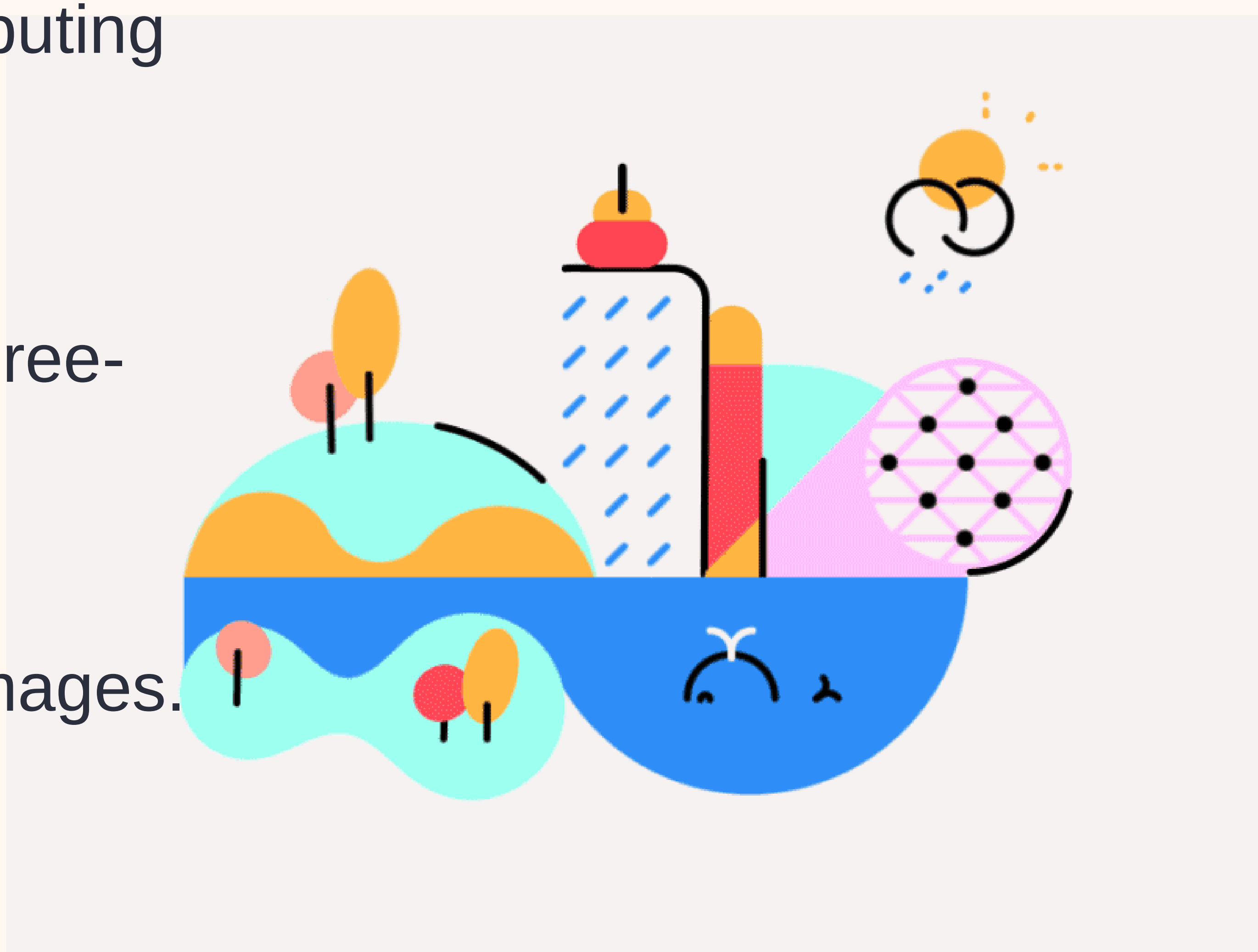
Introduction to 2D and 3D Graphics

Introduction

- Graphics play a crucial role in modern computing and visual media.
- Graphics can be two-dimensional (2D) or three-dimensional (3D).
- 2D graphics refer to flat, two-dimensional images.
- 3D graphics create the illusion of depth and volume.



Introduction

- 3D graphics allow for more realistic and immersive digital experiences.
- Understanding the fundamentals of both 2D and 3D graphics is essential.
- This knowledge is important for developers, designers, and artists.
- Relevant industries include:
  - Video games
  - Animation
  - Scientific visualization
  - Digital design

Display Devices: Introduction to CRT

1 Cathode Ray Tube (CRT)

The CRT, or Cathode Ray Tube, was the dominant display technology for many decades, used in televisions and computer monitors. CRTs work by firing a beam of electrons at a phosphor-coated screen, causing the phosphors to emit light and create an image.

2 Electron Beam Scanning

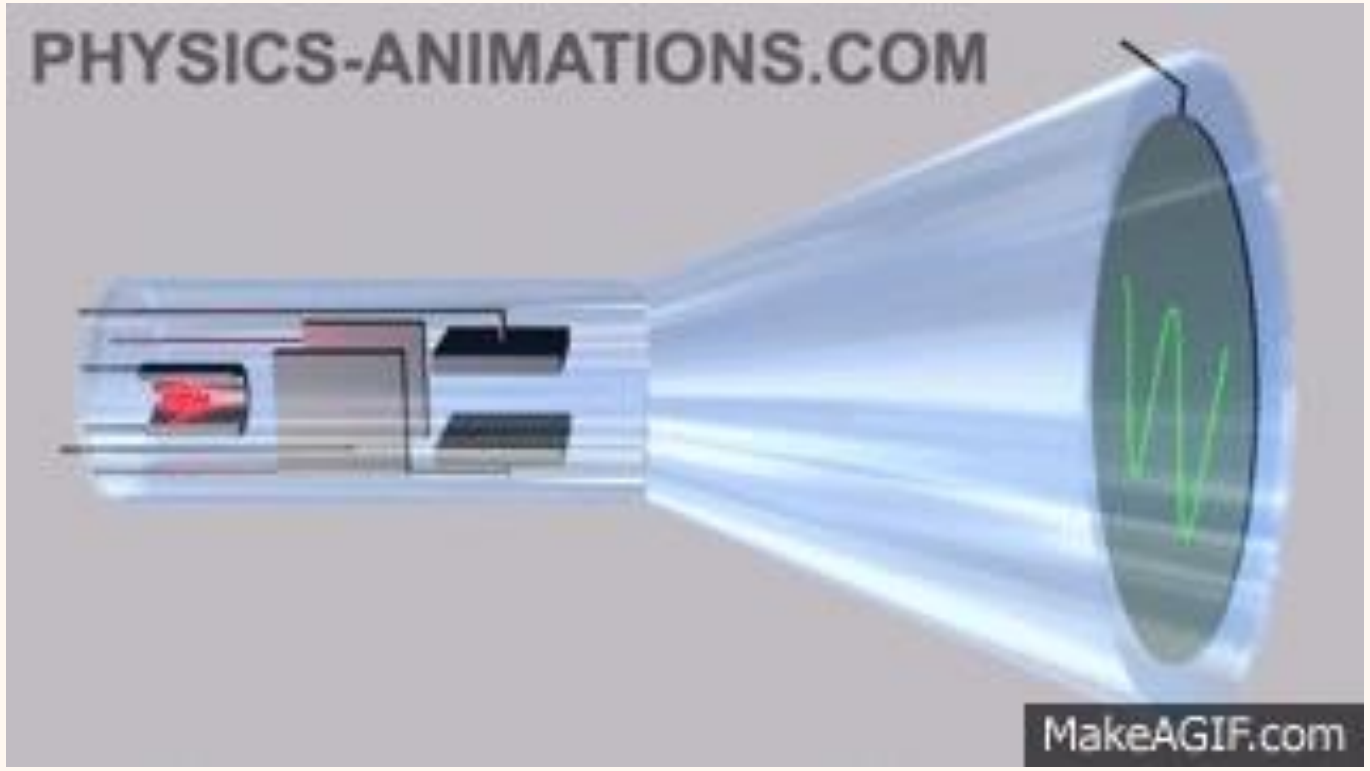
The electron beam in a CRT scans across the screen, line by line, to create the entire image. This process is known as raster scanning, and it allows for the display of both static and dynamic images.

Display Devices: Introduction to CRT

3

Analog Signals

CRT displays rely on analog signals to control the intensity and position of the electron beam, resulting in a continuous and smooth image. This differs from the digital signals used in modern display technologies.

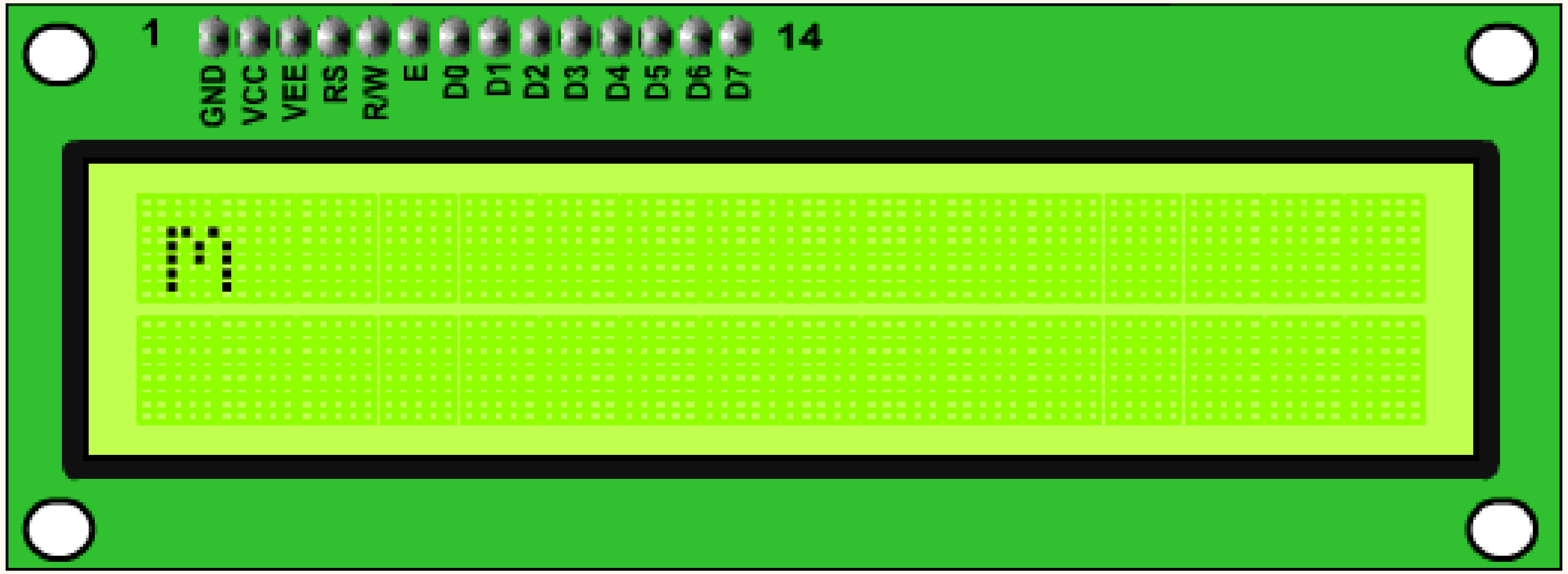


Flat Panel Displays

How LCD Works

LCD displays utilize a liquid crystal layer sandwiched between two layers of glass or plastic. Here’s a step-by-step breakdown of how they function:

1. Backlight
2. Liquid Crystal Layer
3. Manipulation of Light
4. Color Filters
5. Creating Images



Flat Panel Displays

Applications of LCD displays

1. Monitors
2. Televisions
3. Portable Devices

Flat Panel Displays

Plasma Displays

Plasma displays are a type of flat panel display technology that uses small cells containing electrically charged ionized gases (plasma) to produce images. They were once popular for large-screen televisions due to their excellent color reproduction, high contrast ratios, and wide viewing angles.

Flat Panel Displays

Working Principle of Plasma Displays

1. **Gas Cells**: Contains small cells filled with ionized gases (neon and xenon).
2. **Electrical Stimulation**: When electricity excites the gas, it emits ultraviolet light.
3. **Phosphor Coating**: Ultraviolet light interacts with phosphor coating inside the cells, producing visible light in red, green, and blue colors.

Flat Panel Displays

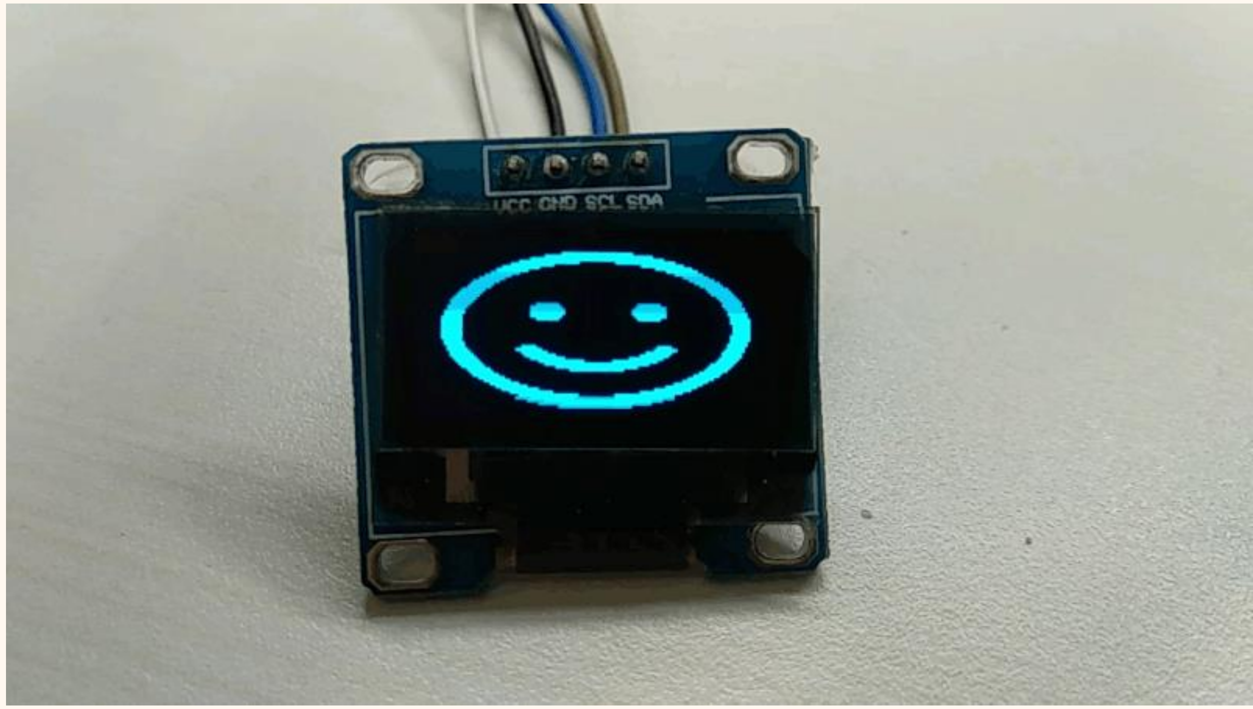
Applications of Plasma Displays

1. **Large Televisions**: Known for their high-quality display, especially in larger screen sizes.
2. **Public Displays**: Used in venues like airports and sports arenas due to their bright, vivid display.

