CMSC 435/634

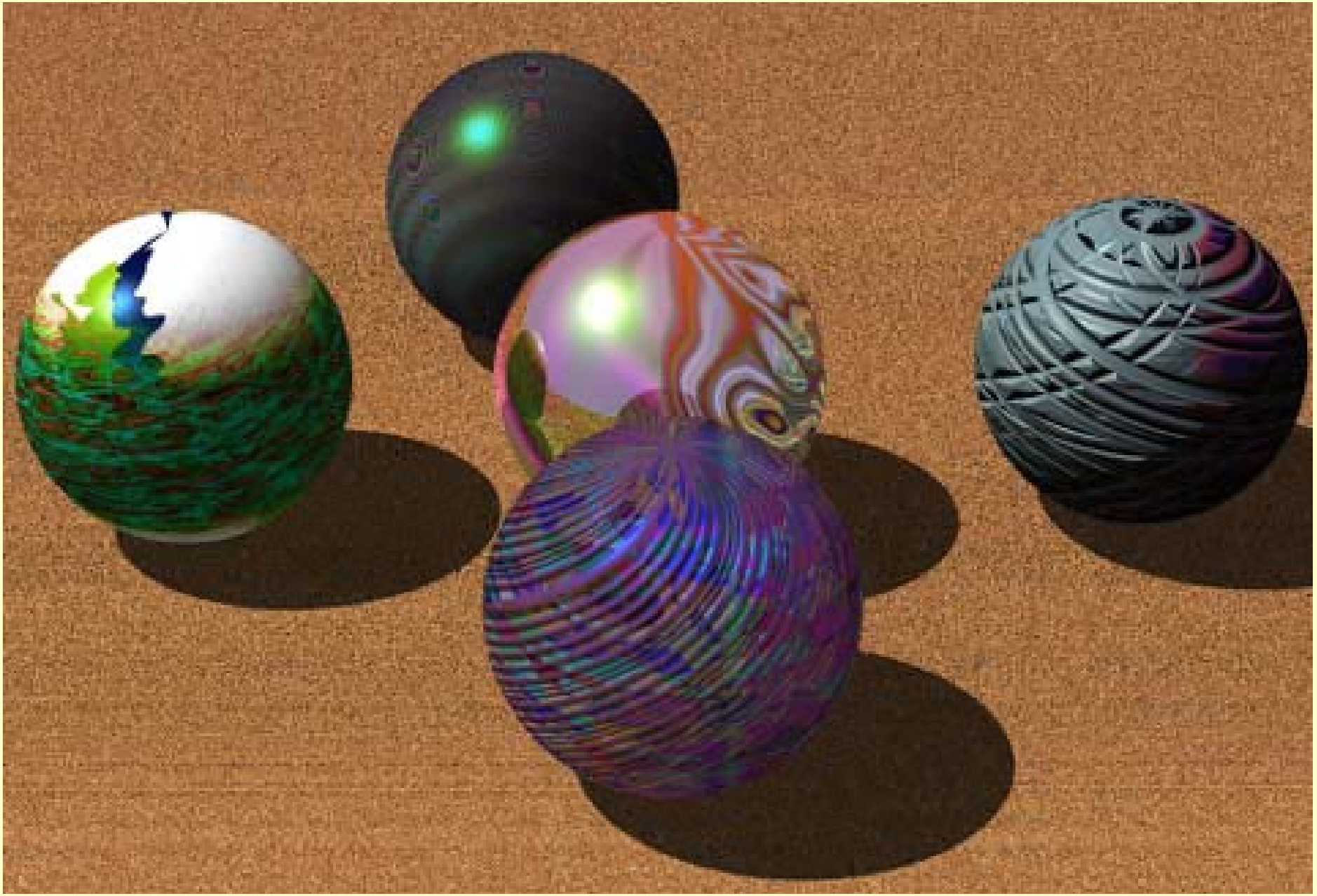
RenderMan

Project 1

Use Renderman/3Delight (www.3delight.com) to model and marbles in a bowl on top of a table

- Grading
  - follow directions
  - accuracy
  - creativity/effort

www.wildthingsbeads.com/images/misc/marbles.jpg



www.bryceworks.com/b1gall/marbles.html

www.squaregear.net/raytrace/marbles.jpg

3Delight Environment

Development Cycle

- write C code
- make, linking with lib3delight → executable
- run program → RIB
- renderdl → TIFF file
- display imagefile

Installed on gl (~olano/public/435/3delight)

