

Computer Graphics

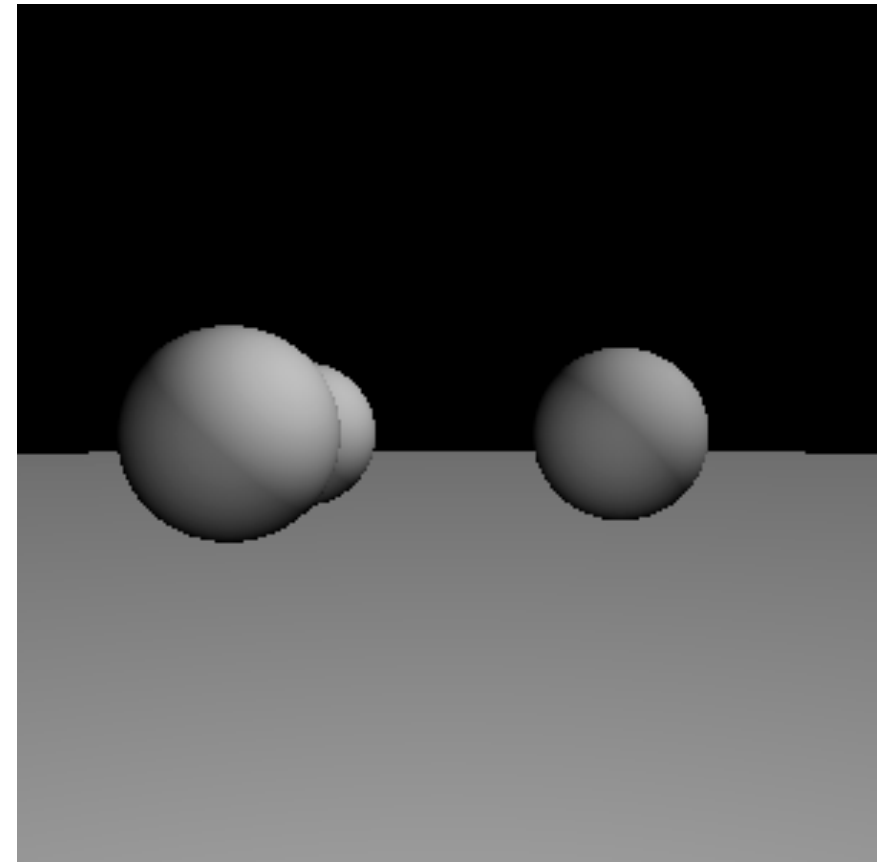
Lecture 10A
Mirror Reflection

Goals

- Be prepared to implement **mirror-reflective surfaces** in the ray tracing framework.

Diffuse Reflection

- Quite physically accurate for Lambertian surfaces
- Many surfaces are (close to) Lambertian
- Many others aren't!



Let's talk shinies.