Flat Panel Displays

OLED

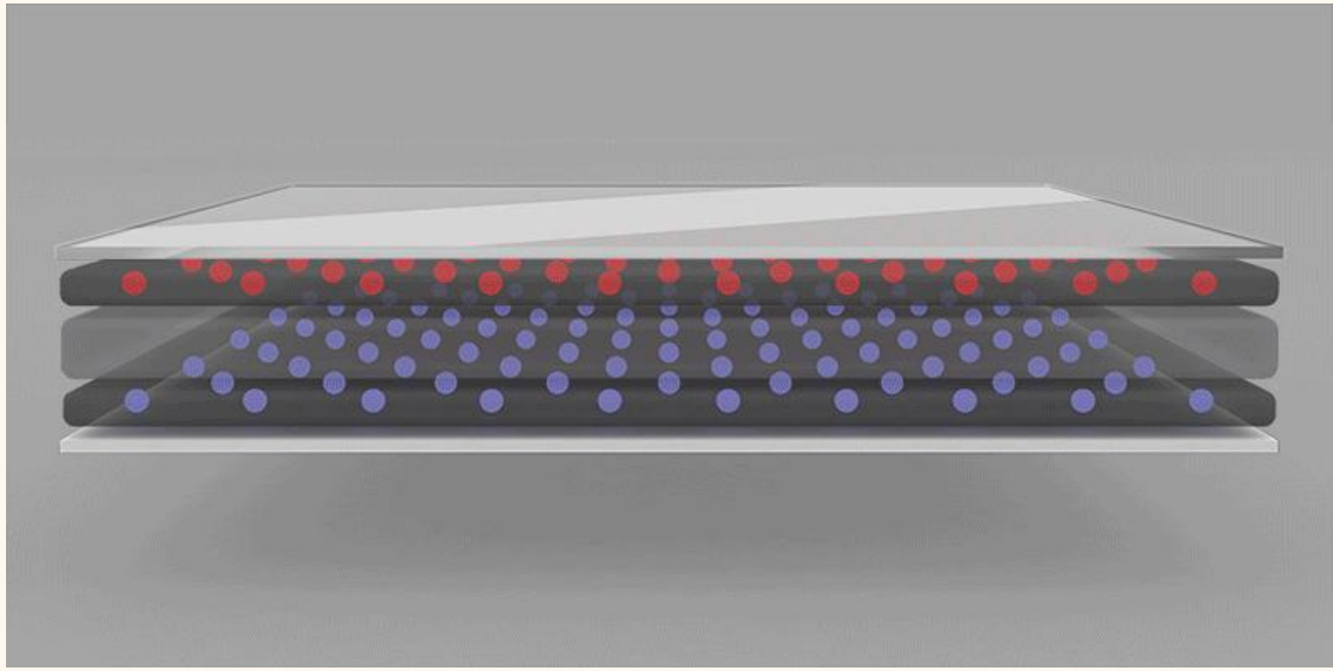
Organic Light-Emitting Diode (OLED) displays represent a significant advancement in display technology. Unlike traditional LCDs and plasma displays, OLEDs do not require a backlight and offer numerous advantages in terms of image quality, design flexibility, and energy efficiency.



Flat Panel Displays

Working Principle of OLED

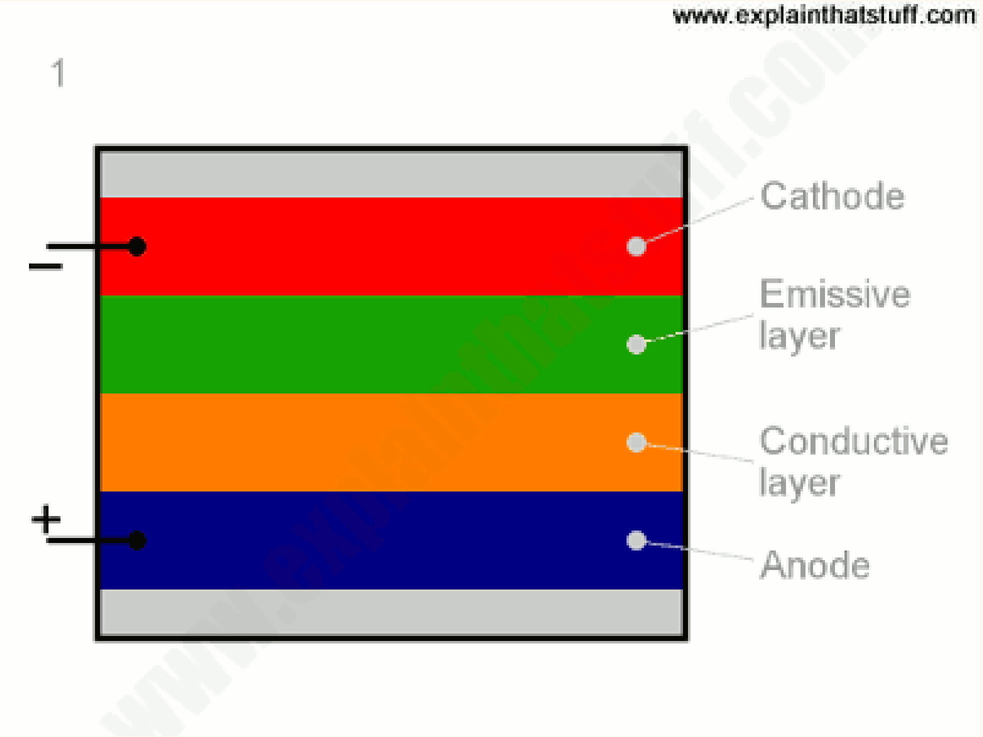
1. **Organic Materials**: Uses organic compounds that emit light when an electric current is applied.
2. **Self-Emitting Pixels**: Each pixel emits its own light, eliminating the need for a backlight.
3. **Color Creation**: Each pixel can emit red, green, and blue light directly, producing vibrant colors.



Flat Panel Displays

Applications of OLED

1. **High-End Televisions**: Preferred for their superior picture quality.
2. **Smartphones and Wearables**: Used in smartphones, smartwatches, and other portable devices.
3. **Flexible Displays**: Can be used in bendable or foldable screens.

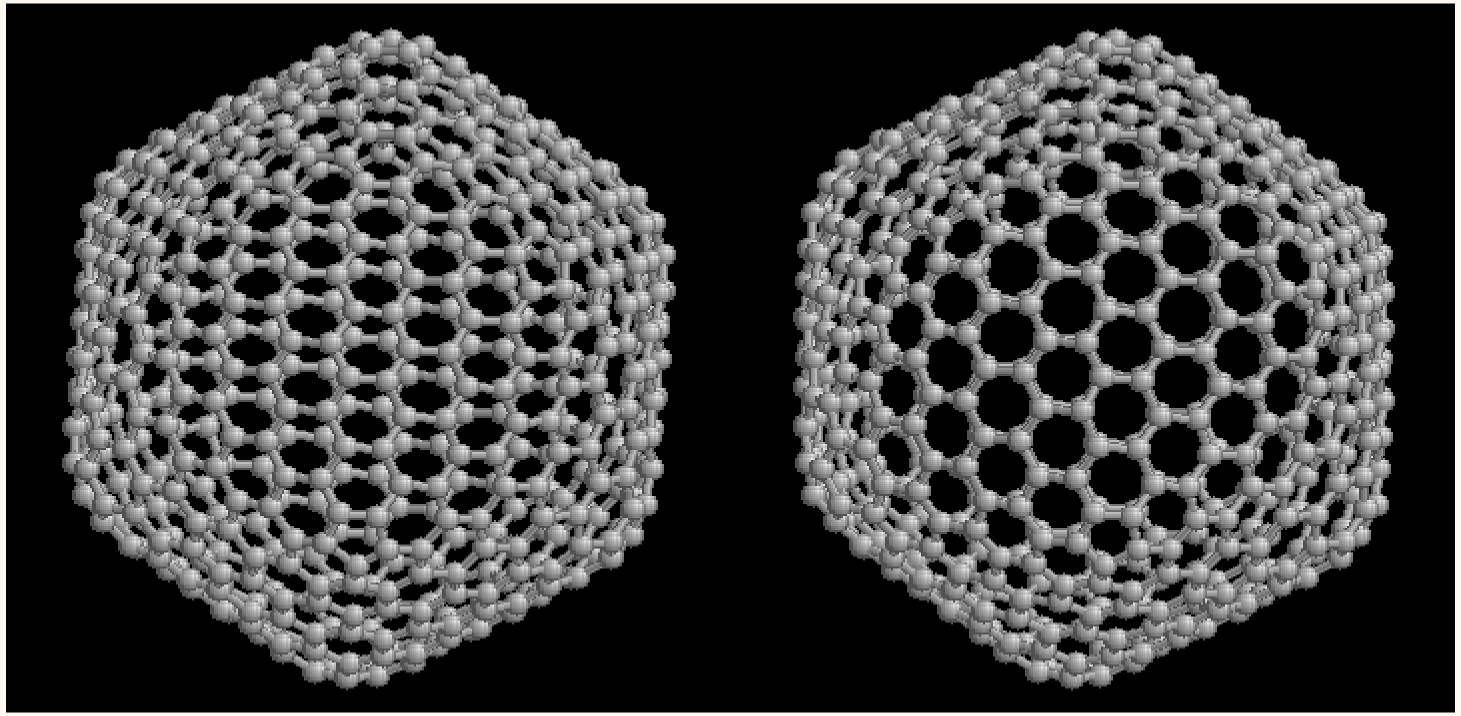


Stereoscopic 3D Displays

1

Binocular Disparity

Stereoscopic 3D displays work by leveraging the binocular disparity, or the slight difference in the images seen by our left and right eyes. This disparity is what allows our brain to perceive depth and create a 3D illusion.



Stereoscopic 3D Displays

2

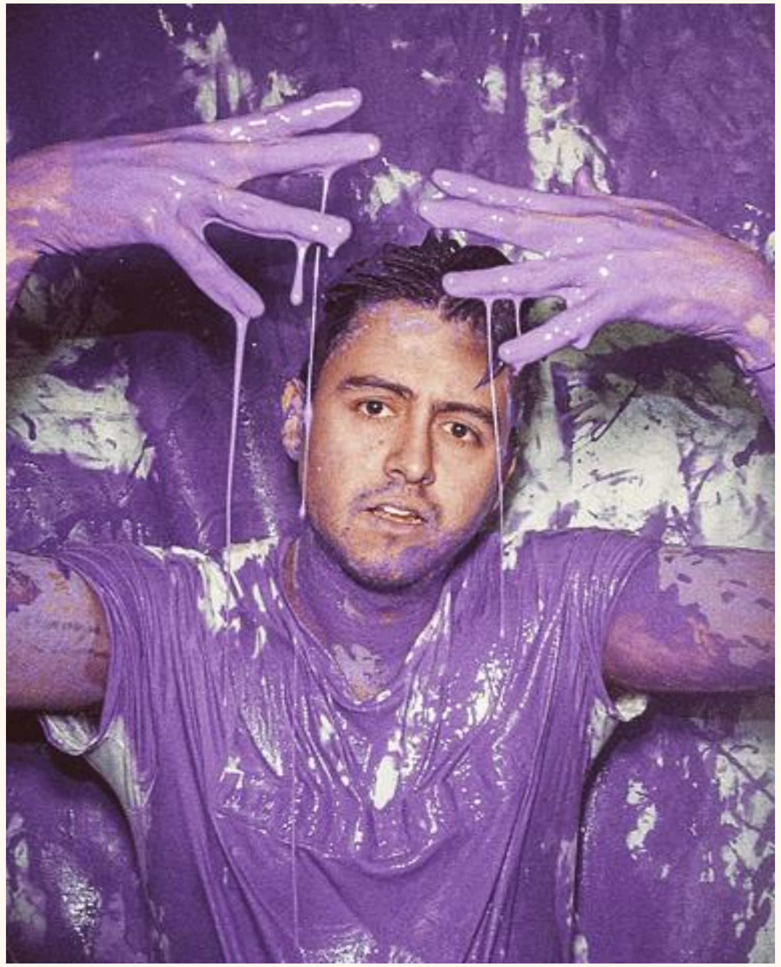
Polarized Glasses

One common method for stereoscopic 3D displays is the use of polarized glasses. The display presents slightly different images to the left and right eyes, which are then filtered by the polarized lenses, creating the 3D effect.

3

Active Shutter Glasses

Active shutter glasses, which rapidly open and close the left and right lenses in sync with the display, are another popular technique for stereoscopic 3D. This allows for a more immersive and realistic 3D experience.