- rlogin, DISPLAY, etc
- source ~olano/public/435/3delight/.3delight_csh
- . ~olano/public/435/3delight/.3delight_bash

Minimal Program

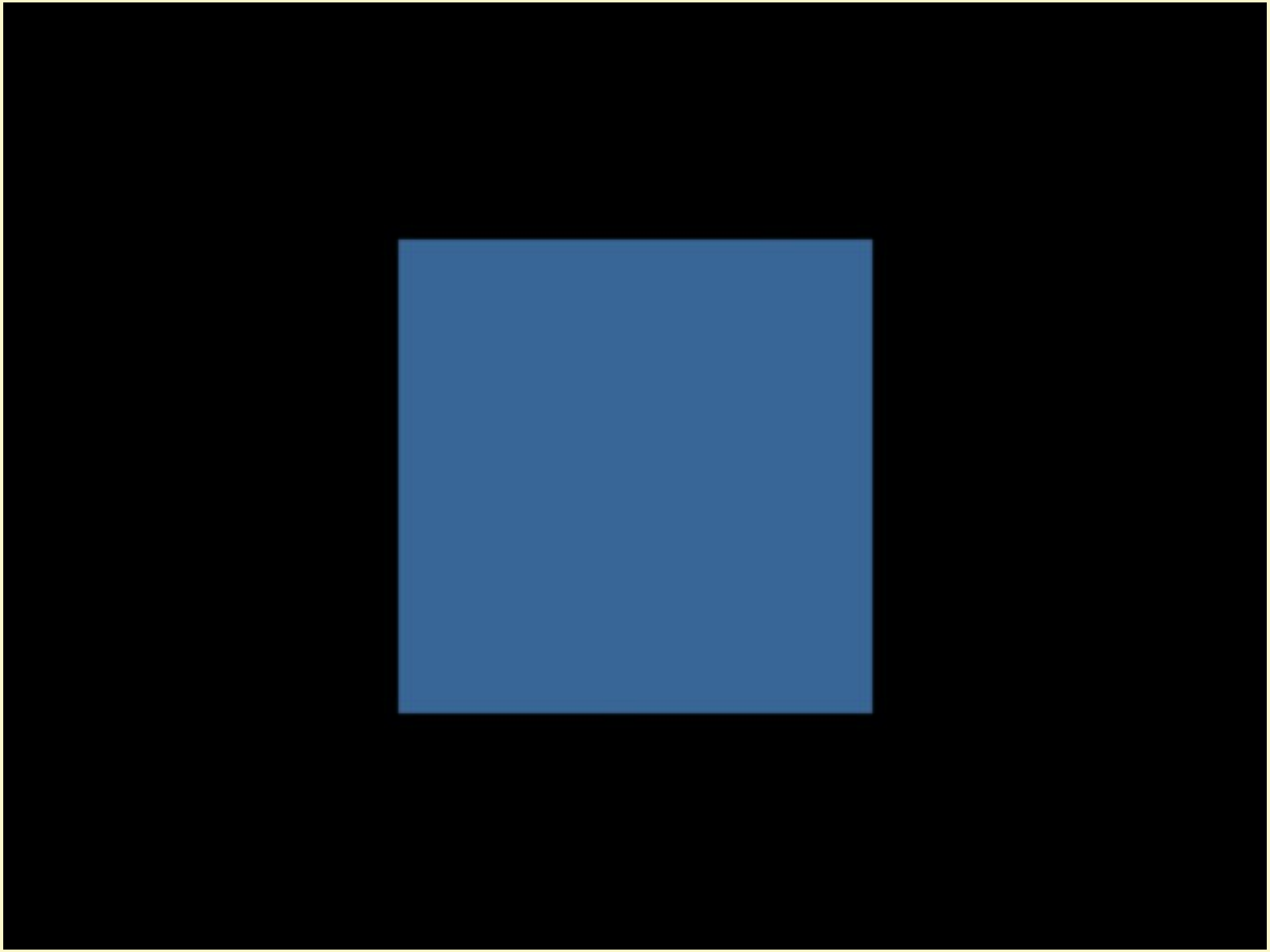
```
#include "ri.h"
RtPoint Square[4]={{.5,.5,.5},{.5,-.5,.5},
           {-.5,.5,.5},{-.5,-.5,.5}};
static RtColor Color = {.2, .4, .6};
int main ()
{
  RiBegin("square.rib"); /* start the renderer */
    RiDisplay("square.tif","file","rgb",RI_NULL);
    RiWorldBegin();
      RiSurface ("constant", 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 2.0 (Jan 29 2004)
##CreationDate Mon Feb  2 13:27:29 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  # }
```