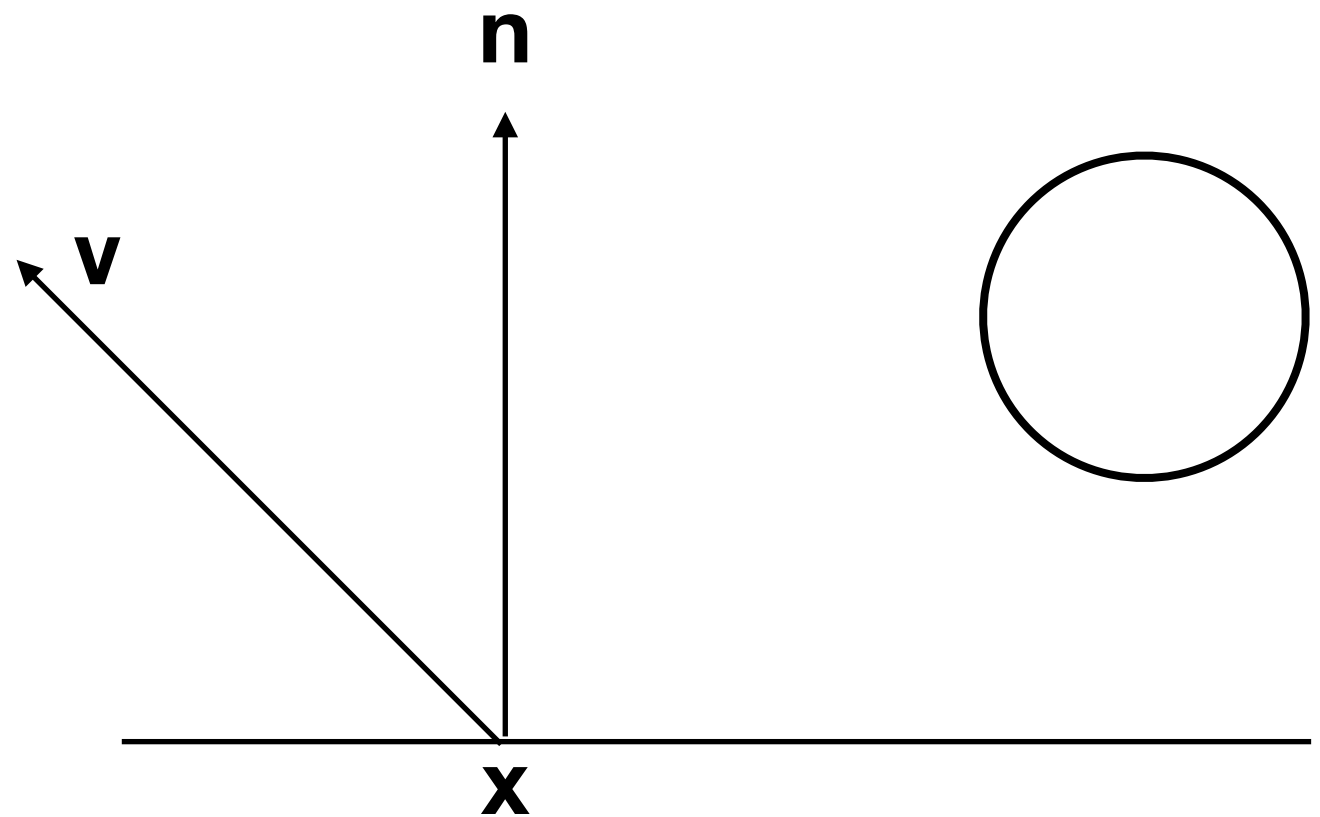
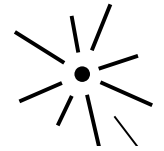
Let's talk shinies.

How does a mirror interact with light?



