Computer Graphics

Here are the answers to your questions:

1. **What is computer graphics?**
   Computer graphics is the field of computer science that deals with generating, manipulating, and displaying visual content using computers. It includes rendering images, animations, and simulations.
2. **Full forms of the following:**
   - **LED**: Light Emitting Diode
   - **LCD**: Liquid Crystal Display
3. **What is the need for homogeneous coordinates?**
   Homogeneous coordinates are used in computer graphics to facilitate matrix transformations such as translation, rotation, and scaling in a consistent mathematical framework, making calculations simpler and efficient.
4. **What is text clipping?**
   Text clipping is the process of removing portions of text that lie outside a defined clipping region, ensuring that only the visible part of the text is displayed within a designated area.
5. **What are the disadvantages of the DDA algorithm?**
   - It involves floating-point arithmetic, which makes it slower compared to Bresenham's algorithm.
   - Rounding operations can introduce errors in pixel positioning, leading to an approximation.
   - It requires more computations per step, making it less efficient for real-time applications.
6. **Properties of Bezier curve:**
   - It is defined by a set of control points.
   - The curve always passes through the first and last control points.
   - It is smooth and lies within the convex hull of control points.
   - It follows the property of affine invariance.
7. **What is projection?**
   Projection is the process of mapping a 3D object onto a 2D surface (such as a screen or paper) using mathematical transformations. It is used in rendering images in computer graphics.
8. **What is transformation?**
   Transformation in computer graphics refers to the operations that modify the position, size, orientation, or shape of objects. Common transformations include translation, scaling, rotation, and shearing.
9. **Applications of computer graphics:**
   - Video games and entertainment
   - CAD (Computer-Aided Design)
   - Virtual reality and simulations
   - Medical imaging
   - Data visualization

- User interface design

10. **What is shearing?**
Shearing is a transformation that distorts an object by shifting its points in a particular direction, changing its shape while keeping the opposite sides parallel.
11. **Two advantages of computer graphics:**
    - It allows the creation of realistic visual representations for simulations and design.
    - It enhances user interaction in applications such as gaming and virtual reality.
12. **What is a flat panel display?**
A flat panel display is a thin, lightweight screen technology used in devices such as monitors, televisions, and mobile phones. Common types include LCD, LED, and OLED displays.
13. **What is a raster scan display?**
A raster scan display is a type of display system that scans the screen row by row (left to right, top to bottom) to form an image. It is commonly used in CRT, LCD, and LED monitors.
14. **What is 3D viewing transformation?**
3D viewing transformation is the process of converting a 3D scene into a 2D projection by applying transformations such as translation, rotation, scaling, and perspective or parallel projection.

1. What is Computer Graphics? Advantages of Computer Graphics

Definition:
Computer Graphics is the field of computer science that deals with creating, storing, and manipulating visual content such as images, animations, and simulations using computers. It is widely used in gaming, movies, simulations, CAD, and more.

Advantages of Computer Graphics:

- **Visualization**: Helps in visualizing complex data, models, and simulations.
- **Interactivity**: Enables user interaction in applications like gaming and UI design.
- **Efficiency**: Speeds up design processes in fields like architecture and engineering.
- **Realism**: Generates photorealistic images and animations for entertainment and virtual reality.
- **Flexibility**: Allows modifications and transformations of images easily.

---

2. Explain the Working of CRT with a Suitable Diagram

A **Cathode Ray Tube (CRT)** is a display device used in older television and computer monitors. It works by firing electrons onto a phosphor-coated screen, which glows to produce an image.

Working:

1. **Electron Gun:** Generates a beam of electrons.
2. **Control Grids:** Regulate the intensity of the electron beam.
3. **Deflection System:** Uses magnetic or electric fields to direct the beam to specific screen positions.
4. **Phosphor Screen:** Emits light when struck by electrons, creating an image.
5. **Refresh Mechanism:** The process repeats at high speed (refresh rate) to maintain a steady image.

📌 **Diagram:**
(A simple CRT diagram should be included here, showing the electron gun, deflection plates, and screen.)

