# Computer Graphics – Exercise 2 – Ray tracing

## *Grades*

| | Grade | Notes |
|---|---|---|
| xxx635073_xxx100121 | 95 | |
| xxx866614_xxx711338 | 97 | |
| xxx327812_xxx322335 | 89 | |
| xxx770623_xxx830590 | 86 | |
| xxx657892_xxx458070 | 91 | |
| xxx159441_xxx756289 | 89 | |
| xxx387275_xxx811524 | 96 | |
| xxx452541 | 96 | |
| xxx165769_xxx677072 | 92 | |
| xxx355717 | 100 | |
| xxx697065_xxx109618 | 80 | |
| xxx854921_xxx530159 | 87 | |
| xxx570388 | 91 | Late (-10) |
| xxx115149_xxx534972 | 95 | |
| | 91.71 | |

## *Testing*

- On each submission I ran 6 tests (7 including bonus MCPT when available)
- Each test is a scene designed to test required features
  - Scene1 – Primitives, hard light, shadowing, Textures, Checkers, materials
  - Scene2 – Emission, attenuation and shininess
  - Scene3 – Reflection
  - Scene4 – Advanced lighting (monte or area)
  - Scene5 – Hemisphere-lighting, Super-sampling

    - Scene6 – Meshes, both flat and smooth
    - Scene7 - MCPT
- Test scenes can be found here:
    - http://www.cs.tau.ac.il/research/chen.goldberg/courses/cg10a/ex2/test.zip
- For bad results I either take away points for that specific scene or try to find if the mistakes also occurs in one of the reference scenes you received
- Mistakes that occur on the reference scenes are treated more severely (as it is up to you to verify your implementation on them before you submit)