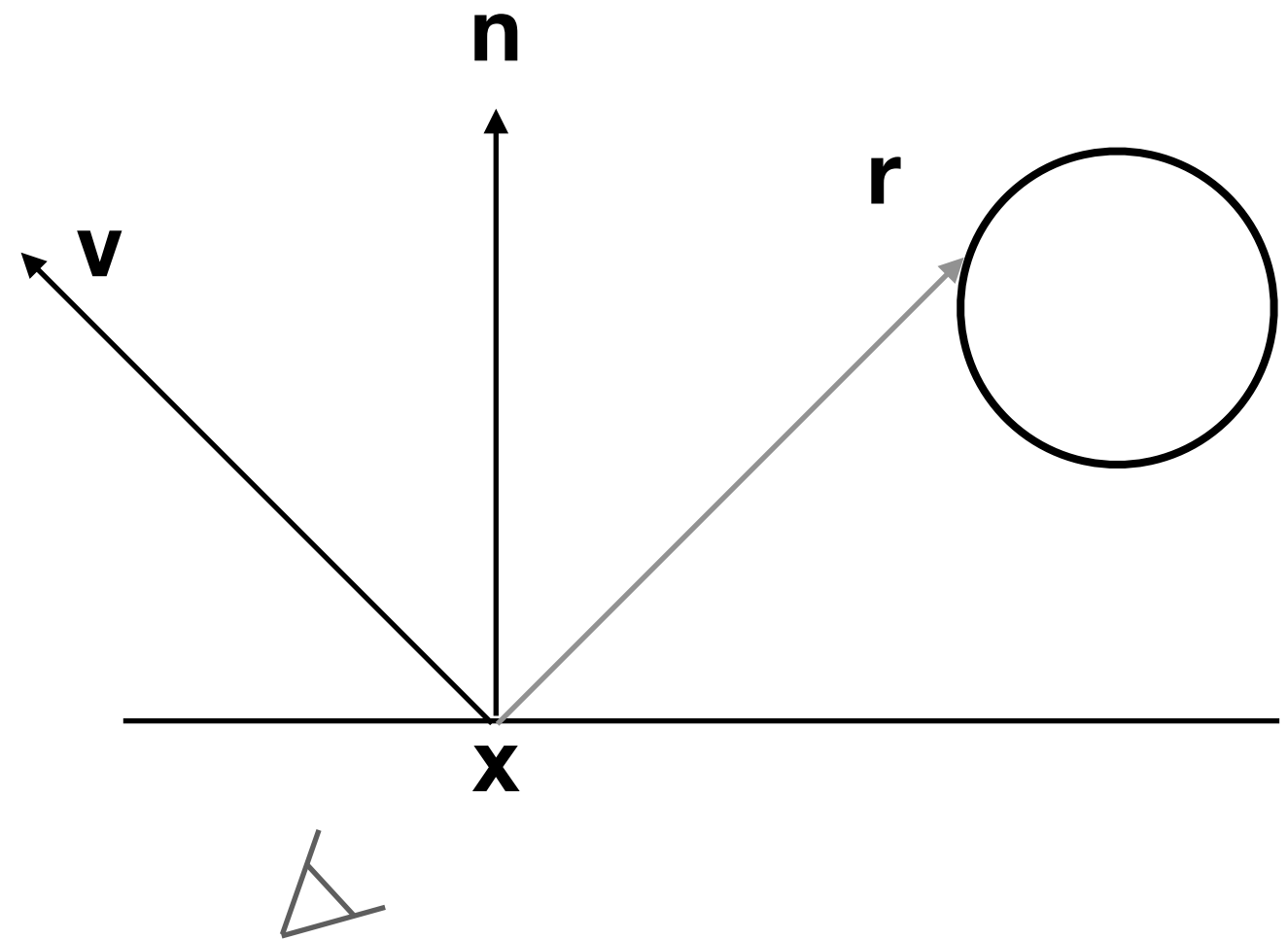
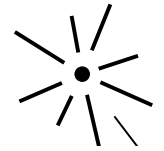
Mirror Reflection

What does a camera see when it looks at a mirror?