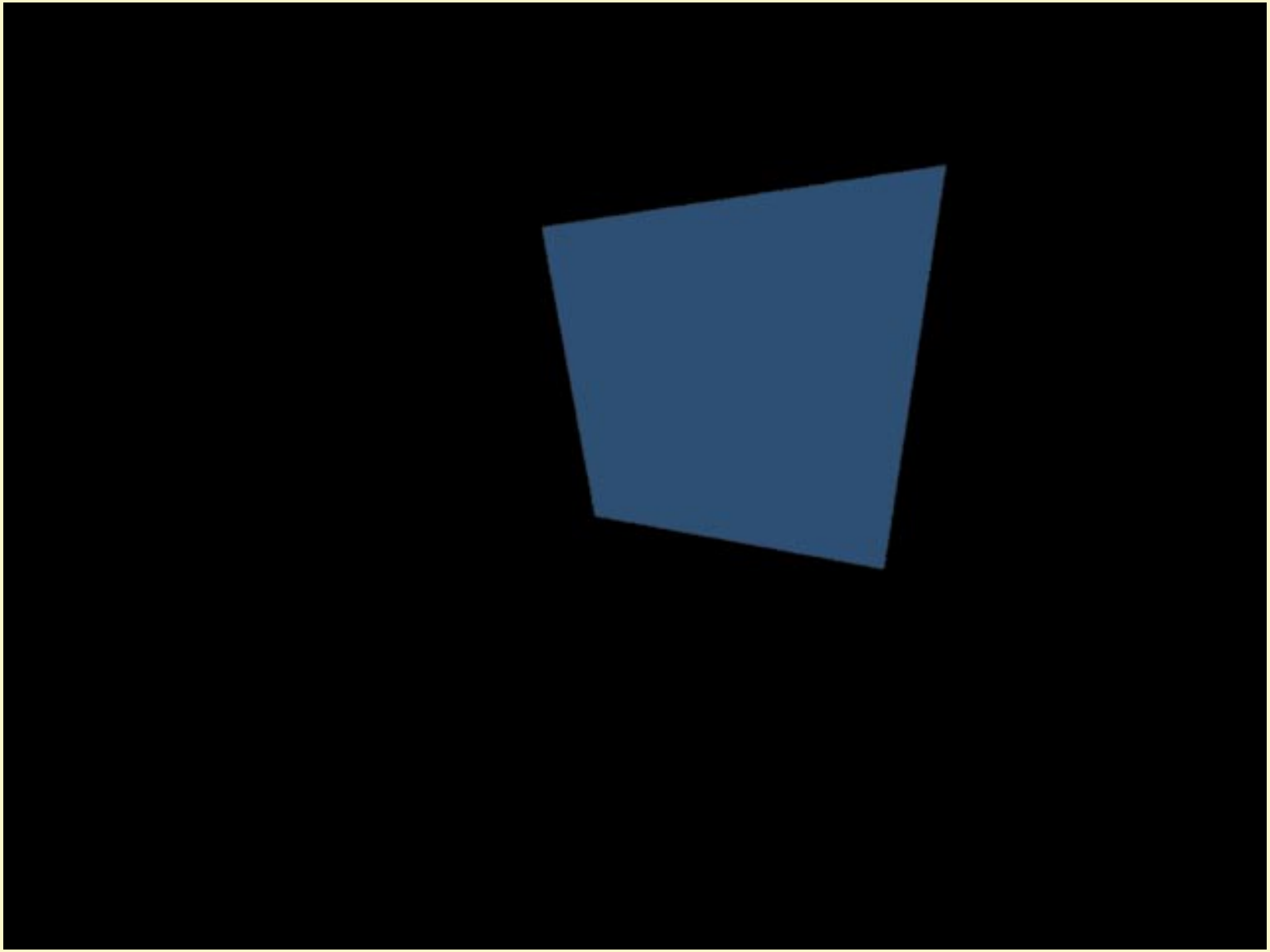
Square results



Refined Program (lights)

```
int main ()
{
  RiBegin("square.rib"); /* start the renderer */
    RiDisplay("square.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();
  return 0;
}
```