---

3. Differences Between Random Scan Display and Raster Scan Display

Feature	Random Scan Display	Raster Scan Display
Working	Uses a beam to draw images directly by connecting line segments.	Uses a raster (grid) of pixels to display an image.
Image Formation	Draws only necessary lines (vector-based).	Displays the whole image by scanning pixel by pixel.
Refresh Rate	Refreshes only when required.	Refreshes the entire screen many times per second.
Resolution	High resolution for line drawings.	Limited by pixel density.
Used In	CAD, vector-based applications.	TVs, modern computer monitors, gaming.
Speed	Fast for simple graphics, slow for complex images.	Can handle complex images easily.

---

4. Short Note on LED / LCD

Light Emitting Diodes (LEDs)

- **Definition:** LEDs are semiconductor devices that emit light when an electric current passes through them.
- **Working:** When current flows through a diode, electrons recombine with holes, releasing energy in the form of light.
- **Advantages:**
    - High brightness and efficiency.
    - Long lifespan.
    - Low power consumption.
    - Used in modern TVs, traffic lights, and display panels.

OR

Liquid Crystal Display (LCD)

- **Definition:** LCDs use liquid crystals that change their orientation to control light passage and display images.
- **Working:**
    1. **Backlight:** A light source illuminates the liquid crystals.
    2. **Liquid Crystals:** Change orientation to allow/block light, forming an image.
    3. **Color Filters:** Create colored pixels for display.
- **Advantages:**
    - Low power consumption.
    - Thin and lightweight compared to CRTs.
    - Used in TVs, monitors, and mobile screens.

---

5. Explain the Vector Scan / Random Scan Display

Definition:
A **Vector Scan Display** (or **Random Scan Display**) is a type of display system that draws images by directly controlling the electron beam to follow the shape of the objects rather than scanning the entire screen.

Working:

1. The system moves the electron beam only to the parts of the screen where drawing is needed.
2. It uses **line-drawing algorithms** to create images by connecting points.
3. The screen refreshes only when needed (not at a fixed rate like raster scan).
4. It is mainly used for **line-drawing applications** such as CAD and engineering simulations.

Advantages:
✔️ High resolution for line drawings.
✔️ Less memory required as only lines are stored.

Disadvantages:
❌ Not suitable for complex images with shading and colors.
❌ More expensive and slower than raster scan for large images.

📌 **Diagram:**
(A simple diagram showing how a vector scan display draws images using line segments should be included.)

Let me know if you need more details or diagrams! 🚀

6. Short Note on Flat Panel Display

A **Flat Panel Display (FPD)** is a thin, lightweight display technology used in modern electronic devices such as laptops, TVs, and smartphones. Unlike bulky CRT displays, FPDs use advanced technology for a slimmer design and better image quality.

Types of Flat Panel Displays:

- **LCD (Liquid Crystal Display):** Uses liquid crystals and a backlight to produce images.
- **LED (Light Emitting Diode):** Uses light-emitting diodes for brighter and energy-efficient displays.
- **OLED (Organic LED):** Uses organic compounds that emit light when an electric current passes through, offering better contrast and flexibility.

Advantages:
✔️ Thin and lightweight.
✔️ Low power consumption.
✔️ High resolution and brightness.

---

7. Short Note on Boundary Fill Algorithm OR Bresenham's Line Drawing Algorithm

Boundary Fill Algorithm:

The **Boundary Fill Algorithm** is a recursive algorithm used to fill a region enclosed by a boundary of a particular color. It is commonly used in paint programs.

Steps:

1. Select a starting point inside the boundary.
2. Check if the pixel is the boundary color or the fill color.
3. If not, fill it with the desired color.
4. Recursively apply the algorithm to neighboring pixels (4-connected or 8-connected).

Limitations:

- Slow for large areas.
- May cause stack overflow due to deep recursion.

OR

Bresenham's Line Drawing Algorithm:

Bresenham's algorithm is an efficient method to draw a straight line between two points using only integer calculations.

Steps:

1. Compute the difference in x and y coordinates (Δx, Δy).
2. Determine the decision parameter **p** to select the next pixel.
3. If **p < 0**, move horizontally; otherwise, move diagonally.
4. Update **p** and repeat until the endpoint is reached.

Advantages:
✔️ Efficient integer calculations.
✔️ No floating-point arithmetic.
✔️ Faster than DDA (Digital Differential Analyzer).