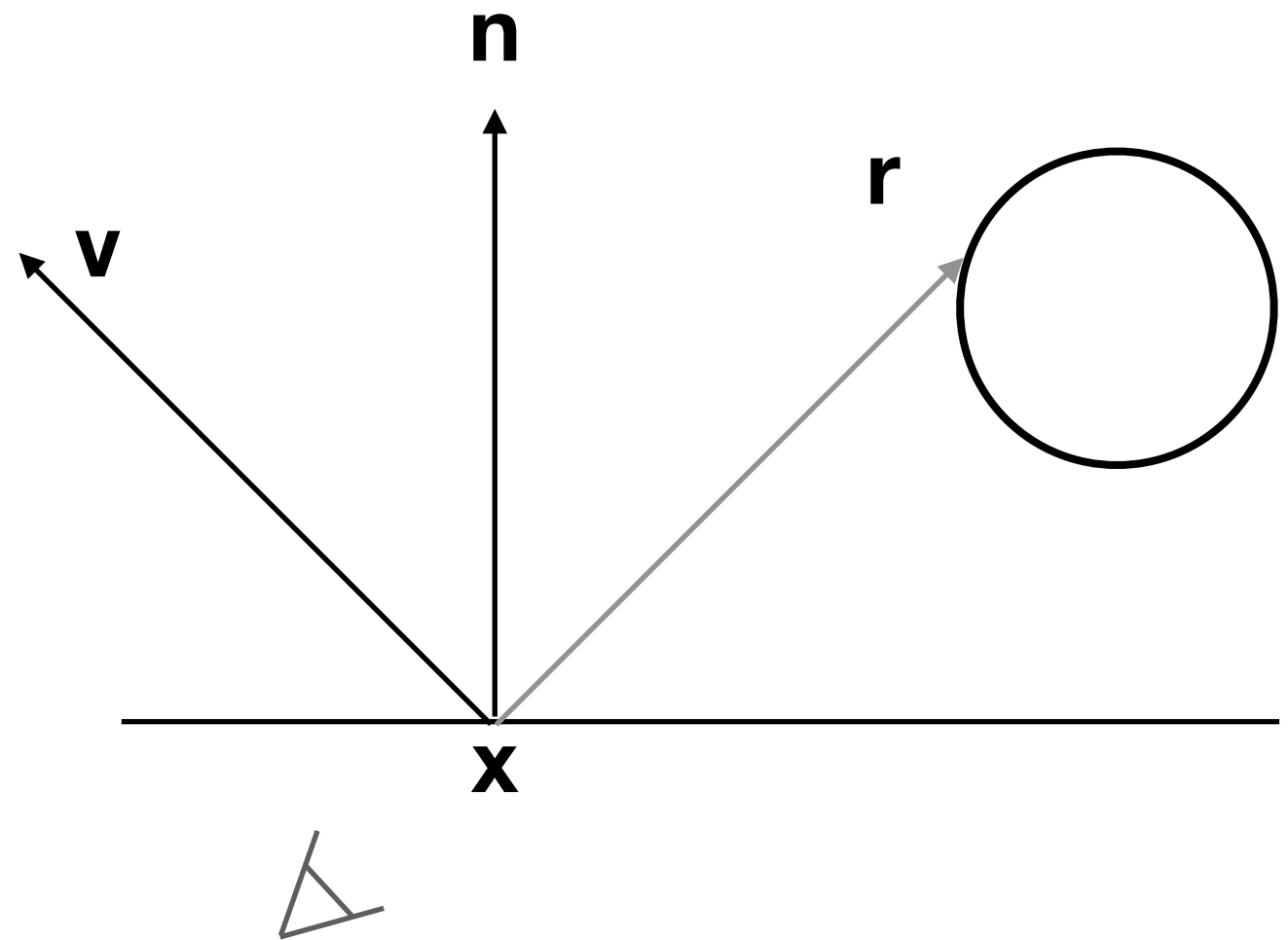
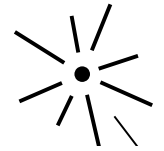
Mirror Reflection

What does a camera see when it looks at a mirror?



Mirror Reflection

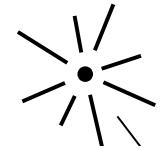
What does a camera see when it looks at a mirror?



Can we do this using the tools we already have?