Stereoscopic 3D Displays in Modern Technologies

Stereoscopic 3D displays create the illusion of depth by presenting slightly different images to each eye, mimicking the natural process of binocular vision. This technology has found applications across various fields:

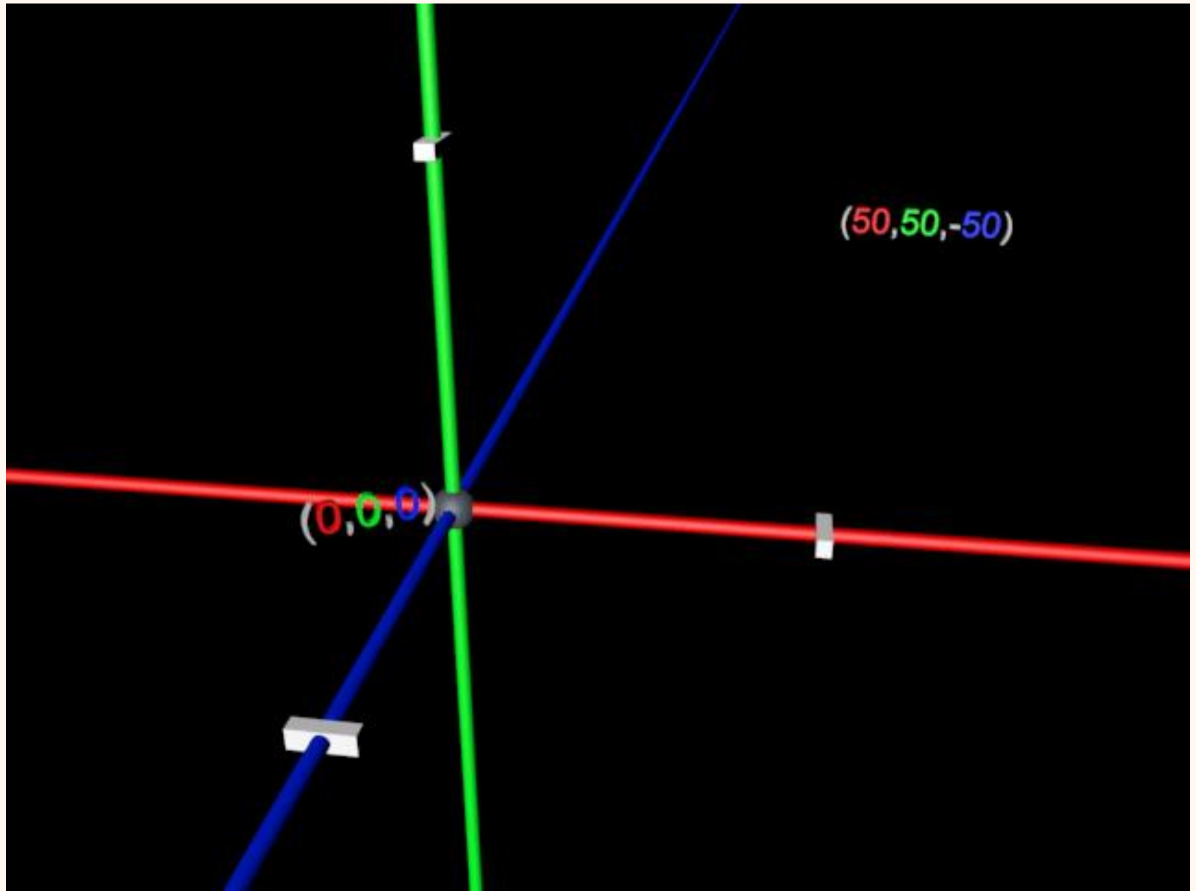
1. **Entertainment:** Stereoscopic 3D displays are widely used in movies and gaming to provide an immersive experience. The enhanced depth perception makes scenes more realistic and engaging for viewers and players.

Stereoscopic 3D Displays in Modern Technologies

2. **Medical Imaging**: In healthcare, stereoscopic 3D displays assist in visualizing complex anatomical structures, aiding in surgery planning and medical education. They provide more detailed views of patient scans, enhancing diagnostic accuracy.
3. **Engineering and Design**: In fields like architecture and product design, stereoscopic 3D displays allow professionals to visualize and manipulate 3D models more effectively. This helps in identifying design flaws and making better-informed decisions.

Overview of Cartesian Coordinate System

The Cartesian coordinate system, is a basic method used in many fields like math, physics, and engineering to describe the location of points. It helps represent shapes, plot functions, and analyze spatial relationships with numerical coordinates.



Overview of Cartesian Coordinate System

Axes

X-Axis

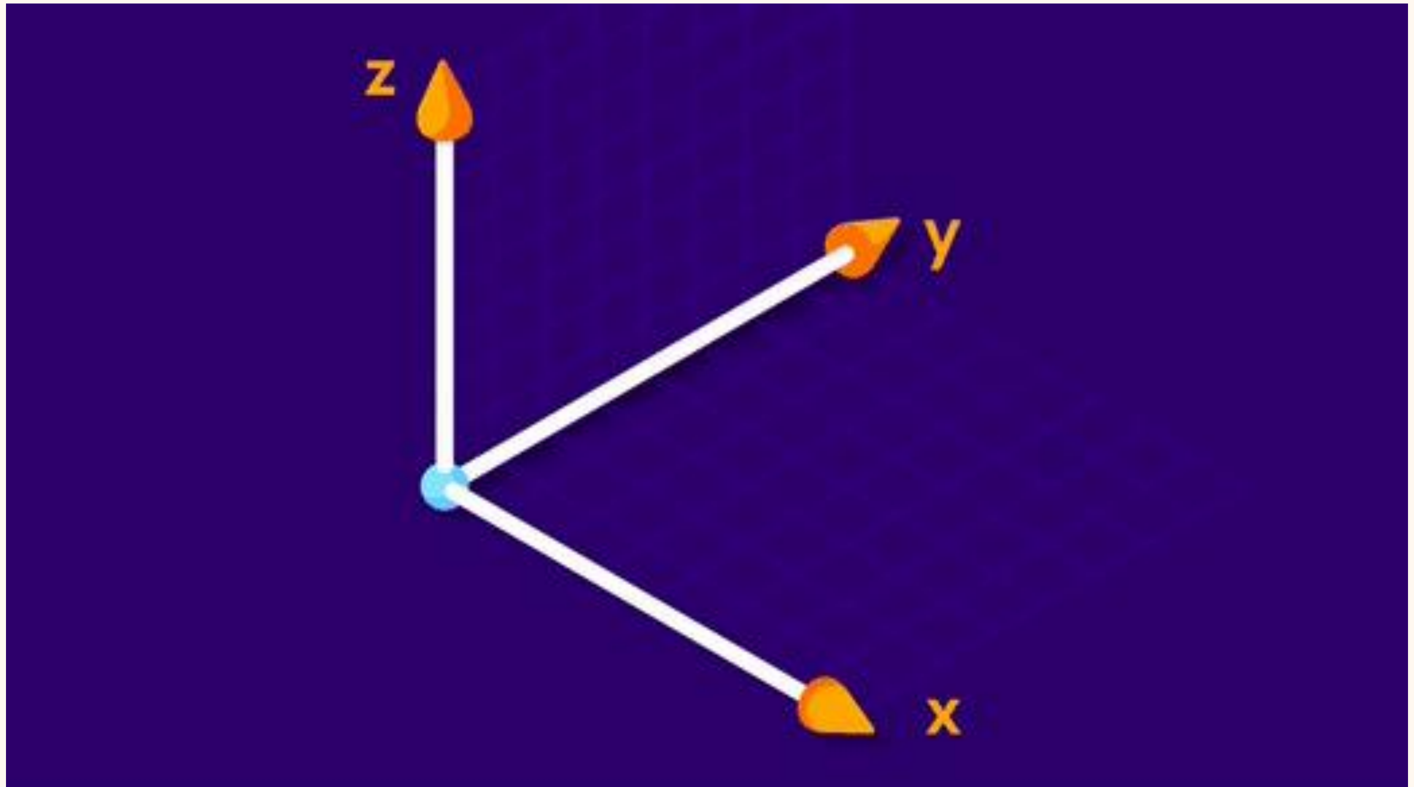
The horizontal axis in the Cartesian coordinate system, representing the left-right dimension.

Y-Axis

The vertical axis in the Cartesian coordinate system, representing the up-down dimension.

Z-Axis

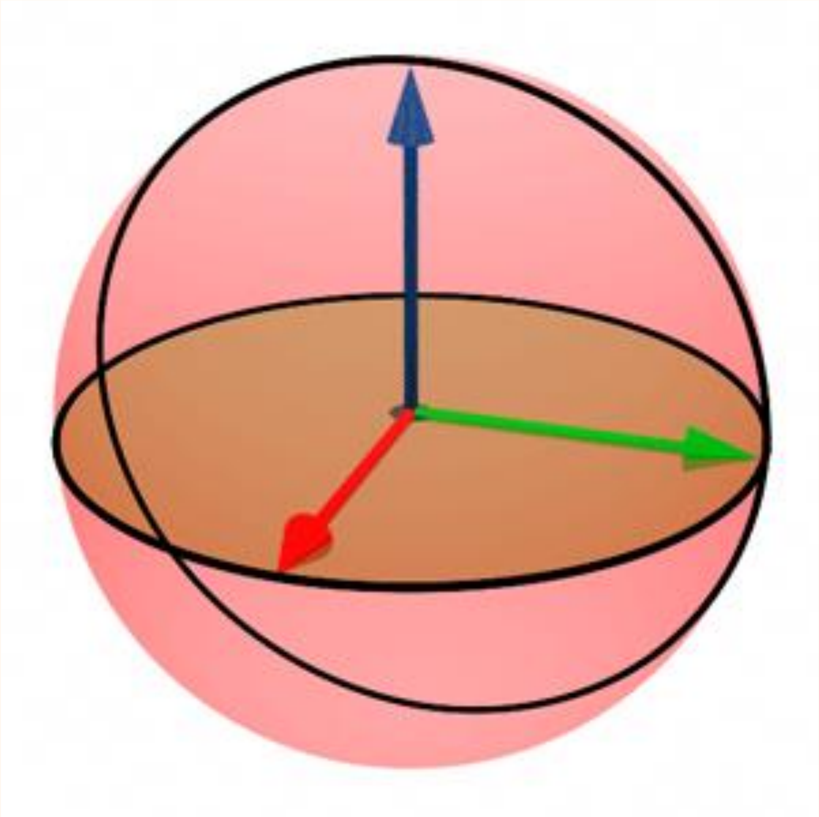
The depth axis in the 3D Cartesian coordinate system, representing the forward-backward dimension.



Overview of Cartesian Coordinate System

Coordinates

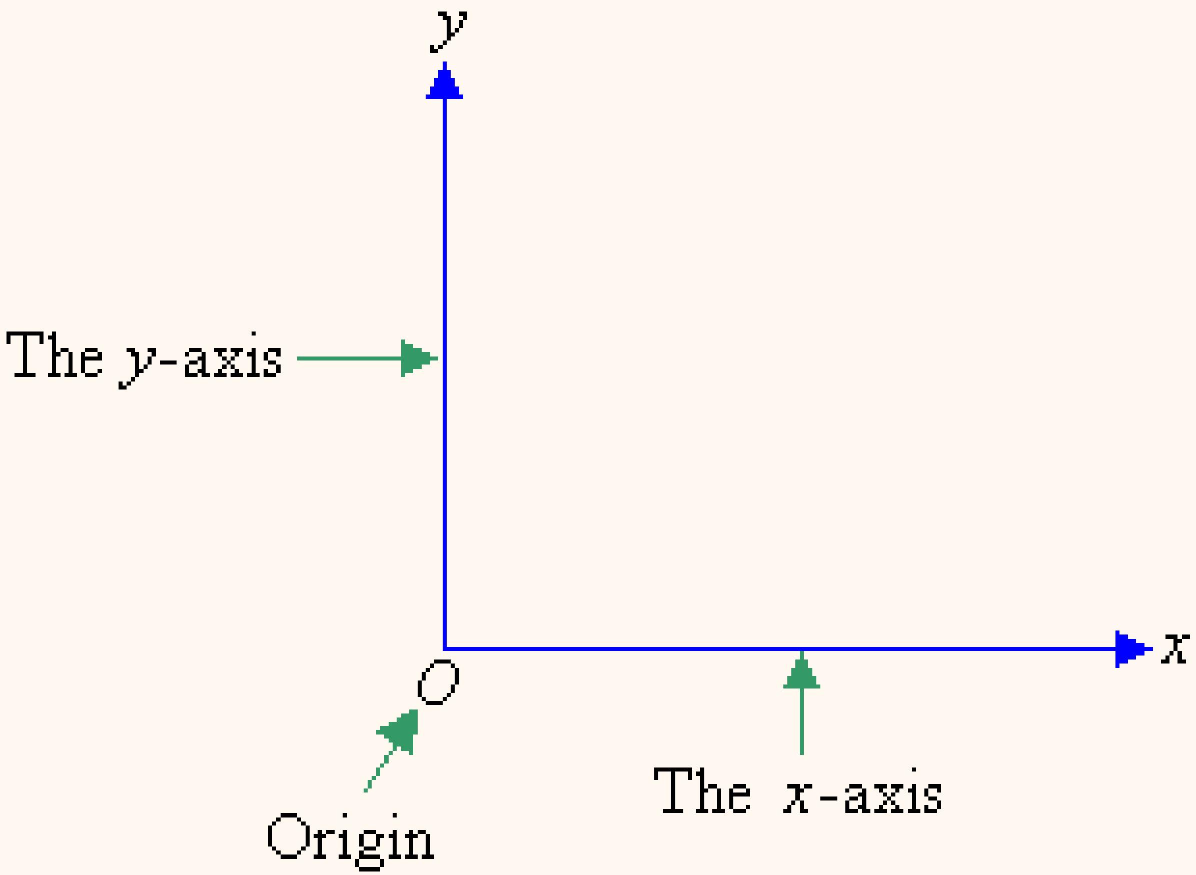
Each point in the Cartesian plane is represented by an ordered pair of numbers (x, y) in 2D, or an ordered triplet (x, y, z) in 3D. These numbers are the coordinates of the point and represent its distances from the axes.