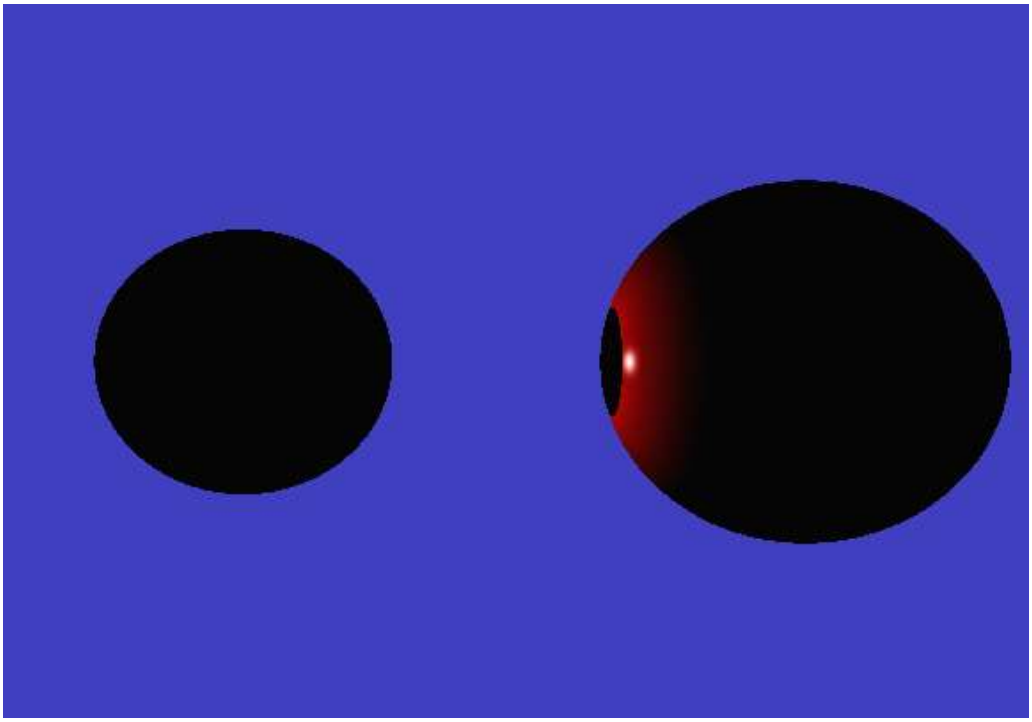
## *Points*

- Background (2 points)
    - Plain color background
    - Background image
- Control the camera and screen (5 points)
    - simple pinhole camera
- Display geometric primitives in space: (20 points)
    - rectangular planes
    - circular planes (discs)
    - spheres
    - boxes (cubes)
    - Cylinders (tubes)
    - Triangular meshes (ply2, off formats)
- Render Surfaces:
    - Simple materials (ambient, diffuse, specular...) (9 points)
        - According to the lighting equation studied in class.
    - Basic "checkers" pattern (7 points)
    - Image Textures (8 points)
- Basic lighting (13 points)
    - parallel directional light
    - omni-direction point light
- Basic hard (sharp) shadows (9 points)
- pixel super sampling (5 points)
- Reflection - reflecting surfaces. mirrors and shiny objects (8 points)
- Advanced Lighting
    - One of the following (7 points)
        - Area light simulated by multiple point lights
        - Monte-carlo Lighting
    - Hemispherical Lighting (7 points)

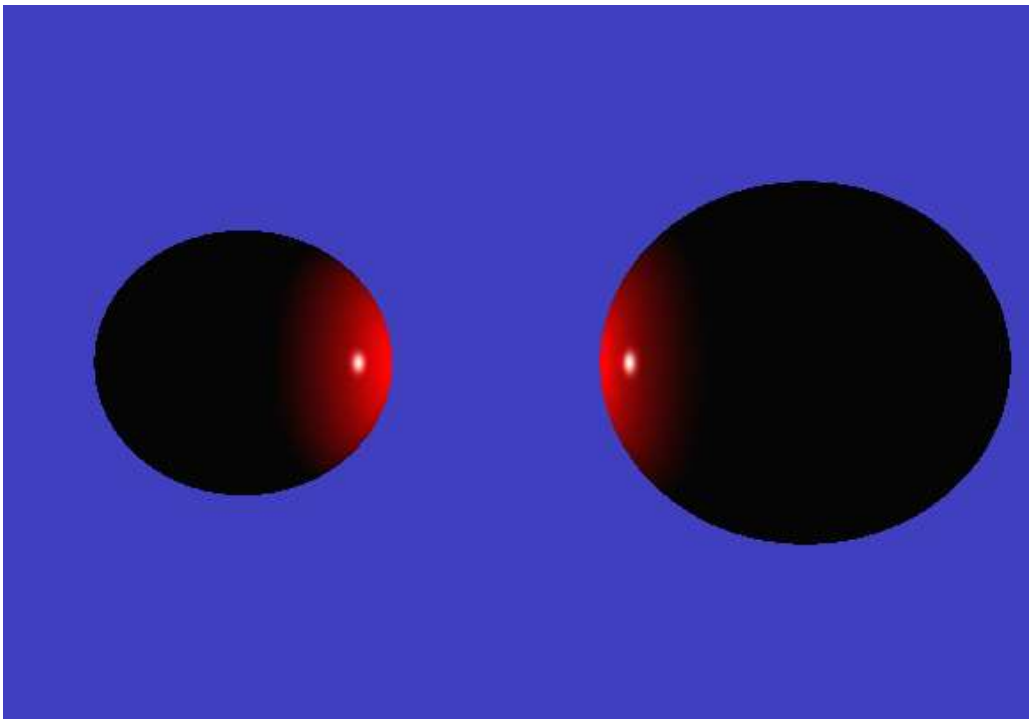
- Bonus
  - Acceleration (5)
    - Uniform grid speedup
    - Octree or BSP for speedup
  - Monte-carlo path tracing (5)

### *Notes*

- **Phantom Shadows** - there was one bug that occurred in 4 submissions. It manifested itself as false shadows scattered all over the scene. After some dwelling on the matter I reached a conclusion that this is caused by bad shadow calculation – by not considering what happens when there is an object AFTER the point light source. Evidently the 10 other submissions did take this into account, and thus the penalty for this bug is: -5 points.