Mirror Reflection

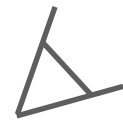
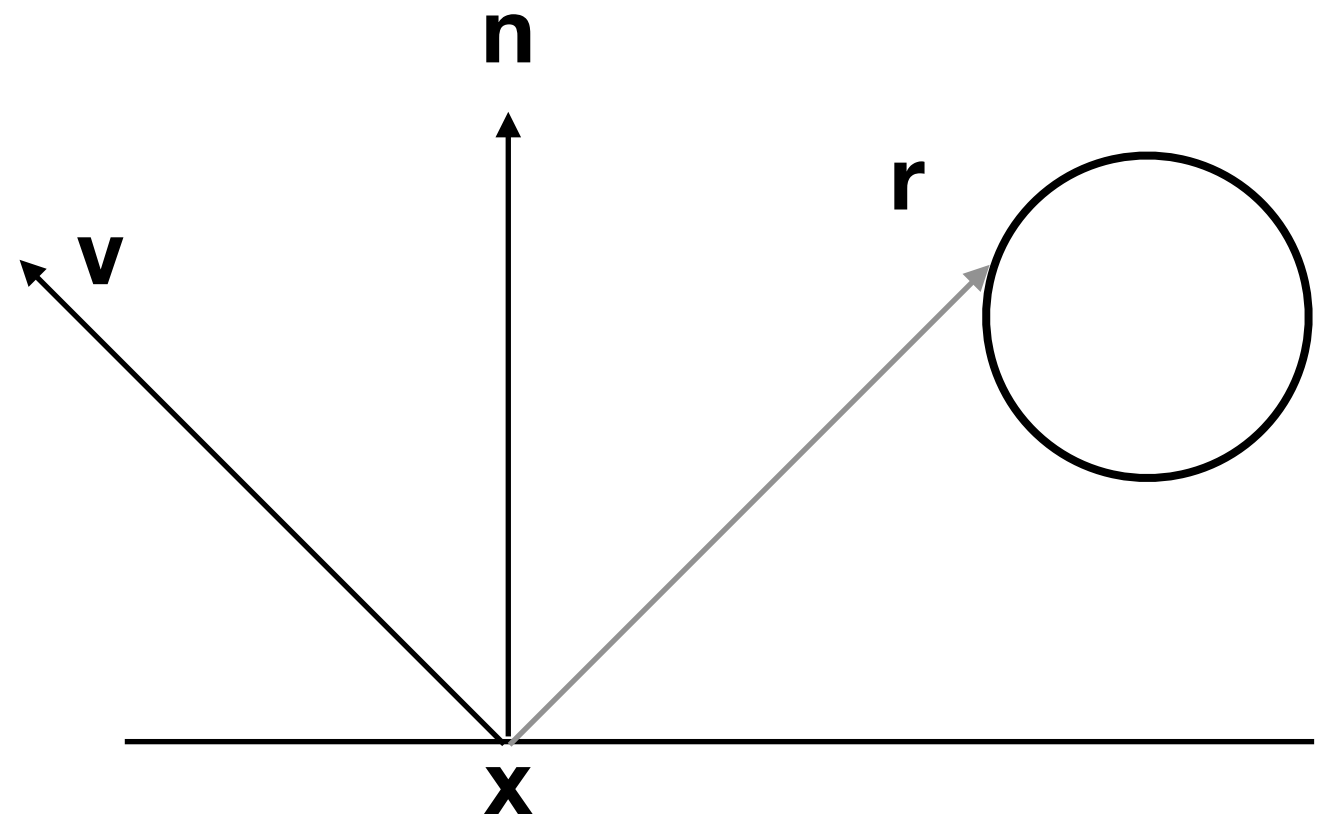
What does a camera see when it looks at a mirror?



Calculate \vec{r} :

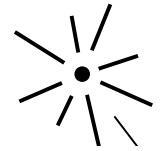
$$\vec{r} = -\vec{v} + 2(\vec{v} \cdot \vec{n})\vec{n}$$

```
mirr_ray.origin = x  
mirr_ray.direction = r
```