Overview of Cartesian Coordinate System

Origin

The point where the X, Y, and (in 3D) Z axes intersect, representing the (0, 0) or (0, 0, 0) position.



Geometric Data Types

1

Point

- A single location in the coordinate system, defined by its X, Y, and (in 3D) Z coordinates.
- Used to define vertices in polygons, control points for curves, and other precise locations.

2

Curves

- Continuous shapes that are not straight.
- Used in vector graphics for smooth shapes, typography, and complex designs.

Geometric Data Types

3

Lines and Line Segments

- A one-dimensional object with a defined starting and ending point, representing a path in the coordinate system.
- Used in drawing basic shapes, paths, and borders.

4

Polygon

- A closed shape defined by a series of connected line segments, used to represent 2D and 3D objects.
- Essential for creating complex shapes, vector graphics, and meshes.

Vectors

Magnitude and Direction

Vectors are defined by both their magnitude (length) and direction. The magnitude represents the strength or size of the vector, while the direction indicates the orientation in the coordinate system.

Vector Operations

Vectors can be added, subtracted, and multiplied, allowing for the creation of more complex geometric relationships and transformations. These operations are essential for many 2D and 3D graphics applications.

Coordinate Representation

Vectors can be represented by their coordinate values, such as (x, y) for 2D or (x, y, z) for 3D. This allows for the precise mathematical manipulation and computation of vector-based graphics and simulations.

Examples: Velocity, Acceleration, Force

Coordinate Vectors and Bases

Coordinate Vectors

Coordinate vectors are the fundamental building blocks of the Cartesian coordinate system, representing the individual axes (X, Y, and Z) and their corresponding directions.

Vector Basis

A vector basis is a set of linearly independent coordinate vectors that span the entire coordinate system, allowing any point or vector to be expressed as a linear combination of the basis vectors.

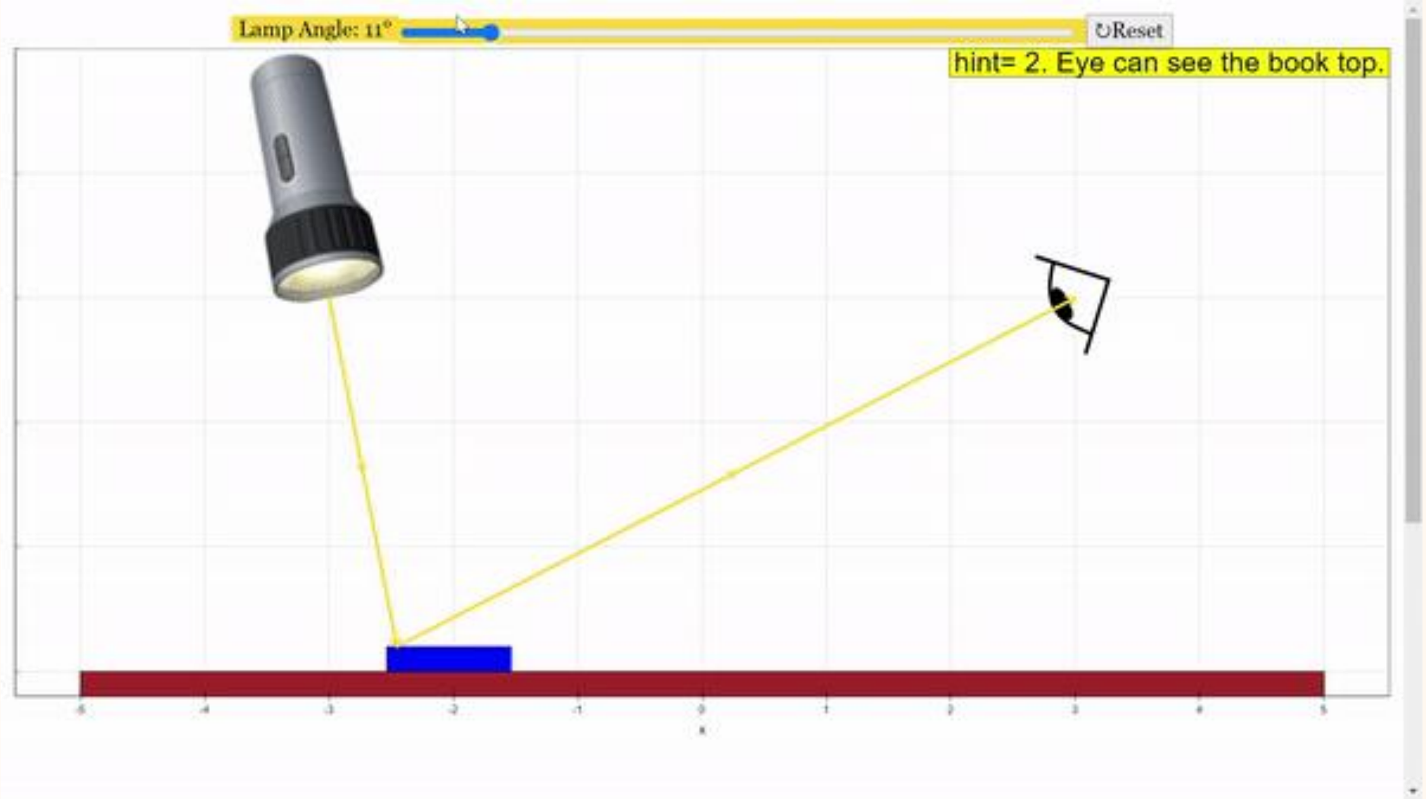
Coordinate Transformations

Understanding the relationship between coordinate vectors and vector bases is crucial for performing coordinate transformations, such as rotations, translations, and scaling, which are essential for 2D and 3D graphics applications.

Basic Lighting and Reflection

Lighting

Lighting in graphic design involves simulating light sources to enhance the realism and depth of the scene. It includes various components such as ambient, diffuse, and specular lighting, each contributing to how light interacts with surfaces.

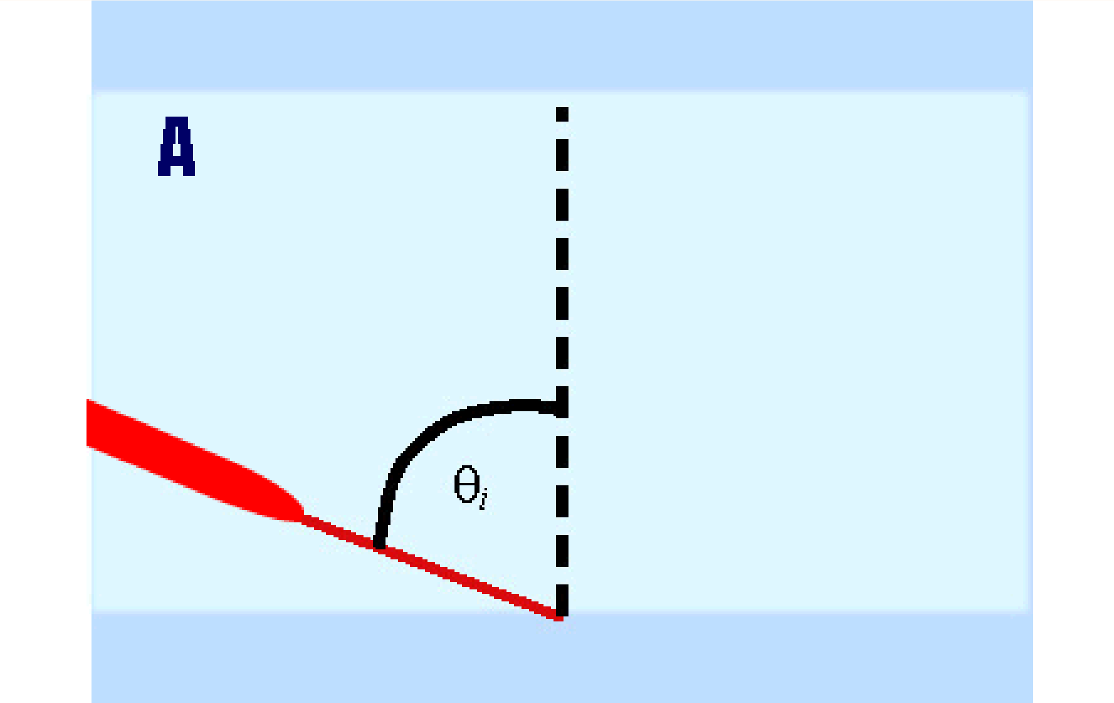


Lamp Angle: 11°

hint= 2. Eye can see the book top.

Basic Lighting and Reflection

Here are some fundamental aspects:



Ambient Lighting: This is the general illumination that affects all objects in the scene uniformly.

Diffuse Lighting: This simulates the light scattered in many directions when it hits a rough surface, giving the object a matte appearance.

Specular Lighting: This models the reflection of light from shiny surfaces, creating highlights and enhancing the perceived shininess of the object.

Basic Lighting and Reflection

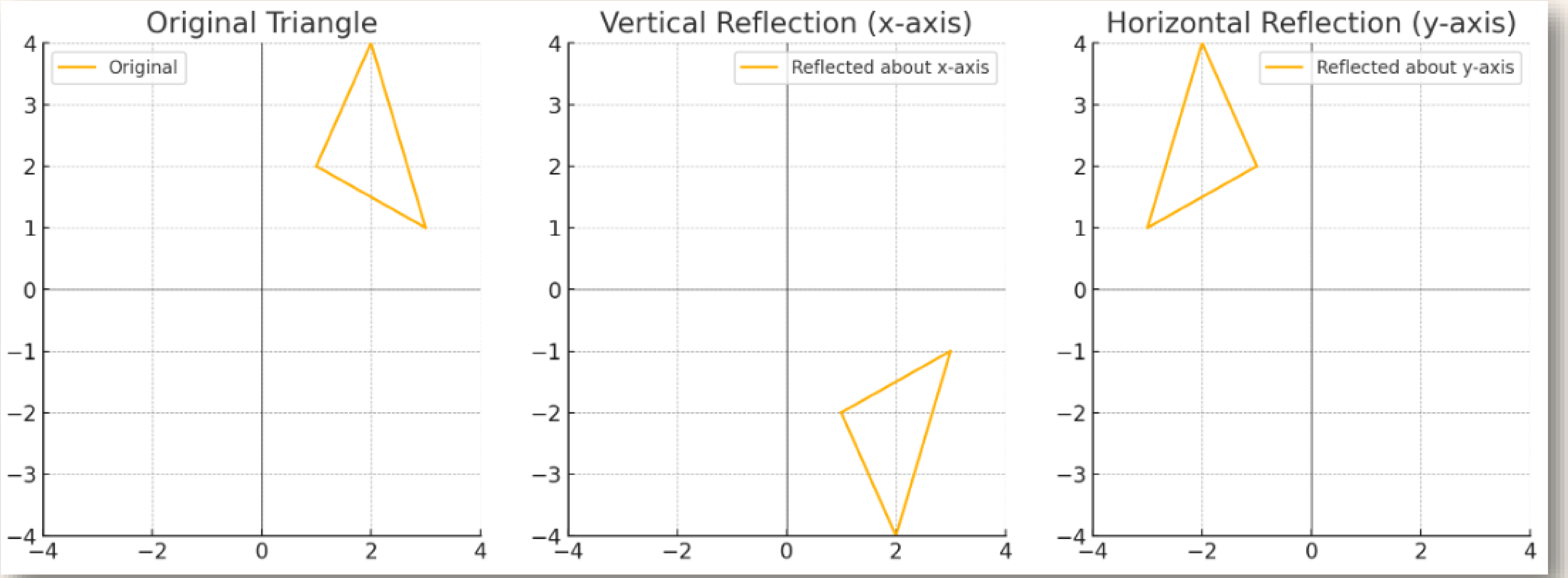
Reflection

Reflection is the phenomenon of light rays bouncing back when they strike a smooth, shiny surface. It creates a mirror image of the object and follows the laws of reflection, where the angle of incidence equals the angle of reflection.

Basic Lighting and Reflection

2D Reflection:

- About the x-axis: Reflecting an object vertically.
- About the y-axis: Reflecting an object horizontally.



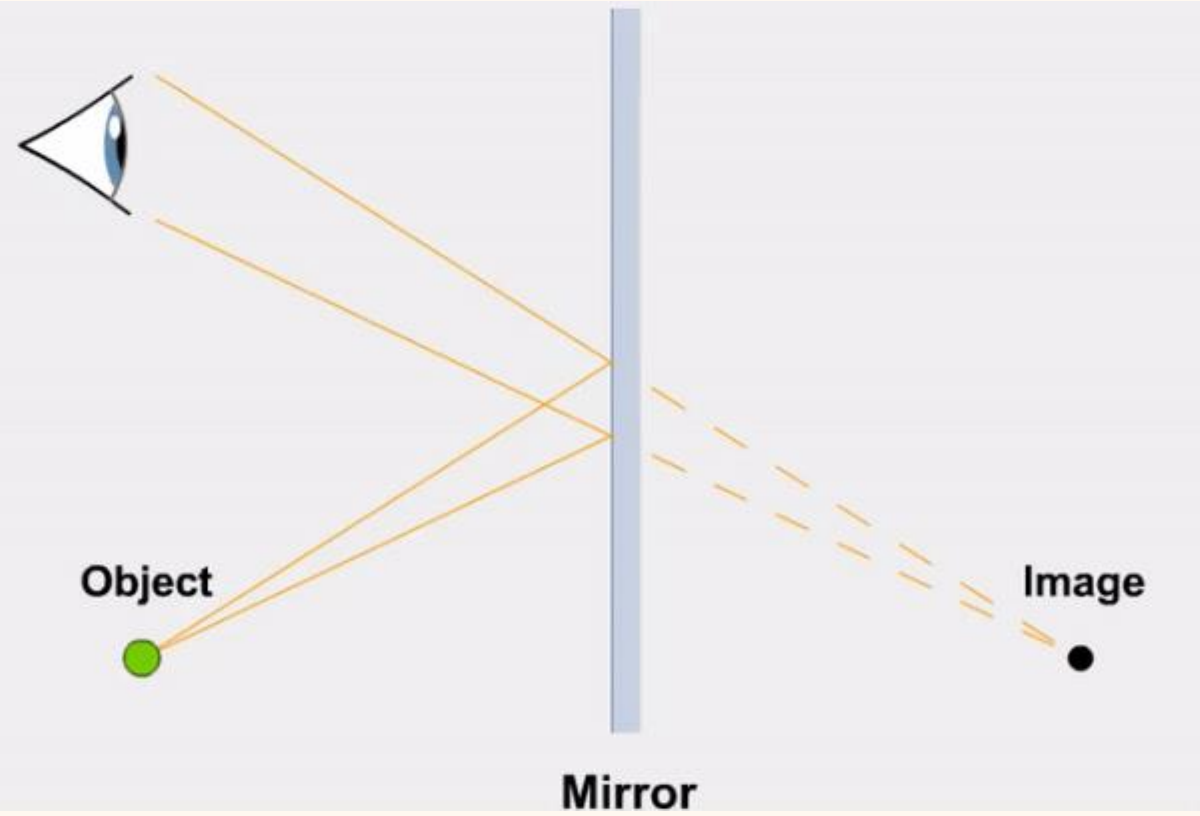
Basic Lighting and Reflection

3D Reflection:

Similar to 2D reflection but involves reflecting objects across planes such as the xy-plane, yz-plane, or xz-plane. This produces a mirror image in three-dimensional space.