*Example of phantom shadow bug - Light is positioned between the two sphere. Each sphrere is projected onto the other as shadow.*



*Without the bug*

- Almost everyone got cylinder inner shadowing wrong (light coming through the other side, or no self-shadowing), including our own reference images. Therefore I didn't take any notice of it. Two students did implement it correctly and we decided to give them a small bonus for that.
- Some were confused by the inconsistencies between the pdf and the online versions of the exercise. I tried to take this into account even though we clearly stated that the pdf is outdated. Furthermore, regardless of what version of the exercise you read – your implementation should always be able to reproduce the reference scenes!
- From what I saw nobody implemented the aforementioned acceleration methods.

## *Submissions*

### xxx635073_xxx100121

- Test2
  - Emission blends with texture
  - Cylinder not self occluding from the outside (-2)
    - Light comes through the cylinder and shows on the other outer-side (not to be confused with the inner-side discussed in the notes)
- Test5
  - Hemisphere lighting causes specular?
  - Supersampling doesn't include background (-2)

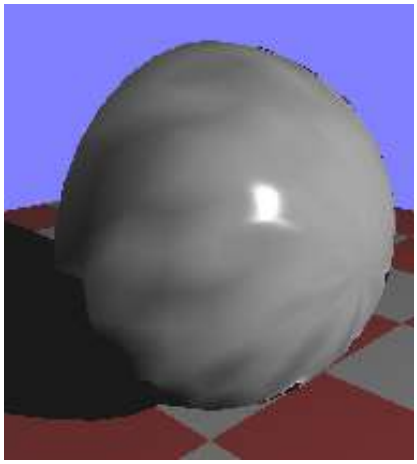
- Test6
  - Phong shaded mesh is a bit off, introduces noticeable artifacts (-1)
- Elegant code and structure, very nice

## xxx866614_xxx711338

- Cylinder parameterization fails when direction is (-1 0 0) (-2)
- Test2
  - Attenuation calculation is inaccurate (-2)
- MCPT – doesn't seem to work well. Why require reflection?
- Code is a bit too dense
- Both scenes are nice (+1)

## xxx327812_xxx322335

- Doesn't take files from scene directory (-1)
- Test2
  - Phantom shadows (-5)
- Test5
  - Supersampling doesn't include background (-2)
- Test6
  - Phong shading meshes is pretty bad (-3)
    - You must have noticed what it does to

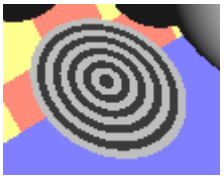


*ref5/mesh_canopy_phong.txt*

- Only supplied a single scene

## xxx770623_xxx830590

- Slower than most (especially with multiple lights)
- Test1
  - Spheres appear blank (-5)
    - They have no texture or checkers!
    - Closer inspection revealed that this bug is caused due the parameterization being effected by the position of the sphere!
  - Disc and cylinder parameterization are wrong (-4)



*"Checkers" pattern*

- Test2
  - Specular shininess is a bit off
- Test5
  - Hemisphere lighting doesn't seem to accept color-ground (-2)
- Test6
  - Phong shading meshes is conceptually wrong... (-3)
    - Appears like flat shading only in triangles...
- Amazingly well organized code and structure
- Cute scenes

## xxx657892_xxx458070

- Test2
  - Phantom shadows (-5)
- Test4
  - Not only strange shadows, but it seems that light spots are all over the place (-3)
    - On training examples also a little off
- Test6
  - No support for flat shading (-2)

        - Even ref5/mesh_canopy_flat.txt renders non-flat
- Nice scene

## xxx159441_xxx756289

- Phantom shadows (-5)
- Test3
    - Cylinder reflection (with texture?) is faulty (-3)
- Test5
    - Supersampling doesn't include background (-2)
- Test6
    - Renders only flat shaded models (-3)
        - ref5/mesh_canopy_phong.txt renders flat!
- Very nice scenes (+2)

