    RiTranslate(0.0, 0.0, 1.0);
    RiRotate(40.0, -1.0, 1.0, 0.0);

    RiWorldBegin();
        RiSurface("matte", RI_NULL);
        RiColor(Color); /* declare color */
        RiPatch(RI_BILINEAR, /* declare the square */
            RI_P, (RtPointer) Square, RI_NULL);
    RiWorldEnd();
RiEnd();
```

Results



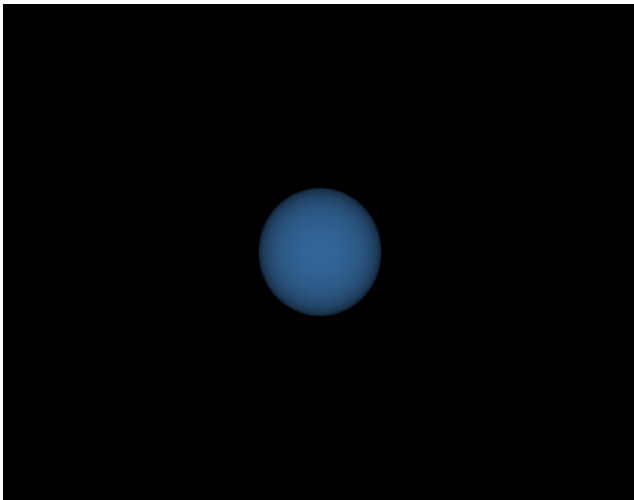
Alternate Primitive (Sphere)

```
RtFloat radius=1.0, zmin=-1.0, zmax=1.0,  
        thetamax=360;
```

...

```
RiWorldBegin ();  
    RiSurface("matte", RI_NULL);  
    RiColor(Color);    /* declare color */  
  
    RiSphere(radius, zmin, zmax,  
            thetamax, RI_NULL);  
  
RiWorldEnd ();
```

Results



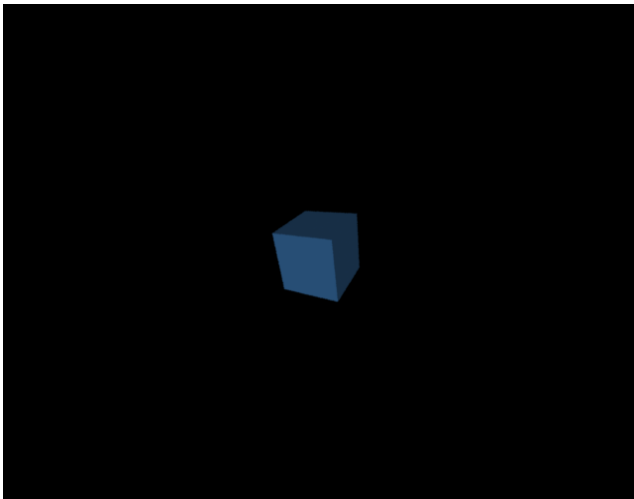
Basic Cube Program

```
RiWorldBegin ();  
  RiSurface ("matte", RI_NULL);  
  RiColor (Color);    /* declare color */  
  
  UnitCube ();  
  
RiWorldEnd ();
```

UnitCube Function

```
#define L -.5 // #define D -.5 // #define N -.5
#define R .5  // #define U .5   // #define F .5
UnitCube () {
    static RtPoint Cube[6][4] = {
        {{L,D,F}, {R,D,F}, {R,D,N}, {L,D,N}}, /*bottom*/
        {{L,D,F}, {L,U,F}, {L,U,N}, {L,D,N}}, /*left*/
        {{R,U,N}, {L,U,N}, {L,U,F}, {R,U,F}}, /*top*/
        {{R,U,N}, {R,U,F}, {R,D,F}, {R,D,N}}, /*right*/
        {{R,D,F}, {R,U,F}, {L,U,F}, {L,D,F}}, /*far*/
        {{L,U,N}, {R,U,N}, {R,D,N}, {L,D,N}}}; /*near*/
    int i;
    for (i = 0; i < 6; i++) {
        RiPolygon(4, RI_P, (RtPointer)Cube[i], RI_NULL);
    }
}
```

Results



Refined UnitCube

```
RtPoint Square[4] = { { .5, .5, .5 }, { .5, -.5, .5 },  
                    { -.5, .5, .5 }, { -.5, -.5, .5 } };  
  
UnitCube ()  
{  
    RiTransformBegin ();  
    RiPatch (RI_BILINEAR, RI_P, (RtPointer) Square,  
            RI_NULL);  
    RiRotate (90.0, 0.0, 1.0, 0.0); /*right face*/  
    RiPatch (RI_BILINEAR, RI_P, (RtPointer) Square,  
            RI_NULL);  
    RiRotate (90.0, 0.0, 1.0, 0.0); /*near face*/  
    RiPatch (RI_BILINEAR, RI_P, (RtPointer) Square,  
            RI_NULL);  
    RiRotate (90.0, 0.0, 1.0, 0.0); /*left face*/  
    RiPatch (RI_BILINEAR, RI_P, (RtPointer) Square,  
            RI_NULL);  
    RiTransformEnd ();  
}
```

Refined UnitCube (cont)

```
RiTransformBegin (); /*bottom face*/
    RiRotate(90.0, 1.0, 0.0, 0.0);
    RiPatch(RI_BILINEAR, RI_P, ( RtPointer ) Square ,
            RI_NULL );
RiTransformEnd ();
RiTransformBegin (); /*top face*/
    RiRotate(-90.0, 1.0, 0.0, 0.0);
    RiPatch(RI_BILINEAR, RI_P, ( RtPointer ) Square ,
            RI_NULL );
RiTransformEnd ();
}
```

Results