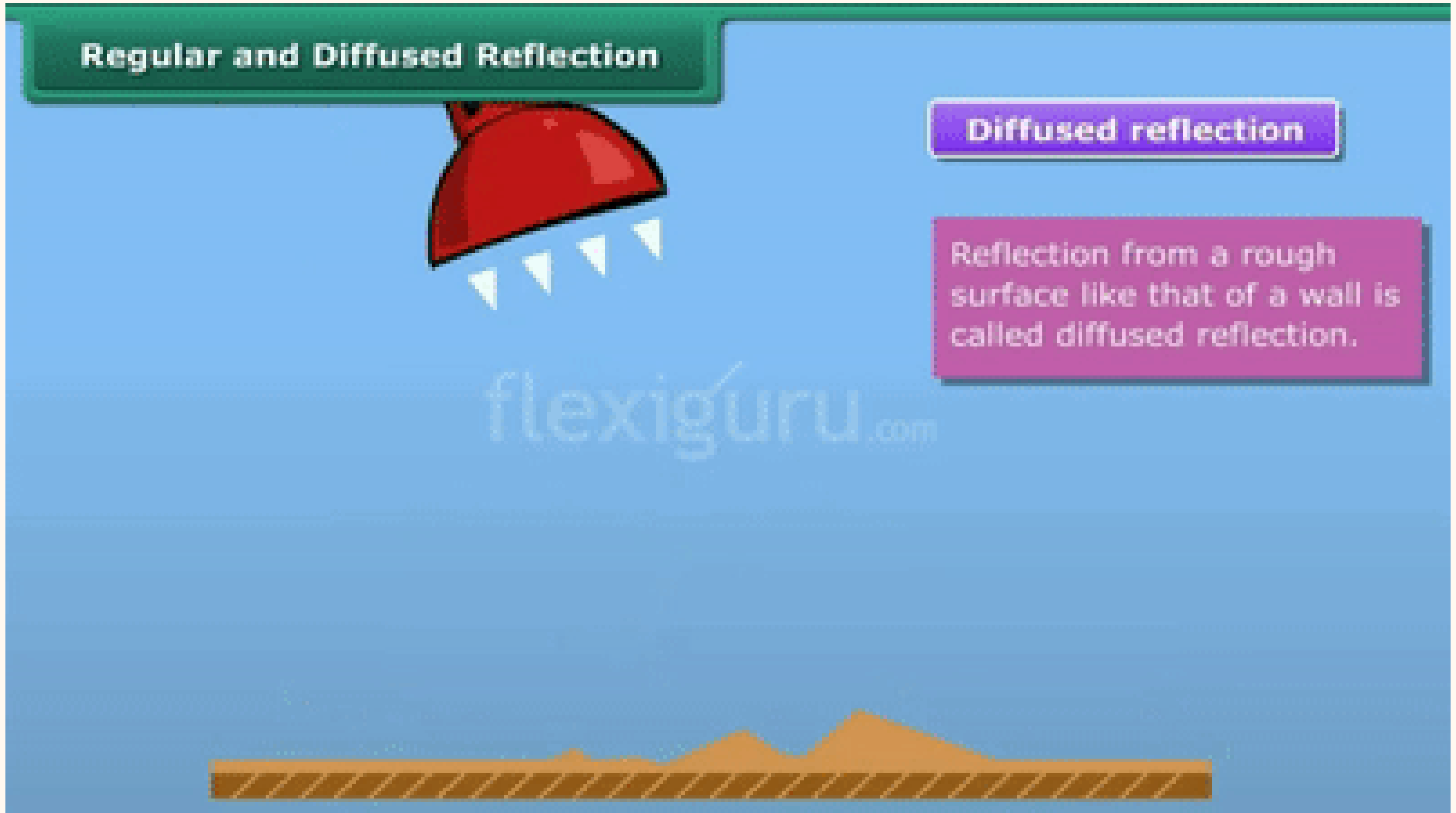
Simple Reflection Models

A **reflection model** in computer graphics is a mathematical and conceptual framework used to simulate how light reflects off surfaces. It defines how light interacts with a surface to produce effects such as diffuse and specular reflections, contributing to the realism of rendered images. Simple reflection models typically focus on basic reflection types like diffuse reflection, where light is scattered uniformly, and specular reflection, where light is reflected in a specific direction.



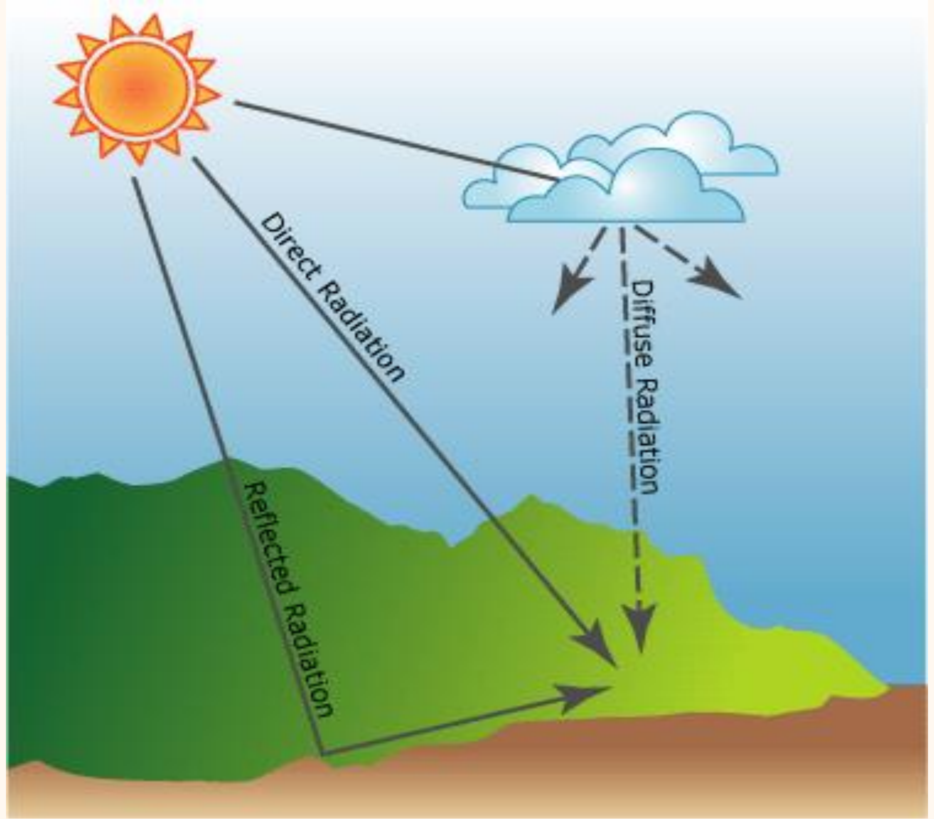
Simple Reflection Models – Diffuse

- Diffuse reflection is a fundamental concept in computer graphics where light interacts with surfaces.
- Unlike specular reflection, which results in a mirror-like reflection, diffuse reflection causes light to scatter uniformly in all directions when it hits a surface.

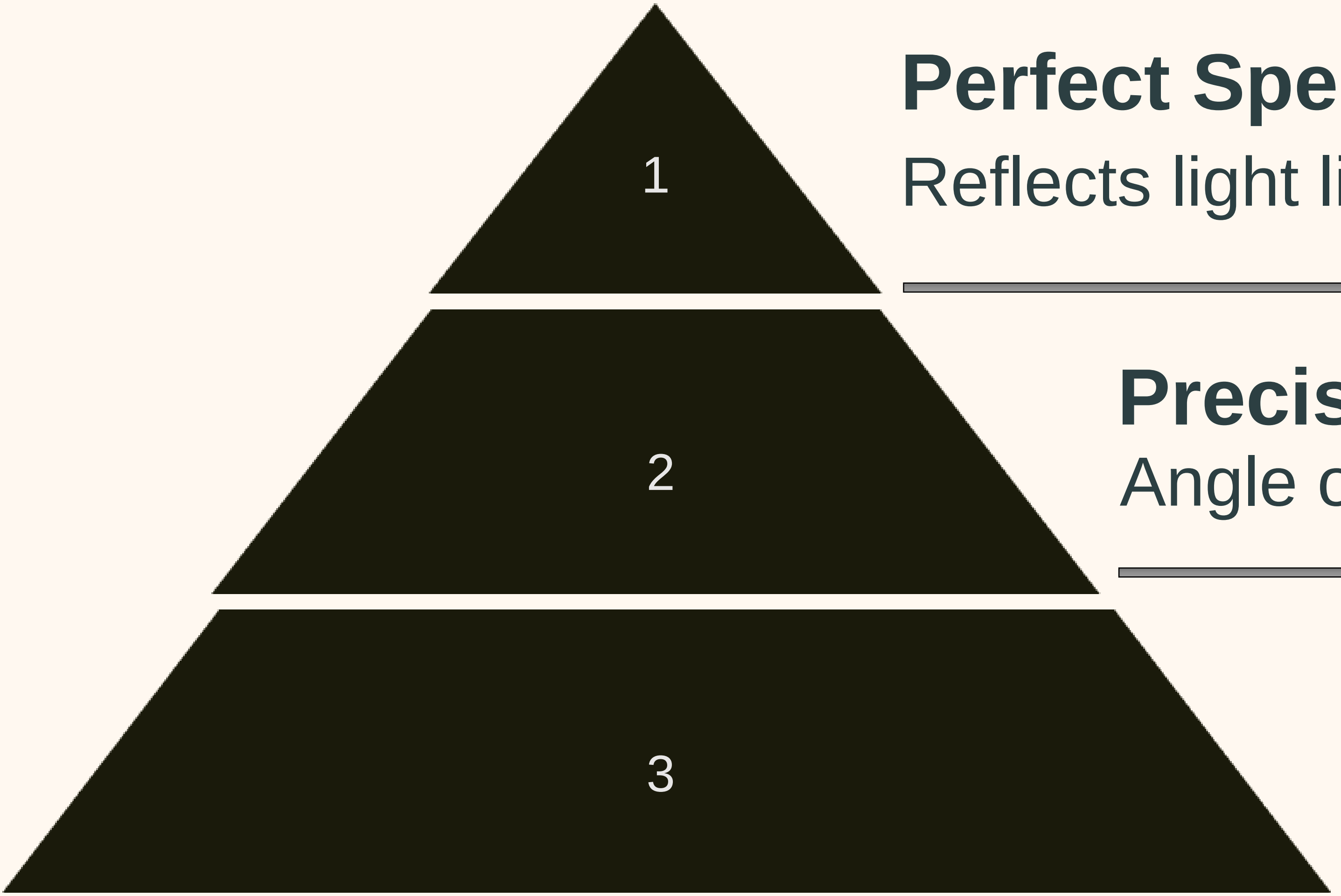

Regular and Diffused Reflection
Diffused reflection
Reflection from a rough surface like that of a wall is called diffused reflection.
flexiguru.com

Simple Reflection Models – Diffuse

- In 3D graphics, diffuse reflection plays a crucial role in shading models like Lambertian reflection.
- Lambertian surface appears equally bright from all viewing angles, as it reflects light diffusely and not specularly.
- Lambertian reflection's intensity of reflected light depends on the angle between the light source direction and the surface normal, simulating realistic lighting conditions for non-metallic objects.

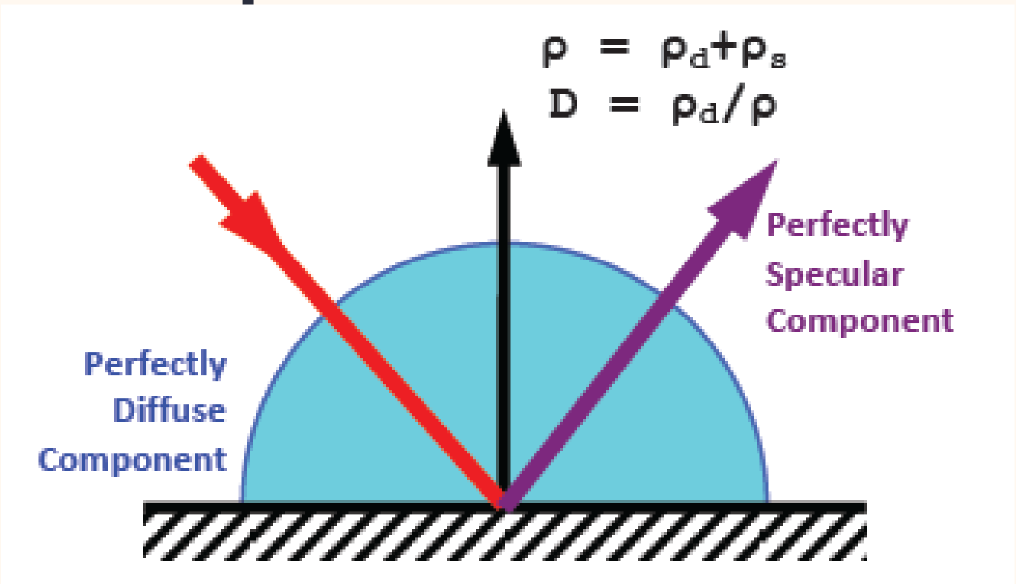


Simple Reflection Models – Perfect Specular



Perfect Specular

Reflects light like a mirror



Precise Reflection

Angle of reflection equals angle of incidence

Highlights

Bright spots on shiny surfaces

In a perfect specular reflection model, the surface acts like a mirror, reflecting light at the same angle as it hits the surface. This creates bright, precise highlights on shiny objects, where the reflected light is concentrated in a small area. This type of reflection is important for rendering realistic metallic and glossy surfaces.