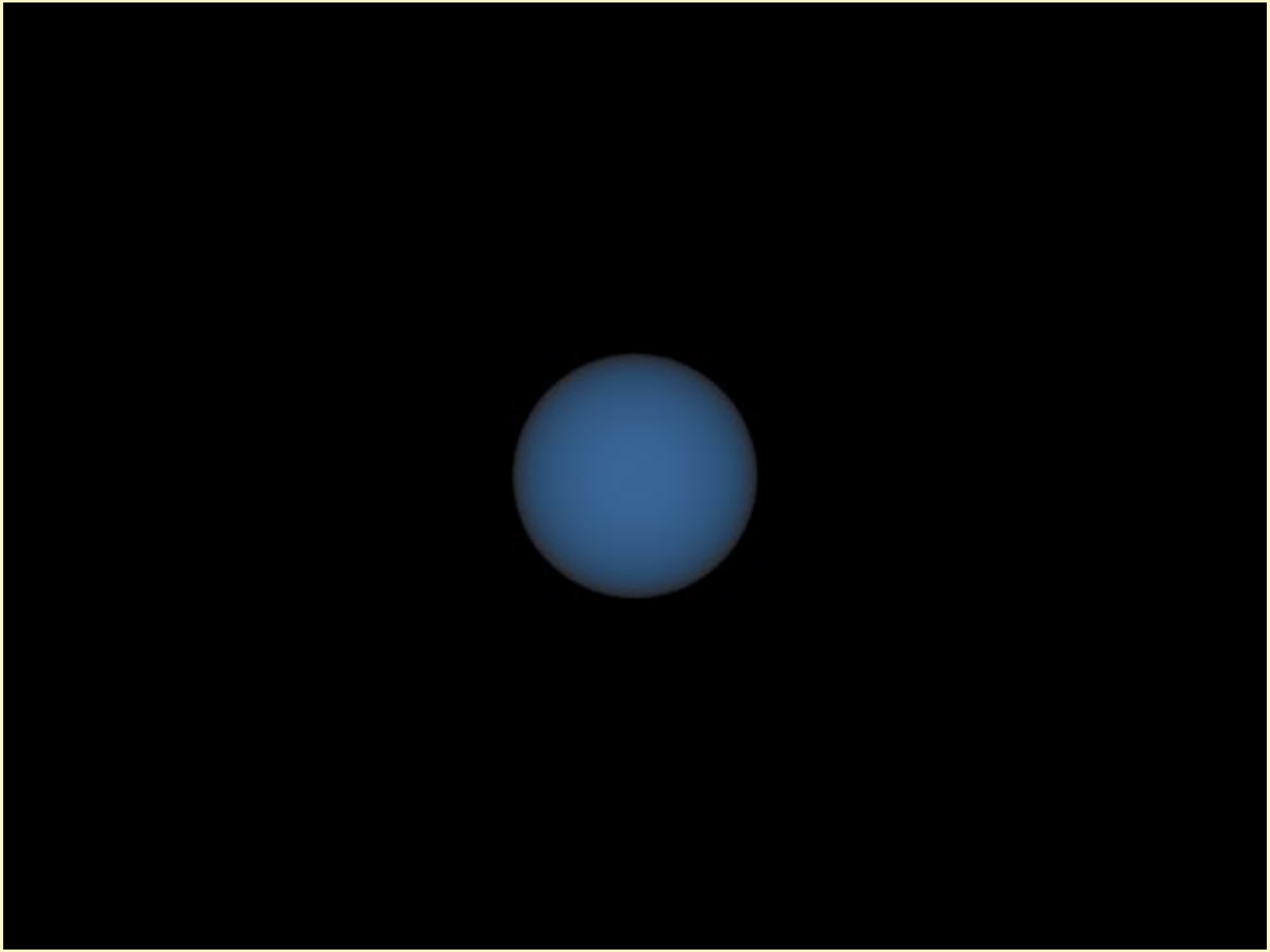
Results



Alternative Primitive (sphere)

```
RtFloat radius=1.0, zmin=-1.0, zmax=1.0, thetamax=360;
int main ()
{
  RiBegin("square.rib"); /* start the renderer */
    RiDisplay("square.tif","file","rgb",RI_NULL);
    RiLightSource ("distantlight", RI_NULL);
    RiProjection("perspective", RI_NULL);
    RiTranslate(0.0,0.0,4.0);
    RiRotate (40.0, -1.0, 1.0, 0.0);
    RiWorldBegin();
      RiSurface ("matte", RI_NULL);
      RiColor (Color);    /* declare color */
      RiSphere (radius, zmin, zmax, thetamax, RI_NULL);
    RiWorldEnd();
  RiEnd();
  return 0;
}
```

