

Name:	Date:	Period:
-------	-------	---------

## Lab11: Ray Casting

- Attach a code printout.
- Render a 3-D scene using pixel-by-pixel ray casting. The scene must contain at least three objects, at least one planar boundary (e.g., a floor), and at least one point source of light. The position of spheres given as a worked example is posted on the website and you may use this data exactly if you like.
- For each pixel...
  - Calculate the unit vector from the eye passing through the center of the pixel.
  - Determine if/where this ray intersects each object in the scene.
  - If not then determine if/where this ray intersects the floor/walls.
  - If not then color this pixel with the background color.
  - If it hits any object or the floor...
    - \* Add in color for ambient light (e.g., a factor of forty percent).
    - \* Calculate the unit normal vector at the point of intersection.
    - \* Push the point of intersection slightly off the surface by adding a small amount in the direction of the unit normal.
    - \* Cast another ray from the point of intersection to the light source.
    - \* Calculate the dot product of this (unit) ray and the normal.
    - \* If the dot product is negative then the object itself blocks the light source.
    - \* If not then determine if this ray intersects any other object (i.e., if we'll be in shadow) between the point of intersection and the light source.
    - \* If not then add in color (e.g., ninety percent) and multiply by the previously calculated dot product and an attenuation factor for the length of the ray.

---



---

### Official Use Only

Header:	Name	Correct Date	Program Description
Style:	Comments	Variable Names	Modular
Data Structures:	Obvious	General	Lean
Algorithm:	Clear	Correct	Efficient
Scoring:	Raw _____	Late _____	Total _____