---

8. Midpoint Circle Drawing Algorithm OR Midpoint Ellipse Drawing Algorithm

Midpoint Circle Drawing Algorithm:

This algorithm is used to draw a circle using **midpoint calculations** instead of floating-point arithmetic.

Steps:

1. Start from the topmost point of the circle **(0, r)**.
2. Use the **decision parameter (p)** to decide the next pixel.
3. If **p < 0**, move to the right; otherwise, move diagonally.
4. Repeat for 1/8th of the circle and mirror the points for symmetry.

📌 **Diagram:** *(A diagram showing pixel placements in one quadrant should be included.)*

OR

Midpoint Ellipse Drawing Algorithm:

This algorithm is used to draw ellipses efficiently using integer calculations.

Steps:

1. Start at the point **(0, b)**.
2. Use decision parameter **p1** for Region 1 (steep part) and **p2** for Region 2 (flat part).
3. Update decision parameters and choose the next pixel accordingly.
4. Use symmetry to complete the ellipse.

---

9. Scan Line Polygon Filling Algorithm

The **Scan Line Polygon Filling Algorithm** is used to fill a polygon by checking intersections of scan lines with the polygon edges and filling pixels between pairs of intersections.

Steps:

1. Find the **minimum and maximum y-coordinates** of the polygon.
2. Draw horizontal scan lines from **y_min to y_max**.
3. Determine **intersection points** of the scan line with polygon edges.
4. Sort the intersection points by x-coordinates.
5. Fill pixels **between pairs of intersection points**.

📌 **Diagram:** *(A diagram showing a polygon being filled with scan lines should be included.)*

Advantages:
✔️ Efficient for filling polygons with complex shapes.
✔️ Works well with convex and concave polygons.

---

10. DDA Line Drawing Algorithm

The **Digital Differential Analyzer (DDA) Algorithm** is a simple method to draw a line by incrementing one coordinate at a constant rate.

Steps:

1. Compute the differences **Δx = x2 - x1** and **Δy = y2 - y1**.
2. Determine the number of steps as **steps = max(|Δx|, |Δy|)**.
3. Compute **X_increment = Δx / steps** and **Y_increment = Δy / steps**.
4. Start from **(x1, y1)** and increment by **X_increment, Y_increment** until reaching **(x2, y2)**.

Limitations:

- Uses floating-point calculations, making it slower.
- Not as efficient as Bresenham's Algorithm.

Let me know if you need diagrams or further explanations! 🚀

11. Flood Fill Algorithm

The **Flood Fill Algorithm** is used to fill a connected region with a specific color, similar to the **paint bucket tool** in image editing software. It is widely used in computer graphics for filling closed shapes.

Algorithm Steps:

1. Start from a given pixel **(x, y)** inside the region.
2. Check if the current pixel has the boundary color or the fill color.
3. If not, replace the current pixel's color with the new color.
4. Recursively apply the algorithm to its **neighboring pixels** (4-connected or 8-connected).

📌 **Types of Flood Fill:**

- **4-Connected Flood Fill:** Fills in four directions (up, down, left, right).
- **8-Connected Flood Fill:** Fills in eight directions (includes diagonals).

Limitations:

- **Stack overflow** for large regions (recursive calls).
- **Slow** compared to scan-line filling algorithms.

---

12. Short Note on 2D Rotation OR 2D Reflection

2D Rotation

Rotation is a transformation that **rotates a point or an object** about a fixed pivot point (usually the origin).

📌 **Rotation Formula:**
For rotating a point **(x, y)** by an angle **θ** around the origin:

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

✔️ Counterclockwise rotation → **Positive θ**
✔️ Clockwise rotation → **Negative θ**

2D Reflection

Reflection is a transformation that **mirrors an object** across a specified axis.

📌 **Reflection Matrices:**

- Reflection about **X-axis:**

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Reflection about **Y-axis:**

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Reflection about **Origin:**

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

✔️ Used in **mirror imaging, CAD applications, and animation effects**.

---

13. What is Homogeneous Coordinate? Why is it Required?

Homogeneous Coordinates are an **extended coordinate system** used in computer graphics to simplify transformations like translation, scaling, and rotation using matrix multiplication.

