When Do We Do the Shading Calculation?

Shading can be done at different stages of the rendering process...what we call the process depends on when the shading calculation occurs. Match the correct label to each of these three algorithms

```
For each light:
  For each object affected by the light:
      framebuffer += object * light
```

- (a) Multi-pass Lighting
- (b) Single Pass Lighting
- (c) Deferred Shading
- (d) Deferred Lighting

page 4-6 of deferred shading

```
For each object:
  Render to multiple targets

For each light:
  Apply light as a 2D postprocess
```

- (a) Single Pass Lighting
- (b) Multi-pass Lighting
- (c) Deferred Shading
- (d) Deferred Lighting

```
For each object:
  Render mesh, applying all lights in one shader
```

- (a) Deferred Lighting
- (b) Single Pass Lighting
- (c) Multi-pass Lighting
- (d) Deferred Shading

Deferred Complexity

Suppose we render a scene with n_L lights and n_T polygons. Which function best describes the algorithmic complexity of rendering when using *deferred shading*.

- ○ (a) $O(n_L^{n_T})$
- ○ (b) $O(n_T^{n_L})$
- ○ (c) $O(n_L + n_T)$
- ○ (d) $O(n_L \times n_T)$

Save & Grade | Save only | New variant

page 6
deferred
Shading

Deferred Shading Drawbacks

Which of the following effects are non-performant when implemented with deferred shading?

- [ ] (a) The Blinn-Phong reflection model
- [ ] (b) Transparency
- [ ] (c) Multi-Sample Anti-Aliasing (MSAA)
- [ ] (d) A full-scene directional light source
- [ ] (e) Ambient Occlusion

Select all possible options that apply.

Save & Grade | Save only | New variant

Page 36
Deferred Shading

Light Volumes

When using *deferred shading* one opimization is to render a *light volume* mesh enclosing the world space that the light hits, and only shade the pixels that within that light volume. Assuming the light volume is convex and does not intersect the far clip plane, which of the following is true?

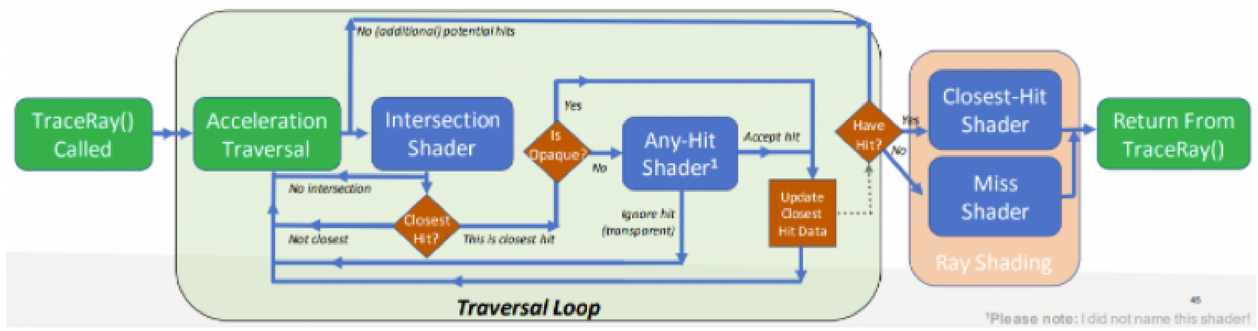
- (a) Backfacing polygons should be rendered.
- (b) None of the above are true.
- (c) Backfacing polygons should be culled.
- (d) The light volume is not large enough.

Save & Grade | Save only | New variant

page 28-30 deferred Shading

GPU Ray Tracing Shaders

There are 5 DirectX HLSL ray tracing shaders, please answer the following multiple choice questions.



We have a camera, viewport, and a scene of objects in an acceleration structure. What shader do we use to define our primary rays?

- ○ (a) Closest-hit Shader
- ○ (b) Intersection Shader
- ○ (c) Miss Shader
- ○ (d) Any-hit Shader
- ○ (e) Ray Generation Shader

For a ray, the entire acceleration structure has been traversed and **no** intersections have been found. What shader is called next?

- ○ (a) Intersection Shader
- ○ (b) Miss Shader
- ○ (c) Ray Generation Shader
- ○ (d) Closest-hit Shader
- ○ (e) Any-hit Shader

When traversing through the acceleration structure, what shader is used with each hittable sub-structure first?

- ○ (a) Any-hit Shader
- ○ (b) Intersection Shader
- ○ (c) Ray Generation Shader
- ○ (d) Closest-hit Shader
- ○ (e) Miss Shader

An intersection for a ray is found but the surface has some level of transparency. What shader is called next?