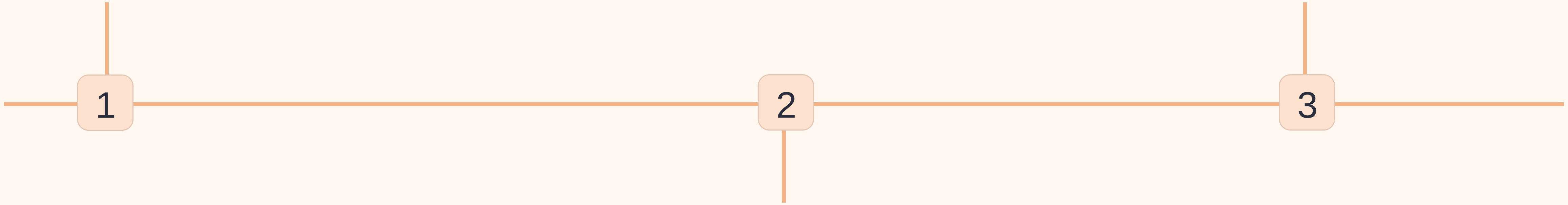
Simple Reflection Models – General Specular

Blending Diffuse and Specular

General specular models combine diffuse and specular reflection to create a more realistic appearance. The diffuse component provides the base color while the specular component adds highlights.

Anisotropic Reflections

Advanced specular models can account for anisotropic surfaces, where reflections vary based on the orientation of the surface. This creates realistic metallic or brushed appearances.

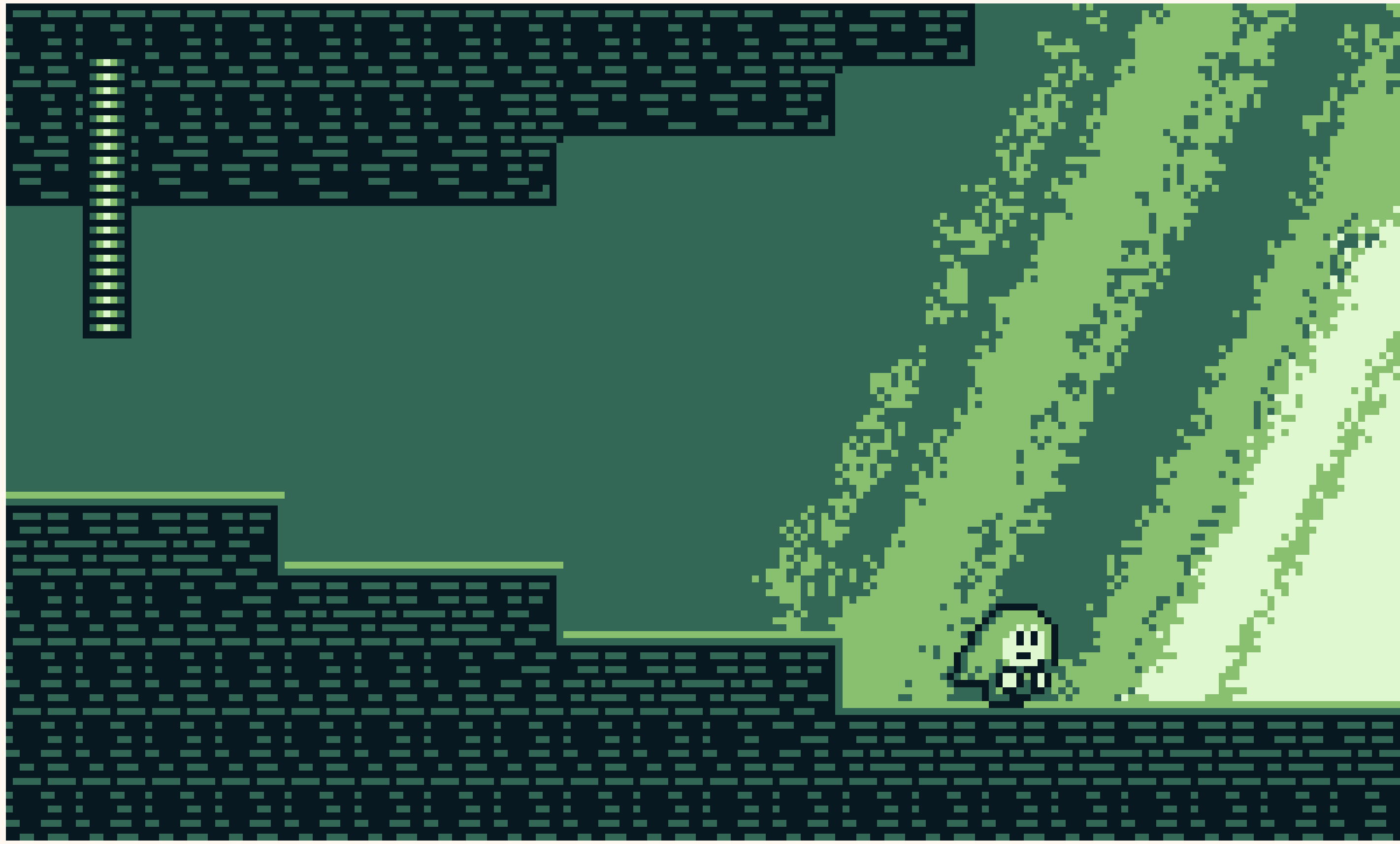


Controlling Shininess

The shininess parameter adjusts the size and intensity of specular highlights. Higher shininess values create tighter, brighter highlights, while lower values result in broader reflections.

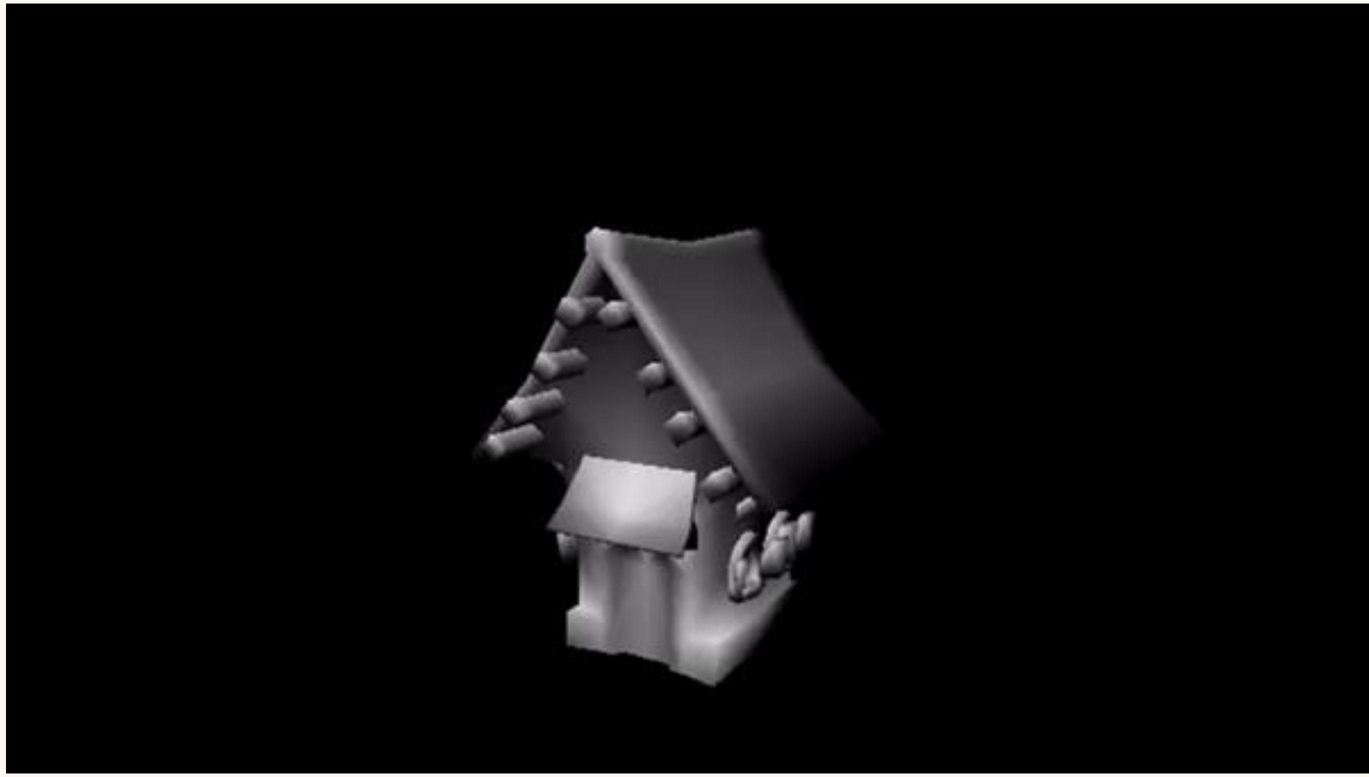
Shading

- Shading is the process of adding color or darkness to an object or representation to give it depth, form, and realism.
- It simulates the effects of light and shadow on surfaces, enhancing visual perception by creating contrasts, gradients, and textures.

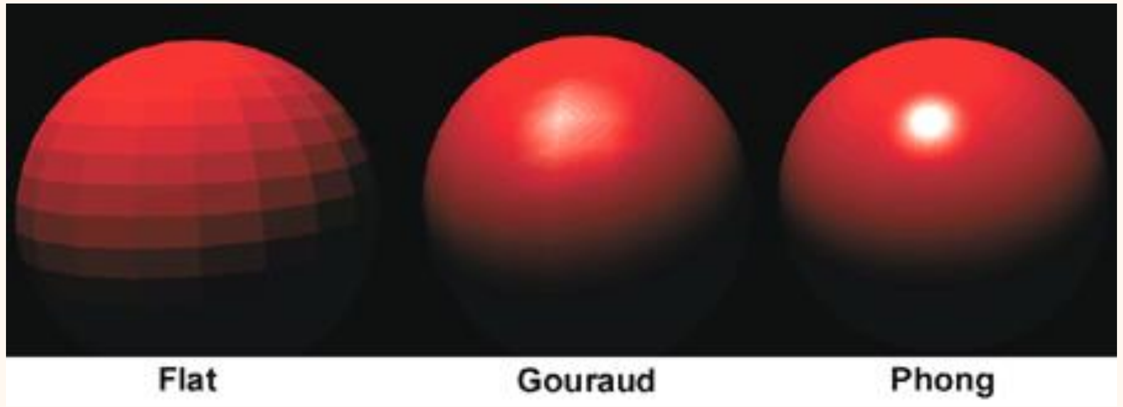


Shading

- Shading techniques are used to mimic how light interacts with materials, crucial for realistic lighting effects in 3D rendering.
- Shading is essential in visual arts, animation, gaming, and virtual simulations, adding depth and dimension to flat images.



Shading – Flat Shading



1

Simplest Shading Model

Assigns a single color to each polygon

2

Calculates Lighting Once

Computes lighting at polygon center

3

Fast and Efficient

Requires minimal computation

In flat shading, **no textures are used and only one colour tone is used for the entire object, with different amounts of white or black added to each face of the object to simulate shading**. The resulting model appears flat. This simple approach is fast and efficient, making it well-suited for real-time rendering, but lacks the smooth appearance of more advanced shading techniques.

Shading – Interpolative shading

1 Smooth Transitions

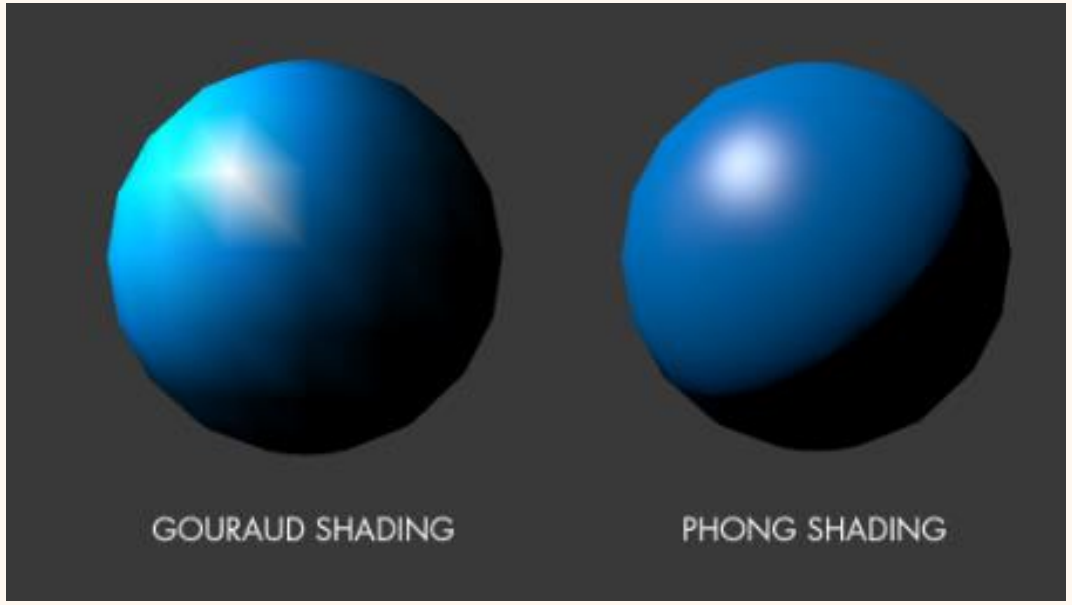
Interpolative shading creates smooth, continuous shading between vertices by interpolating the values of the vertices. This results in a more natural and realistic appearance compared to flat shading.