📌 **Homogeneous Representation:**
A 2D point **(x, y)** is represented as **(x, y, 1)** in homogeneous coordinates.

📌 **Why is it Required?**

- Allows **all transformations (translation, rotation, scaling)** to be represented using **matrix multiplication**.
- Makes it easier to **combine multiple transformations** into a single matrix.
- Used in **3D graphics, perspective transformations, and projections**.

---

14. Liang-Barsky Line Clipping Algorithm

The **Liang-Barsky Algorithm** is an efficient method for **line clipping**, which determines the portion of a line inside a rectangular clipping window.

Advantages Over Cohen-Sutherland:
✔️ Faster because it uses **parametric equations** instead of multiple region codes.
✔️ Reduces unnecessary calculations by directly computing **intersection points**.

📌 **Algorithm Steps:**

1. Represent the line using parametric equations:

$$x = x_1 + t(x_2 - x_1)$$

$$y = y_1 + t(y_2 - y_1)$$

2. Compute the **entering** and **leaving** values of **t** by checking against the clipping window.
3. Clip the line segment by adjusting the starting and ending points based on **t_min** and **t_max**.
4. If **t_min ≤ t_max**, draw the clipped line; otherwise, discard it.

✔️ **More efficient than Cohen-Sutherland for large datasets**.

---

15. Cohen-Sutherland Line Clipping Algorithm OR Nicholl-Lee-Nicholl Line Clipping Algorithm

Cohen-Sutherland Line Clipping Algorithm

The **Cohen-Sutherland Algorithm** is used to clip lines that extend beyond a rectangular clipping window.

📌 **Steps:**

1. Assign a **4-bit region code** to each endpoint:
   - **Bit 1:** Above the window.
   - **Bit 2:** Below the window.
   - **Bit 3:** Right of the window.
   - **Bit 4:** Left of the window.
2. Check conditions:
   - If both endpoints **have code 0000** → Line is **completely inside** (accept it).
   - If **logical AND** of both codes is **not 0000** → Line is **completely outside** (reject it).
   - Otherwise, **calculate intersection points** and clip accordingly.

✔ **Widely used due to its simplicity but slower than Liang-Barsky**.

Nicholl-Lee-Nicholl (NLN) Line Clipping Algorithm

The **Nicholl-Lee-Nicholl Algorithm** is an optimized line clipping algorithm that **avoids repetitive intersection calculations**.

📌 **Key Features:**

- Faster than Cohen-Sutherland because it classifies lines into **trivial accept, trivial reject, or clipping cases**.
- Reduces the number of **intersection calculations** required.
- Works best when **lines are mostly inside or outside the window**.

✔ **More efficient than Cohen-Sutherland for real-time applications**.

Let me know if you need further explanations or diagrams! 🚀

15. Cohen-Sutherland Line Clipping Algorithm OR Nicholl-Lee-Nicholl Line Clipping Algorithm

Cohen-Sutherland Line Clipping Algorithm

The **Cohen-Sutherland Algorithm** is a widely used method for clipping lines against a rectangular window. It assigns region codes to endpoints and efficiently determines if a line should be clipped, accepted, or rejected.

📌 **Algorithm Steps:**

1. **Assign a 4-bit region code** to each endpoint using the clipping window boundaries.
   - **Bit 1:** Above the window.
   - **Bit 2:** Below the window.
   - **Bit 3:** Right of the window.

- **Bit 4:** Left of the window.

2. **Check region codes:**
   - If **both endpoints have code 0000**, the line is **completely inside** (accepted).
   - If the **logical AND of both codes ≠ 0000**, the line is **completely outside** (rejected).
   - Otherwise, the line is **partially inside**, so calculate the intersection points and clip the line accordingly.

✔️ **Widely used but slower than Liang-Barsky for large datasets.**

Nicholl-Lee-Nicholl (NLN) Line Clipping Algorithm

The **Nicholl-Lee-Nicholl Algorithm** is an optimized line clipping algorithm that avoids unnecessary intersection calculations.

📌 **Key Features:**

- Classifies lines into **trivial accept, trivial reject, or clipping cases**.
- Reduces the number of **intersection calculations** required.
- Works best when **lines are mostly inside or outside the window**.