- ○ (a) Any-hit Shader
- ○ (b) Miss Shader
- ○ (c) Ray Generation Shader
- ○ (d) Closest-hit Shader
- ○ (e) Intersection Shader

For a ray, the entire acceleration structure has been traversed and **a few** intersections have been found. What shader is called next?

- ○ (a) Intersection Shader
- ○ (b) Closest-hit Shader
- ○ (c) Any-hit Shader
- ○ (d) Ray Generation Shader
- ○ (e) Miss Shader

Structure of GPU shaders lecture

HLSL Shader Code

What visual effect does the following HLSL code implement?

```
[shader("closesthit")]
void IndirectClosestHit(inout IndirectPayload pay,
                        BuiltinTriangleIntersectAttribs attribs) {
    ShadingData hit = getHitShadingData( attribs );
    float3 directLight = DiffuseShade( hit.pos, hit.norm, hit.difColor );

    float3 bounceDir = selectRandomDirection();
    float3 indirectColor = shootColorRay( hit.pos, bouncDir );
    float3 indirectLight = DiffuseIndirect( bounceDir, indirectColor );

    pay.color = directLight + indirectLight;
}
```

- (a) Total Internal Reflection
- (b) Shadows
- (c) Global Illumination
- (d) Transparency
- (e) Ambient Occlusion

Save & Grade | Save only | New variant

Multi-bounce lighting lecture

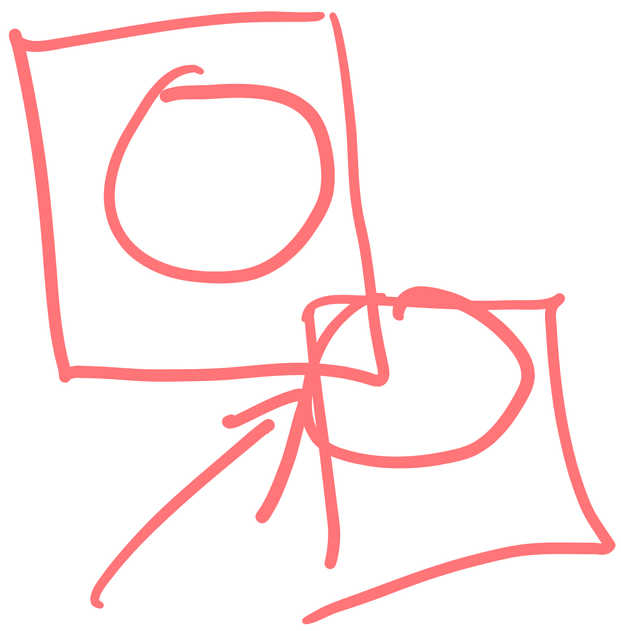
BVH Facts

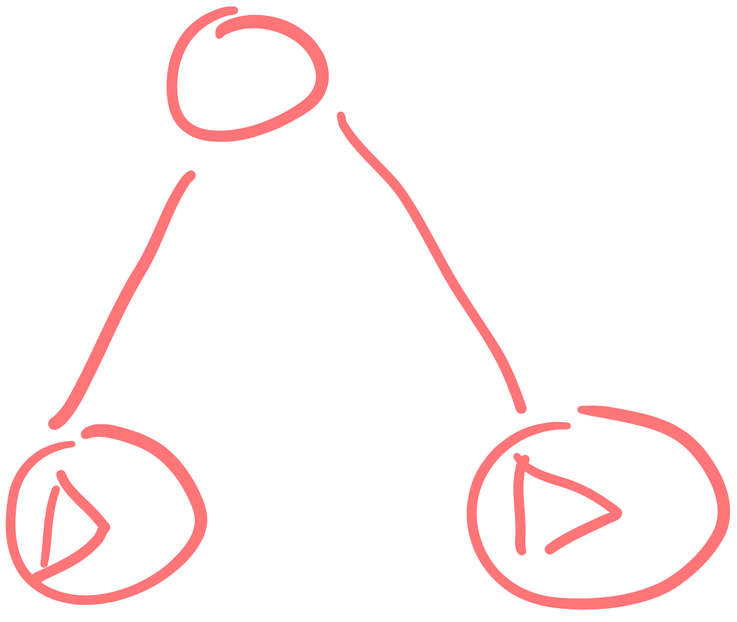
Which of the following are true of a binary bounding volume hierarchy based on axis-aligned bounding boxes (AABBs)?