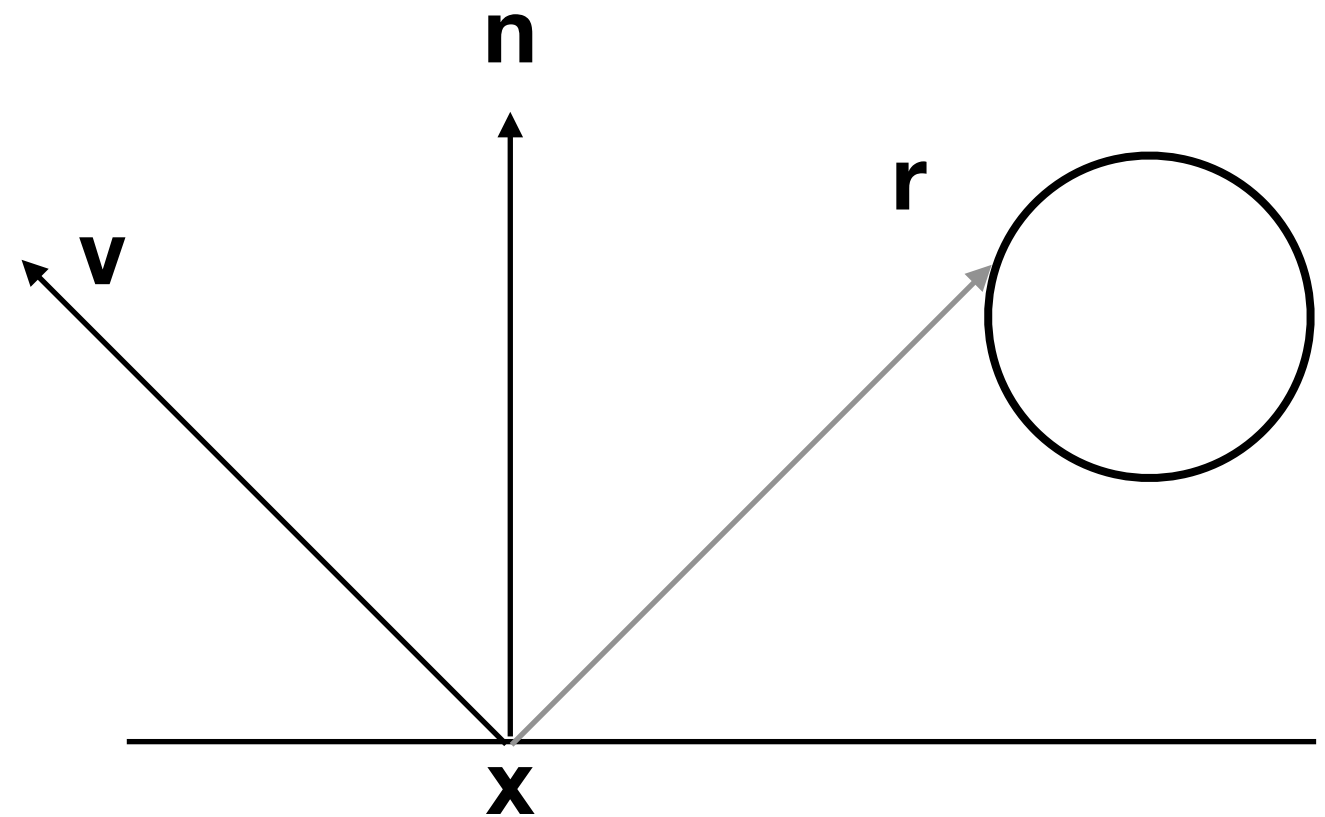
Mirror Reflection

What does a camera see when it looks at a mirror?



Calculate \vec{r} :

$$\vec{r} = -\vec{v} + 2(\vec{v} \cdot \vec{n})\vec{n}$$



```
mirr_ray.origin = x
```

```
mirr_ray.direction = r
```

```
color = traceray(scene, mirr_ray):
```

Mirror case in traceray

```
function traceray(ray, scene):  
    t, rec = find_intersection(ray, scene)  
    if rec.obj is a mirror:  
        compute r, the reflection direction  
        mirror_ray = Ray(rec.x, r)  
        return traceray(mirror_ray, scene)  
    # other cases, ...
```