✔️ **More efficient than Cohen-Sutherland for real-time applications.**

---

16. Window-to-Viewport Transformation OR Clipping Operations

Window-to-Viewport Transformation

The **window-to-viewport transformation** is used to map a part of the world coordinate system (window) onto a screen coordinate system (viewport).

📌 **Transformation Formula:**

$$X_v = X_vmin + \left(\frac{(X_w - X_wmin)}{(X_wmax - X_wmin)}\right)(X_vmax - X_vmin)$$

$$Y_v = Y_vmin + \left(\frac{(Y_w - Y_wmin)}{(Y_wmax - Y_wmin)}\right)(Y_vmax - Y_vmin)$$

where:

- (X_w, Y_w) = Window coordinates
- (X_v, Y_v) = Viewport coordinates
- $(X_wmin, Y_wmin, X_wmax, Y_wmax)$ = Window boundaries
- $(X_vmin, Y_vmin, X_vmax, Y_vmax)$ = Viewport boundaries

✔️ **Used in computer graphics to scale world objects to fit different screen sizes.**

---

17. 2D Transformations OR Shearing

2D Transformations

2D transformations allow objects to be manipulated in space.

📌 **Types of 2D Transformations:**

1. **Translation:** Moves an object by adding displacement (dx, dy).

$$x' = x + dx, \quad y' = y + dy$$

2. **Scaling:** Resizes an object.

$$x' = x \cdot S_x, \quad y' = y \cdot S_y$$

3. **Rotation:** Rotates an object around the origin.

$$x' = x\cos\theta - y\sin\theta, \quad y' = x\sin\theta + y\cos\theta$$

4. **Shearing:** Skews the shape of an object.

Shearing (with Example)

Shearing transforms an object such that **one axis is shifted while the other remains fixed.**

📌 **Shearing Formula:**

- **X-Shear:**

$$x' = x + sh_x \cdot y, \quad y' = y$$

- **Y-Shear:**

$$x' = x, \quad y' = y + sh_y \cdot x$$

✔ **Example:** A square can be sheared into a parallelogram by applying a shear factor.

---

18. Bezier Curve OR Parametric and Geometric Continuity OR Hermite Interpolation

Bezier Curve

A **Bezier Curve** is a parametric curve commonly used in computer graphics for smooth and flexible curve modeling.

📌 **Bezier Curve Equation (for 3 control points):**

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \quad 0 \le t \le 1$$

where **P0, P1, P2** are control points.

✔ **Properties of Bezier Curves:**

- Starts at **P0** and ends at **Pn**.
- The shape of the curve depends on **control points**.
- The curve is always inside the **convex hull** of control points.

✔️ **Applications:**

- Used in **vector graphics, animation, and font design**.

---

19. Basic Geometry Properties of B-Spline Curve OR Applications of Parametric Cubic Curve

Basic Properties of B-Spline Curve:

- **Piecewise-defined:** Composed of multiple polynomial segments.
- **Degree:** A B-spline of degree n has **n+1 control points**.
- **Local control:** Moving one control point only affects a small portion of the curve.

✔️ **Used in CAD modeling and 3D graphics.**

---

20. Short Note on Depth Cueing

Depth Cueing is a shading technique used in 3D graphics to enhance depth perception by adjusting object brightness based on distance.

📌 **How It Works:**

- **Near objects → Brighter**
- **Far objects → Darker**

✔️ **Used in 3D rendering, gaming, and virtual reality to simulate real-world depth.**

Let me know if you need detailed explanations or diagrams! 🚀

21. Interpolation Spline vs. Approximation Spline OR Parallel vs. Perspective Projection

Interpolation Spline vs. Approximation Spline

Feature	Interpolation Spline	Approximation Spline
Definition	Passes through all given control points.	Does not pass through control points but stays near them.
Usage	Used for **data fitting** where exact points must be reached.	Used for **curve smoothing** in CAD, animation, etc.
Mathematical Basis	Uses Lagrange, Bezier, or B-Spline methods.	Uses Bezier curves or B-Splines to approximate shapes.
Control	More rigid; hard to adjust without affecting all points.	More flexible; control points influence overall shape but not strictly followed.

✔️ **Example:**

- **Interpolation:** Used in **digitized handwriting recognition**.
- **Approximation:** Used in **3D modeling and font design**.