2 Gouraud Shading

Gouraud shading is a technique that smooths the appearance of surfaces by averaging the normals at each vertex and using these averages to calculate the color at every pixel.

3 Phong Shading

Phong shading computes the surface normal at each pixel from vertex data, creating realistic specular highlights.



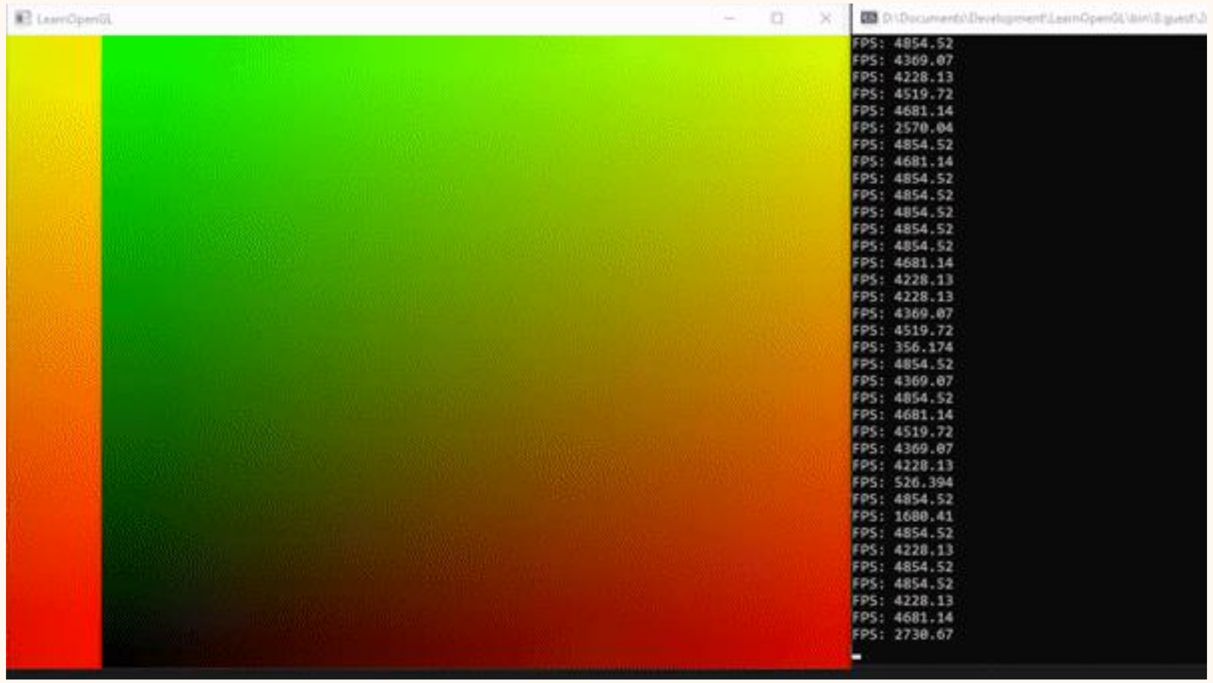
Shading – Shading in OpenGL

OpenGL (Open Graphics Library) is a cross-platform API (Application Programming Interface) for rendering 2D and 3D vector graphics. Here are key points about OpenGL:

1. Developed to provide a standardized platform-independent interface for hardware-accelerated rendering of 2D and 3D graphics.
2. Allows developers to access graphics hardware's features through a set of functions to render complex scenes from simple primitives.

Shading – Shading in OpenGL

3. Evolves through versions, each offering improved functionality and efficiency. Developers use specific versions compatible with their hardware and software requirements.
4. Supports extensions for additional features beyond the core functionality, allowing flexibility and adaptation to new technologies.



```
FPS: 4854.52
FPS: 4369.07
FPS: 4228.13
FPS: 4519.72
FPS: 4681.14
FPS: 2570.04
FPS: 4854.52
FPS: 4681.14
FPS: 4854.52
FPS: 4854.52
FPS: 4854.52
FPS: 4854.52
FPS: 4854.52
FPS: 4681.14
FPS: 4228.13
FPS: 4228.13
FPS: 4369.07
FPS: 4519.72
FPS: 356.174
FPS: 4854.52
FPS: 4369.07
FPS: 4854.52
FPS: 4681.14
FPS: 4519.72
FPS: 4369.07
FPS: 4228.13
FPS: 526.394
FPS: 4854.52
FPS: 1680.41
FPS: 4854.52
FPS: 4228.13
FPS: 4854.52
FPS: 4854.52
FPS: 4228.13
FPS: 4681.14
FPS: 2730.67
```

Shading – Shading in OpenGL

Tiling & Repeating

Tiling and repeating are techniques used to cover large surfaces with a texture by repeating it seamlessly. Key points include:

1. **Tiling**: Tiling involves repeating the texture multiple times over a surface to cover it completely. This is useful for large areas where a single texture would appear stretched or pixelated.
2. **Seamless Textures**: For tiling to look good, textures need to be seamless, meaning there are no visible edges when the texture is repeated.

Shading – Shading in OpenGL

Tiling & Repeating

3. **Repeating UVs**: UV coordinates can be manipulated to repeat a texture. For example, UVs greater than 1 or less than 0 will repeat the texture in a tiling fashion. This allows for a continuous pattern to be created across a surface.
4. **Use Cases**: Common use cases for tiling and repeating include backgrounds, floors, walls, and other large surfaces in 3D environments.

Shading – Shading in OpenGL

Shading techniques in OpenGL play a crucial role in determining the visual realism and quality of rendered scenes by simulating how light interacts with surfaces. Here's how:

1. Types of Shading: OpenGL supports various shading techniques such as flat shading, smooth shading, Gouraud shading, and Phong shading, each offering different levels of realism and computational complexity.

Shading – Shading in OpenGL

2. Process: Shading involves calculating the color of pixels based on the interaction between light sources, surface properties (like reflectivity and texture), and the viewpoint. This process helps in creating smooth transitions of light and shadow across surfaces.

3. Visual Quality: By accurately computing how light reflects off surfaces and interacts with materials, shading techniques enhance the perception of depth, surface details, and overall realism in rendered scenes. Techniques like Phong shading, for example, simulate highlights and smooth transitions between light and shadow, contributing to a more lifelike appearance.

Shading – Texture Mapping

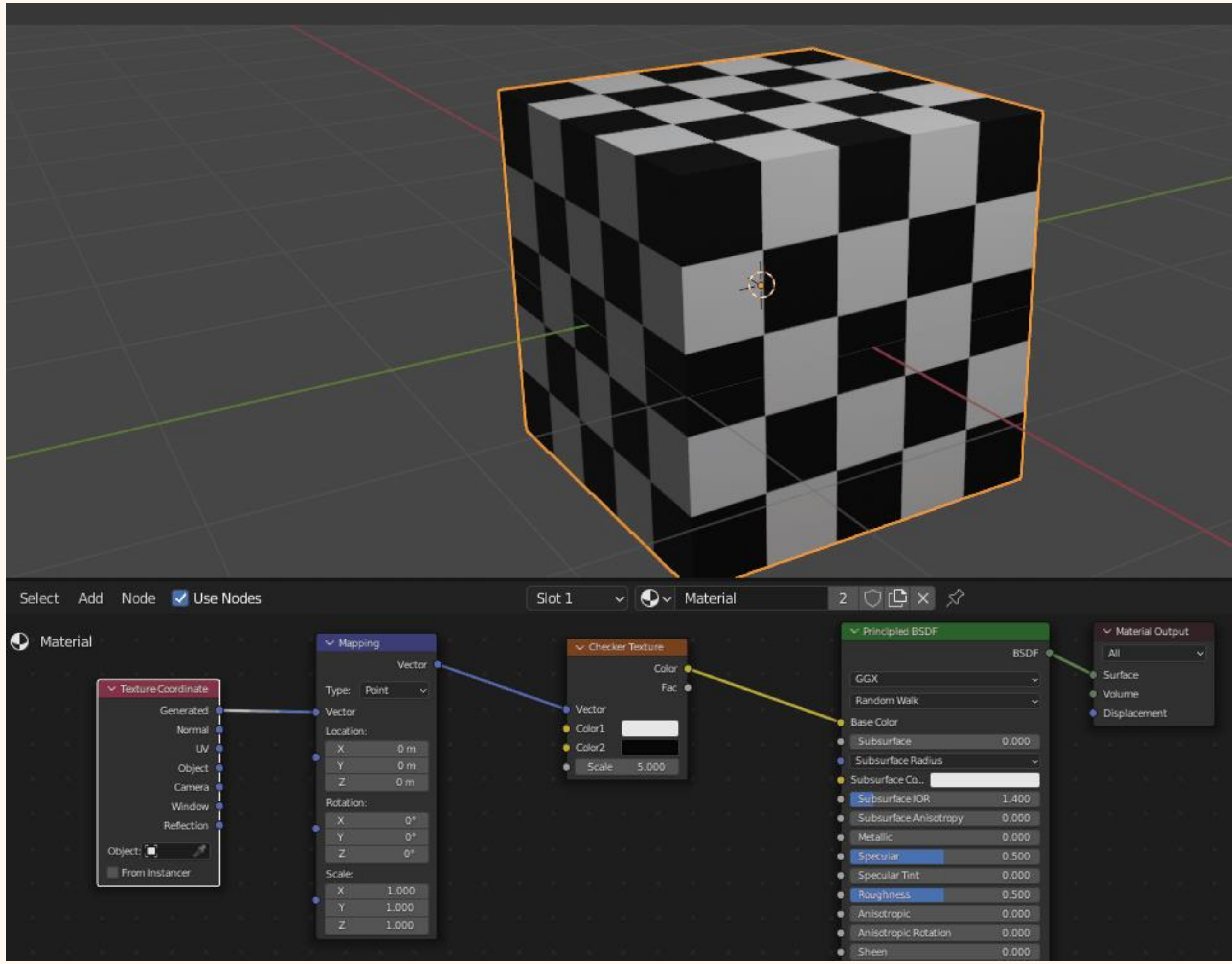
Texture mapping is a technique in computer graphics where a 2D image (texture) is mapped onto the surface of a 3D object. This method enhances the object's visual detail without increasing its geometric complexity, allowing for realistic and intricate surface appearances.

Shading – Texture Mapping

UV Coordinates

UV coordinates are a system for mapping 2D textures onto 3D models. They are used to determine how a texture is wrapped around a model. Key points about UV coordinates include:

1. **UV Mapping**: UV mapping involves assigning coordinates (U, V) to the vertices of the 3D model. These coordinates correspond to points on the 2D texture map.



Mapping — Type: Point
Location: X 0 m, Y 0 m, Z 0 m
Rotation: X 0°, Y 0°, Z 0°
Scale: X 1.000, Y 1.000, Z 1.000

Checker Texture — Scale 5.000

Principled BSDF — GGX, Random Walk
Subsurface 0.000
Subsurface Radius
Subsurface IOR 1.400
Subsurface Anisotropy 0.000
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000

Material Output — All

Shading – Texture Mapping

UV Coordinates

2. **Normalization**: UV coordinates are typically normalized, meaning they range from 0 to 1. This allows the texture to be mapped consistently regardless of its size.
3. **Unwrapping**: The process of creating UV coordinates is often referred to as unwrapping, where the 3D model is laid out in a 2D space to apply the texture accurately.

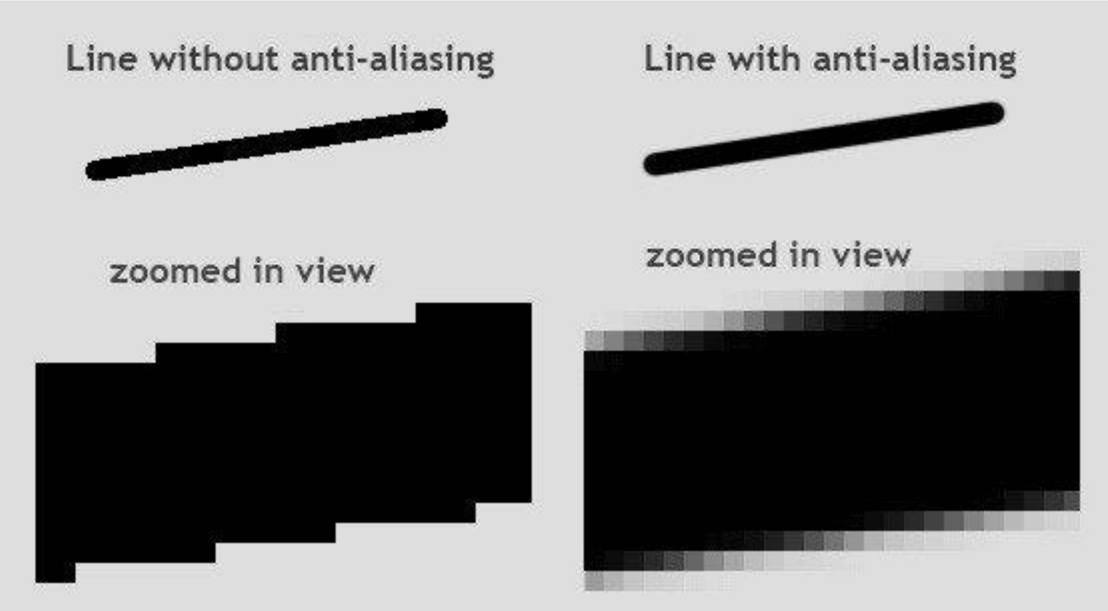
Shading –

Significance of Texture Mapping

1. **Enhanced Realism**: Adds detail to 3D models, making them look more clear.
2. **Efficient Rendering**: Provides detailed visuals, optimizing performance.
3. **Versatility:** Can be used for a variety of effects, including surface color, reflectivity, and bumpiness.