Results



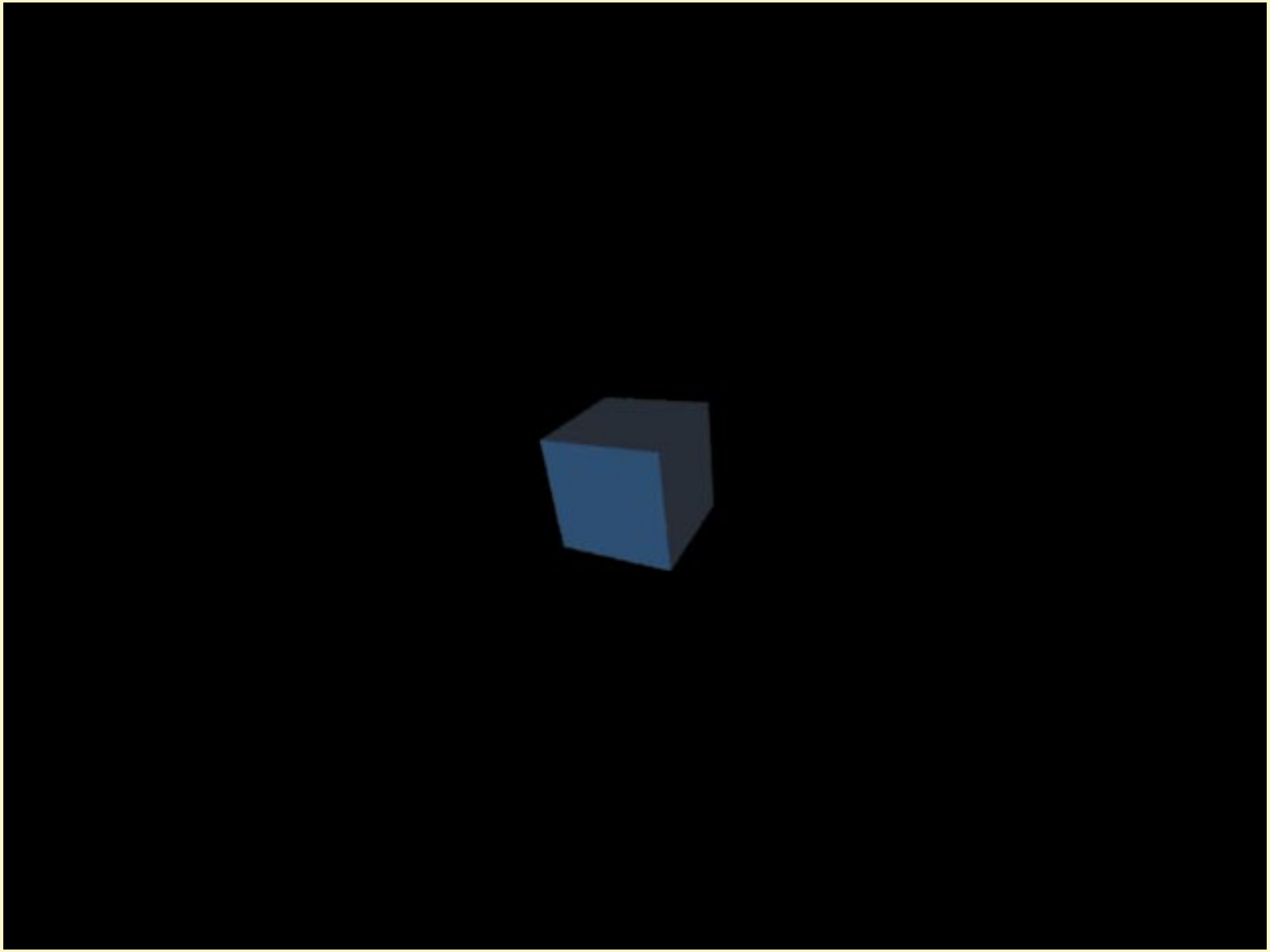
Basic cube program

```
main () {
  RiBegin(RI_NULL); /* start the renderer */
    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 */
      UnitCube();
    RiWorldEnd();
  RiEnd();
}
```

Cube Procedure

```
#define L -.5       #define R .5
#define D -.5       #define U .5
#define N -.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 Cube Procedure

```
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();
  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