Parallel Projection vs. Perspective Projection

Feature	Parallel Projection	Perspective Projection
Definition	Projecting objects onto a plane without considering depth.	Objects appear smaller as they move farther from the viewer.
Realism	Less realistic, used in technical drawings.	More realistic, used in 3D games and simulations.
Lines Convergence	Parallel lines remain parallel.	Parallel lines converge at a vanishing point.
Types	Orthographic, Oblique, Isometric projections.	One-point, Two-point, Three-point perspective.

✔ **Example:**

- **Parallel Projection: Architectural and engineering drawings.**
- **Perspective Projection: 3D rendering in movies and games.**

---

22. 3D Viewing Transformation OR 3D Translation OR 3D Reflection

3D Viewing Transformation

3D viewing transformation is the process of converting **3D world coordinates** into **2D screen coordinates**.

📌 **Steps:**

1. **Modeling Transformation:** Place objects in the world.
2. **Viewing Transformation:** Define the camera/viewer position.
3. **Projection Transformation:** Convert 3D objects into a 2D view.
4. **Clipping & Viewport Mapping:** Remove unseen parts and map the visible portion to the display.

✔ **Used in 3D rendering pipelines (OpenGL, DirectX).**

3D Translation

Translation moves an object in **3D space** by shifting it along **x, y, and z axes**.

📌 **Matrix Representation:**

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

✔ **Example:** Moving a **car in a 3D game**.

3D Reflection

Reflection flips an object across a plane (XY, YZ, or XZ).

📌 **Matrix for Reflection Across XY-plane:**

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

✔️ **Example:** Used in **mirror effects in games and simulations**.

---

23. What is Projection? List Out Various Types of Projection

Definition

Projection is the method of **mapping 3D objects onto a 2D plane**.

📌 **Types of Projections:**

1. **Parallel Projection**
   - **Orthographic Projection** (Top, Front, Side views)
   - **Oblique Projection** (Cabinet, Cavalier)
   - **Isometric Projection** (Equal angles)
2. **Perspective Projection**
   - **One-point Perspective** (Single vanishing point)
   - **Two-point Perspective** (Most common, used in games)
   - **Three-point Perspective** (For extreme perspectives)

✔️ **Applications:**

- **Engineering Drawings** → Parallel Projection
- **3D Video Games** → Perspective Projection

---

24. Scaling in 3D Transformations

Definition

Scaling resizes an object in **3D space** by modifying its **x, y, and z** dimensions.

📌 **Scaling Formula:**

$$x' = S_x \cdot x, \quad y' = S_y \cdot y, \quad z' = S_z \cdot z$$

📌 **Scaling Matrix:**

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

✔️ **Uniform Scaling:** When **Sx = Sy = Sz**, size changes but shape remains the same.
✔️ **Non-Uniform Scaling:** When **Sx ≠ Sy ≠ Sz**, different dimensions are scaled differently.

---

25. Comparison: Perspective vs. Parallel Projection & Types of Parallel Projection

Comparison Table: Perspective vs. Parallel Projection

Feature	Perspective Projection	Parallel Projection
Depth Effect	Objects appear smaller as they move farther.	Objects remain the same size, no depth effect.
Realism	More realistic, used in **3D games and animations**.	Less realistic, used in **technical drawings**.
Lines Convergence	Parallel lines meet at a vanishing point.	Parallel lines remain parallel.
Application	Used in **movies, 3D modeling, and games**.	Used in **architecture, engineering, and CAD drawings**.

Types of Parallel Projections

📌 1. Orthographic Projection

- No distortion, used in **technical drawings**.
- Subtypes: **Front View, Top View, Side View**.

📌 2. Oblique Projection

- Projected at an angle to show depth.
- **Cavalier Projection** (full depth) vs. **Cabinet Projection** (half-depth).

📌 3. Isometric Projection

- Angles between axes are **120°**, making it appear 3D.
- Used in **game design, engineering graphics**.

✔️ Example:

- **Isometric Projection** is used in **SimCity** and **old-school RPGs**.
- **Orthographic Projection** is used in **blueprints and technical designs**.

That covers all the topics with definitions, formulas, and practical applications. Let me know if you need explanations with diagrams! 🚀