- [ ] (a) A BVH will result in faster ray intesection tests than a kd-tree.
- [x] (b) Suppose a node has 2 child subtrees L and R. It is possible that a traversal in ray tracing will need to descend both subtrees.
- [ ] (c) Splitting n objects into two groups using the *median heuristic* requires $\Theta(n \log n)$ time.
- [x] (d) If we use the constraint that a leaf contains only one geometric primitive, a BVH using a binary tree structure containing n primitives will have $n - 1$ internal nodes.

Select all possible options that apply.

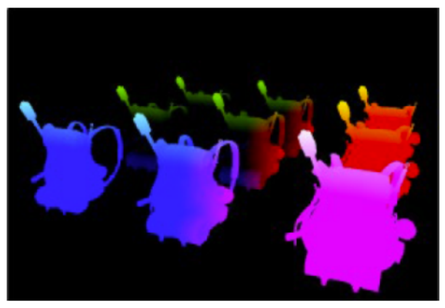
Save & Grade | Save only | New variant



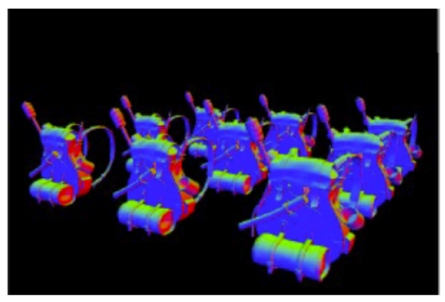


Geometry Buffers for Deferred Shading

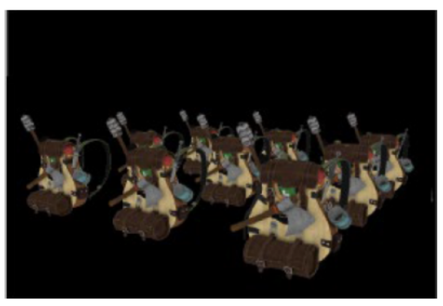
Deferred shading uses geometry buffers to store information needed for a final lighting computation over a scene. Each image below is a geometry buffer that stores a different kind of data. Match each image to the label that most likely describes the kind of data in the buffer.



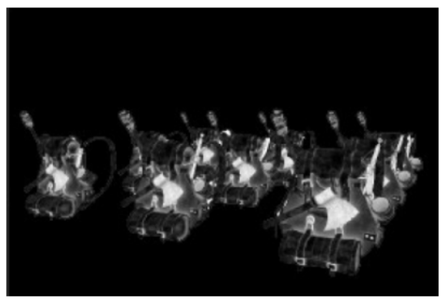
- (a) Light Intensity
- (b) Normals
- (c) Albedo (Diffuse Material)
- (d) Specular Material
- (e) Position



- (a) Position
- (b) Normals
- (c) Specular Material
- (d) Albedo (Diffuse Material)
- (e) Light Intensity



- (a) Specular Material
- (b) Normals
- (c) Albedo (Diffuse Material)
- (d) Light Intensity
- (e) Position



- (a) Position
- (b) Light Intensity
- (c) Specular Material
- (d) Albedo (Diffuse Material)
- (e) Normals

Save & Grade | Save only | New variant

Page 9
deferred shading

Ray Calculation

Imagine that we generate a ray from the camera to the viewplane through the center of a pixel (c, r) where $c = 1$ and $r = 5$.

The viewplane is a distance $d = 1$ away from the camera, located at $z = -d$. You can assume that the camera is set at $(0, 0, 0)$ and the viewplane is perpendicular to the z axis.

Assume an 8×6 pixel grid centered at $(0, 0, -d)$ and a pixel extent $s = 6$.