## xxx387275_xxx811524

- Not only unable to obtain resources from the script directory, but requires them to be placed in special directories! (-1)
- Disk checkers size is twice as big
- Test1
    - Bug rendering a box – fills the entire screens (-2)
        - Probably happens in the case where the camera is positioned on one of the faces of the box
        - A slight move of the camera fixes the bug
- Test2
    - Emission doesn't work (-3)
- Code is too dense
- Very nice scenes (+2)

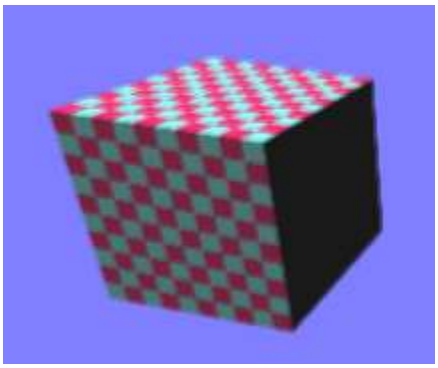
## xxx452541

- Considerably much slower than others
- Camera position and projection calculation is very faulty... (-5)
- Doesn't take files from scene directory (-1)
- MCPT

    - Since it is so slow then I find it hard to evaluate on any real scene (actually ran for hours before I killed it)
    - A small model with modest parameters gave a poor and noisy result (as expected)
    - However that was the most I was able to verify, thus I decided to give you half the bonus (+2)
- Mesh Phong a bit off
- Scenes nice but unoriginal

### xxx165769_xxx677072

- Disk parameterization is wrong – planar (-3)
- Cylinder parameterization fails when direction is (-1 0 0) (-2)
- Supersampling introduces noticeable blurriness (-3)



*A box with super-samp-width=4*

- Very nice model. Lighting a bit funny though.

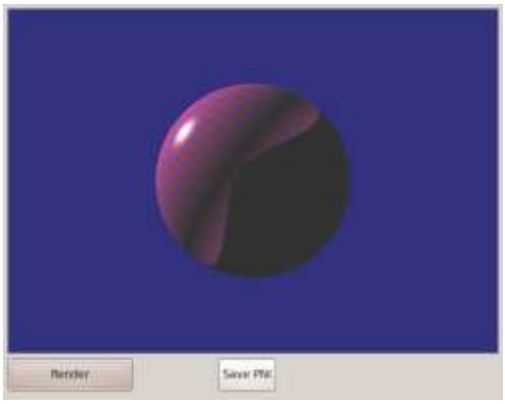
### xxx355717

- Cylinder inner-shadowing works (+1)
- Test2
    - Emission Doesn't work (-3)
- Fast mesh rendering
- Very cute scene. Simple but aesthetic (+2)
- Good job!

### xxx697065_xxx109618

- Texture doesn't blend with light model! Textured objects aren't shaded (-4)
- Phantom shadows (-5)
- Checkers size on sphere a inconsistent with examples

- Bug when computing diffuse of back-facing normals (-3)

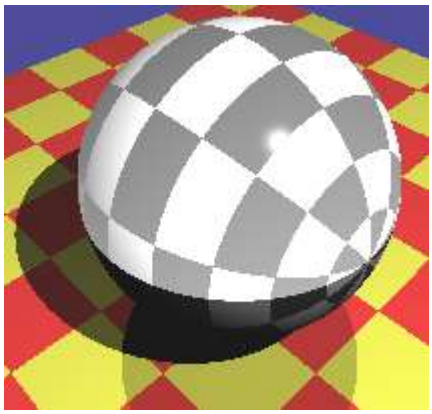


*usphere_check_4 - different angle*

- Test1
  - Crashed when rendered a box! (-5)
    - Exception in thread "main" java.lang.NullPointerException
      - at ParseableObject.getColor(ParseableObject.java:62)
      - at ParseableObject.getColor(ParseableObject.java:22)
- Where is phong shading? (-3)
    - I see that the program doesn't recognize the shader parameter...
    - ref/mesh_canopy_phong.txt – outputs rubbish...
- Impressive models, but don't quite make a complete scene.