Shading – Aliasing

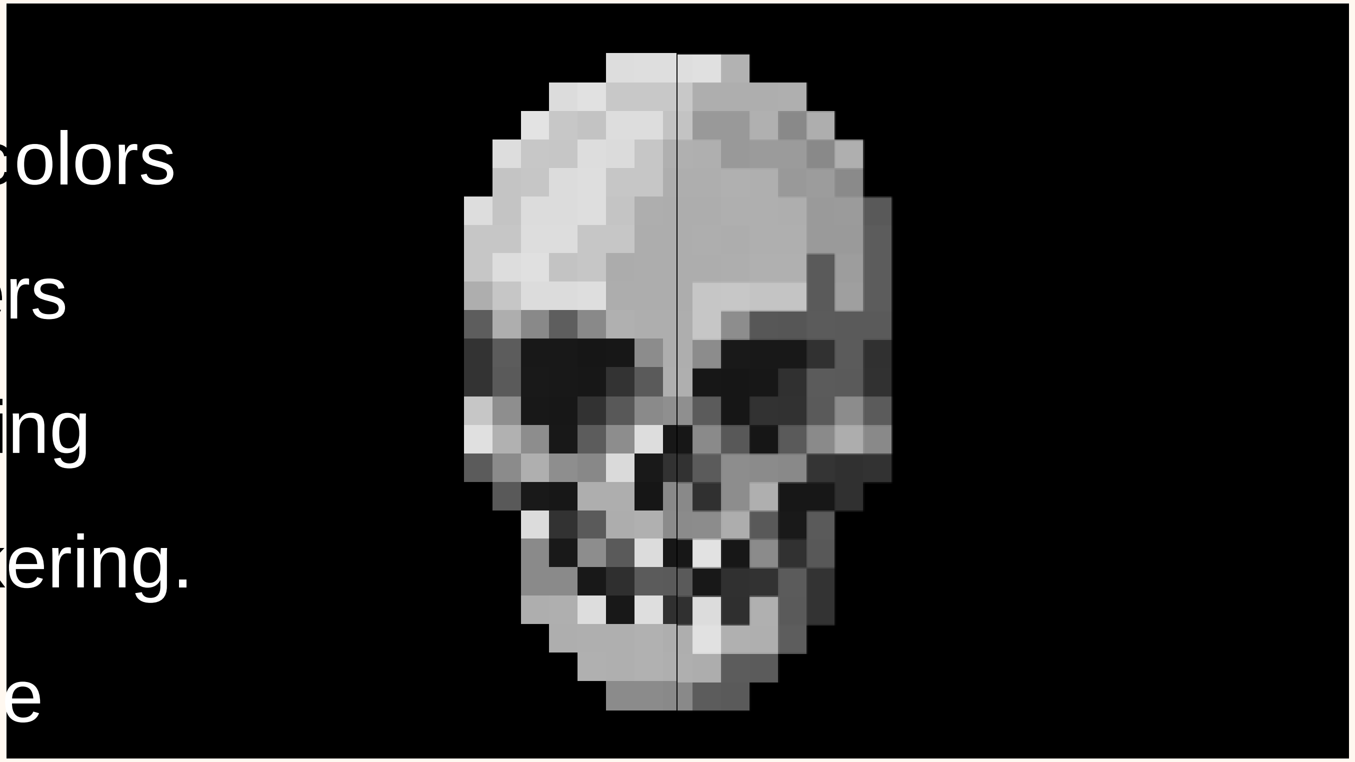
Aliasing in shading happens when there is too much detail in the texture or shape for the screen to show correctly, causing jagged edges or flickering pixels. Anti-aliasing techniques help fix this by smoothing the edges and adjusting the texture.



Shading – Aliasing

Key Points:

1. Shaders are programs that decide the colors of pixels. Aliasing happens when shaders can't properly handle fine details, causing visual glitches like jagged edges or flickering.
2. These include supersampling, where the scene is rendered at a higher resolution and then downsampled, and multisampling, where multiple samples per pixel are used to smooth edges.



Shading – Aliasing

Supersampling: The scene is rendered at a much higher resolution than the display resolution.

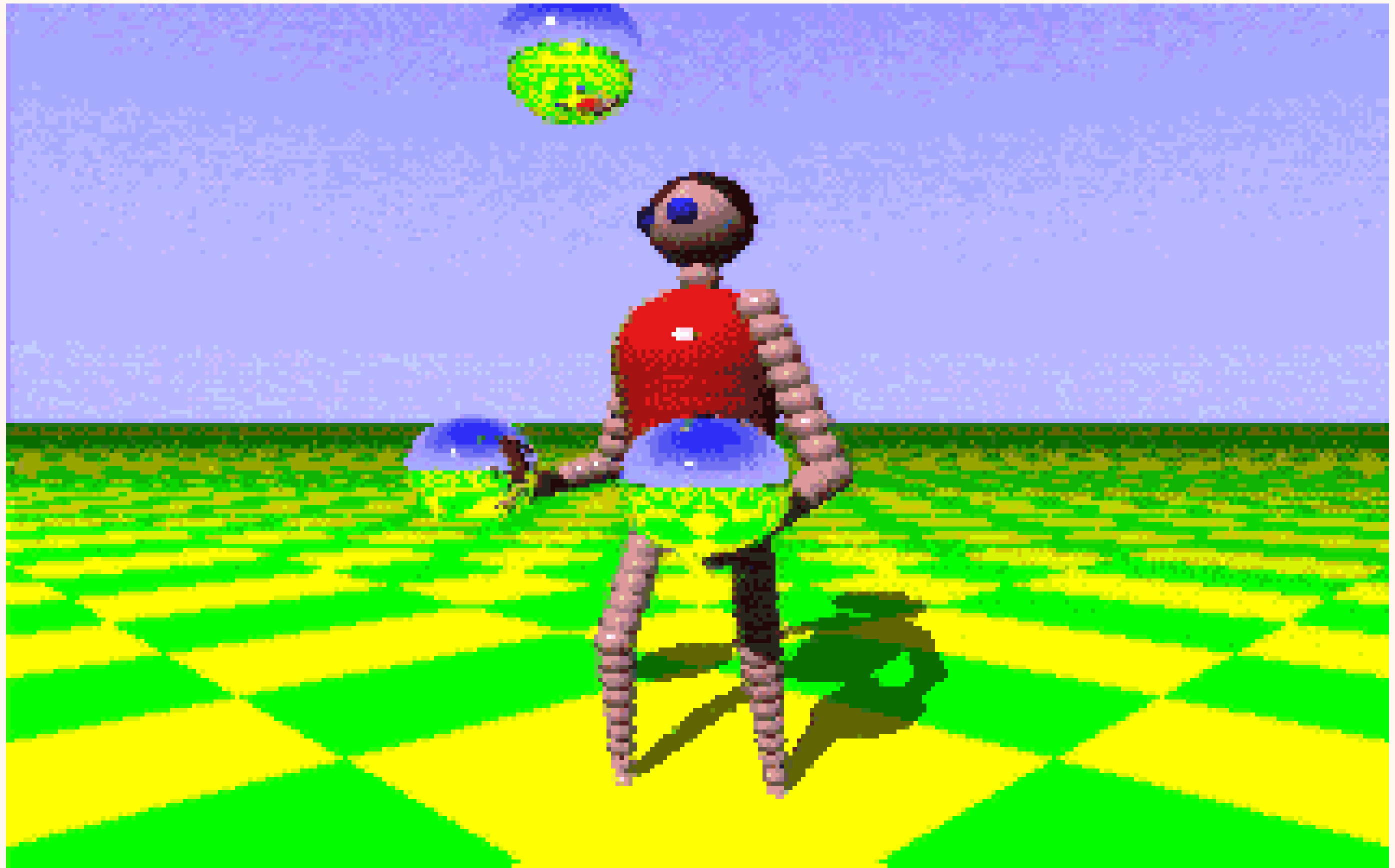
Multisampling: Multisampling Anti-Aliasing (MSAA) involves taking multiple samples per pixel, but it selectively uses these samples primarily on the edges of objects to smooth out jagged edges and reduce aliasing.

Ray Tracing and Path Tracing

1

Ray Tracing

Ray tracing is a rendering technique that simulates the path of light rays as they interact with objects in a 3D scene. It can produce highly realistic images with accurate reflections, refractions, and shadows.



Ray Tracing

Working Principles of Ray Tracing

The core principles include:

- **Ray Casting**: Rays are cast from the viewer's eye (camera) through each pixel on the screen and into the scene to determine which objects are visible.
- **Intersection Tests**: Each ray is tested for intersection with objects in the scene. The closest intersected object determines the visible surface for that pixel.
- **Shading**: Once an intersection is found, shading calculations are performed to determine the color of the pixel.

Ray Tracing

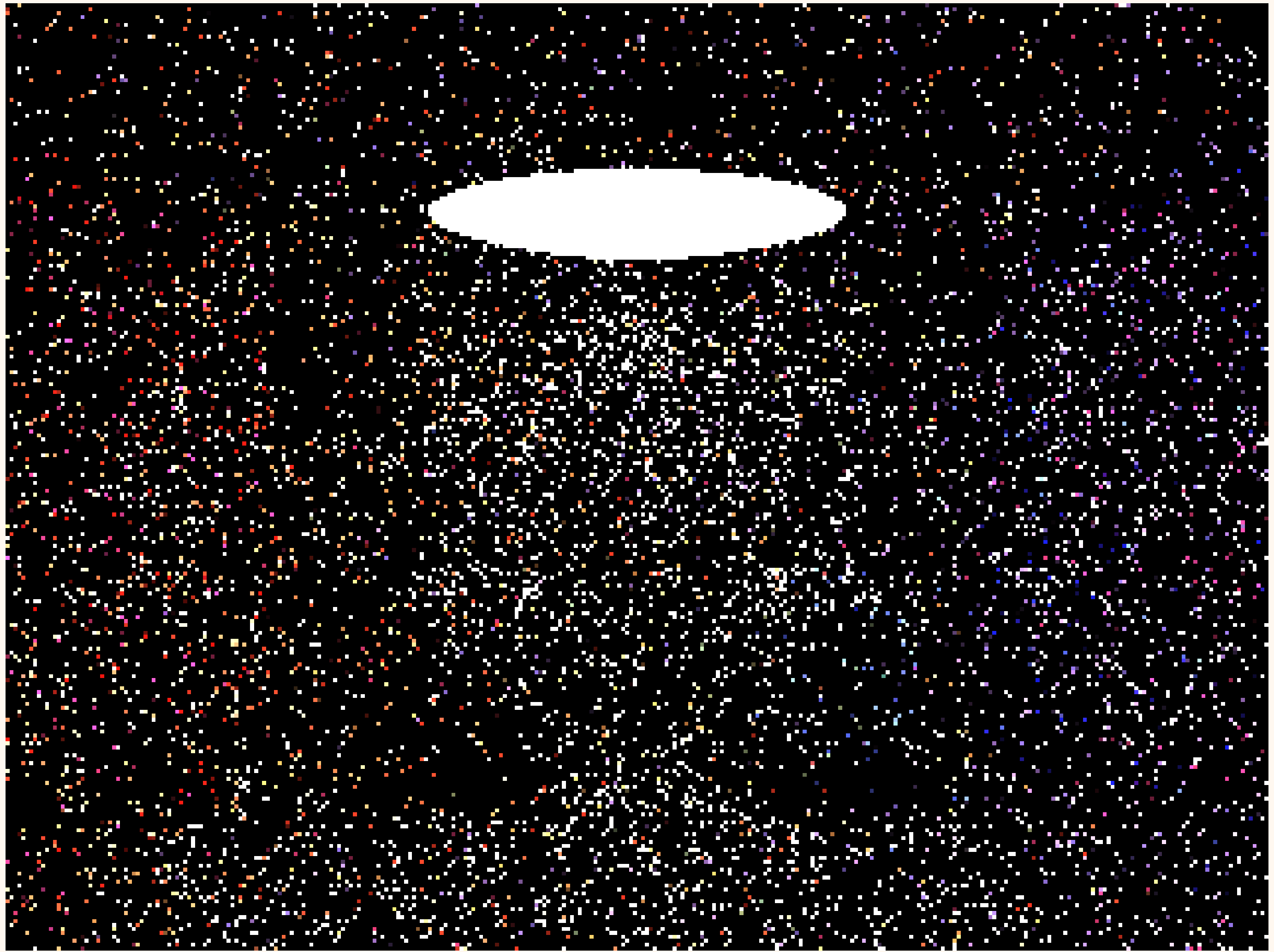
How Ray Tracing is used to create Realistic Images??

1. **Accurate Lighting**: Ray tracing simulates the actual behavior of light, producing realistic lighting effects, including soft shadows, reflections, and refractions.
2. **Detailed Surfaces**: It handles complex surface interactions, allowing for highly detailed textures and materials.
3. **Realistic Visuals**: The technique is extensively used in video games, movies, and architectural visualization to produce photorealistic images and scenes .

2

Path Tracing

Path tracing simulates random light bounces to capture effects like indirect lighting, creating very realistic images but requiring a lot of computational power.



Comparison Between Ray Tracing and Path Tracing

Accuracy

1. **Path Tracing**: Offers higher accuracy in terms of realistic lighting and global illumination due to its comprehensive simulation of light paths and interactions.
2. **Ray Tracing**: Provides accurate reflections and refractions but may lack full global illumination, leading to less realistic images in some scenarios

Comparison Between Ray Tracing and Path Tracing

Computational Complexity

1. **Path Tracing**: Generally more computationally intensive due to the need for numerous samples to reduce noise and achieve a high-quality image. This makes it slower and more resource-demanding.
2. **Ray Tracing**: Less computationally complex compared to path tracing, as it requires fewer samples to produce acceptable results, making it faster and more suitable for real-time applications.