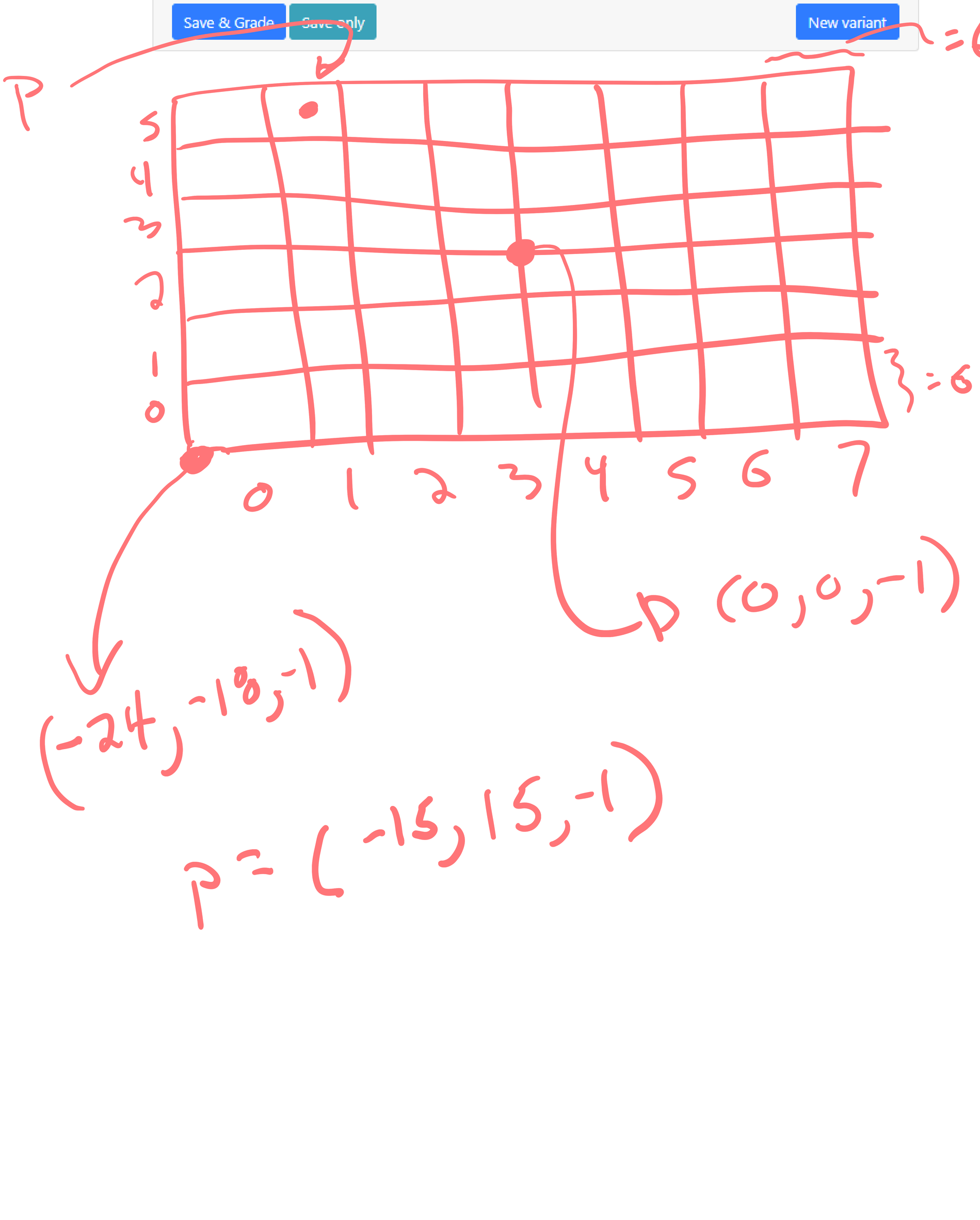
To be clear, c is the column of the pixel and r is the row. Assume that indexing starts at $(0, 0)$ at the lower left corner pixel.

Calculate the origin and direction of the generated ray. Do not normalize the direction.

$Origin(x, y, z) =$ [ ] [ ] [ ]

$Direction(x, y, z) =$ [ ] [ ] [ ]

Save & Grade | Save only | New variant



The Ray Parameter

We shoot a ray $r(t) = O + t\vec{d}$ from the eyepoint e through a point p on the viewplane so that $\vec{d} = p - e$

For $t \leq b_1$ the ray is not in front of the eyepoint. What is b_1?

$b_1 =$ number (2 significant figures)

For $b_2 < t < b_3$ the ray is in between the eyepoint and viewplane. What are b_2 and b_3?

$b_2 =$ number (2 significant figures)

$b_3 =$ number (2 significant figures)

Suppose we make $\vec{d} = \frac{\vec{d}}{\|\vec{d}\|}$ so that $\vec{d}$ is unit length.
Let $e = (0, 0, 0)$ and $p = (10.5, 5.5, -5)$.
If the ray hits an object at $t = 32$ what is the Euclidean distance r from the hitpoint to the eyepoint?

If it is impossible to determine r from the given information, answer that $r = 0$.

$r =$ number (2 significant figures)

If our ray fails to hit an infinite plane with normal $\vec{n}$, what is the value of $\vec{d} \cdot \vec{n}$

$\vec{d} \cdot \vec{n} =$ number (2 significant figures)

Save & Grade | Save only | New variant