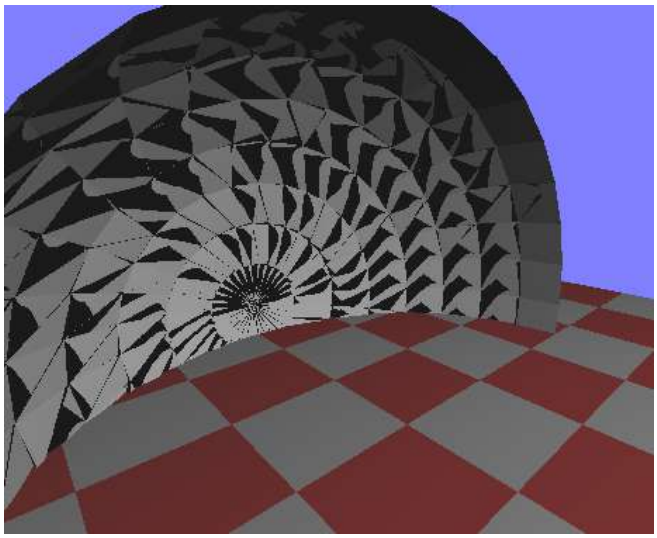
## xxx854921_xxx530159

- Hemisphere lighting messes up everything!
  - Not sure exactly what happens, but everything is reflecting and false shadows appear.
  - What I don't see is Hemisphere lighting
  - The hemisphere lighting code seem to be somehow in the right direction, but since you fail to render the reference examples I must reduce all points (-7)



*ref5/hemisphere_01.txt*

- Supersampling doesn't include background (-2)
- Renders meshes wrong (-4)
  - Probably computes their geometry direction differently
  - Anyway unable to reproduce ref5/mesh_canopy_flat.txt and ref5/mesh_canopy_phong.txt



*ref5/mesh_canopy_phong.txt*

- Test7
  - Too slow for all scene – left only green cylinder and floor
  - Floor appears all white, no checkers. Probably bug with MCPT
  - No color bleeding seen...

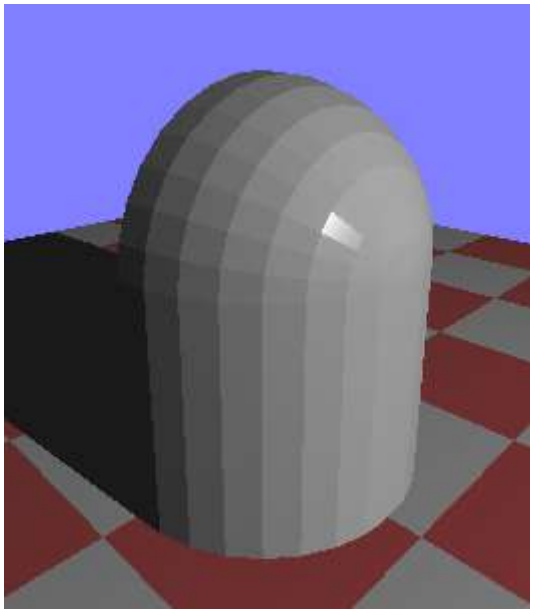
## xxx570388

- Cylinder inner-shadowing works (+1)
- FAST!

- Cute scene
- Best implementation – again!

### xxx115149_xxx534972

- Doesn't take files from scene directory (-1)
- Test5
  - Requires Shader attribute to be a number, which is cute because it's obviously a typo in the exercise doc.
  - Then doesn't seem to use that number for anything.
  - Didn't see any evidence to smooth shading, and even flat shading (default) is distorted (-4)



*ref5/mesh_canopy_flat.txt*

- All the source files in one big folder.
- Powerful scene