**CSC 320 REVISIONS**

**Differentiate the following terms.**

**[4Marks]**

1. **Computer Graphics and Image Processing**

Computer Graphics vs. Image Processing:

Computer Graphics: Computer graphics focuses on the creation, manipulation, and rendering of images using computers. It involves generating images from geometric models and rendering them to produce realistic or stylized visuals. Examples include 3D modeling, rendering, animation, and interactive graphics in video games or virtual reality.

Image Processing Image processing deals with manipulating existing images to enhance or extract information from them. It involves techniques like filtering, enhancement, compression, and pattern recognition applied to images. Examples include medical image analysis, satellite image processing, and digital photography editing.

1. **Simulation and Animation**

Simulation vs. Animation:

Simulation: Simulation involves creating a model of a real-world system or process and running experiments or scenarios on that model to understand its behavior or predict outcomes. It is used in various fields like engineering, economics, and science for training, analysis, or decision-making support.

Animation: Animation involves creating moving images or visuals by displaying a sequence of static images or frames in rapid succession. It is used for entertainment, education, visualization, and storytelling purposes. Examples include animated films, cartoons, and motion graphics.

**Computer Graphics is a field in computer science that is gaining fame day by day, using relevant examples, explain why this is so. [6Marks]**

1. **Advancements in Technology:** Continuous advancements in hardware capabilities, such as faster processors, better graphics cards, and increased memory, enable more complex and realistic graphics to be generated in real-time.

2. **Increasing Demand:** With the growth of industries like gaming, virtual reality, augmented reality, and visual effects in movies, there's a higher demand for skilled professionals in computer graphics to create immersive and engaging experiences.

3. **Applications in Various Fields:** Computer graphics finds applications not only in entertainment but also in areas like scientific visualization, data analysis, education, architecture, and product design, leading to a broader range of career opportunities.

4. **Accessibility of Tools:** The availability of powerful and user-friendly software tools for computer graphics, such as Adobe Creative Suite, Blender, Unity, and Unreal Engine, has lowered the barrier to entry for enthusiasts and professionals alike.

**The higher the resolution, the better the quality of pictures. What effects does high resolution have on pixels? [2Marks]**

Effects of high resolution on pixels:

- High resolution means that there are more pixels packed into a given area, resulting in finer detail and sharper images.

- With higher resolution, individual pixels become smaller, leading to smoother curves and edges in images.

- High-resolution images can display more information and nuances, making them visually appealing and suitable for detailed analysis or printing in large formats.

- However, extremely high resolutions may not always be necessary or practical, as they can lead to larger file sizes and increased computational requirements for processing and displaying the images.

**Computer Graphics borrows many concepts from different scientific disciplines, explain relationship between computer graphics and other 3 disciplines. [6Marks]**

Computer graphics is an interdisciplinary field that draws concepts and techniques from various scientific disciplines to create and manipulate digital images. Here's how computer graphics relates to three specific disciplines:

**1. Mathematics:**

- Computer graphics heavily relies on mathematical principles, particularly geometry, linear algebra, and calculus.

- Geometry is essential for defining the shapes and positions of objects in a virtual space. Concepts like vectors, matrices, transformations, and coordinate systems are fundamental in representing and manipulating 3D objects.

- Linear algebra is used extensively in tasks such as projection, rotation, scaling, and interpolation of objects in a scene.

- Calculus comes into play for tasks like surface shading, lighting calculations, and curve fitting.

**2. Physics:**

- Physics concepts are crucial in simulating realistic behaviors of objects and phenomena in computer-generated environments.

- Mechanics principles govern the motion, collision, and deformation of objects in animations and simulations.

- Optics principles are used to simulate how light interacts with surfaces, leading to effects like reflection, refraction, and diffraction.

- Fluid dynamics simulations are employed for realistic rendering of water, smoke, fire, and other fluid-based phenomena.

- Thermodynamics principles are applied in simulating heat transfer and material properties.

**3. Psychology and Cognitive Science:**

- Computer graphics aims to create visually appealing and engaging experiences for users, which involves understanding human perception and cognition.

- Principles from psychology, such as color theory, visual perception, and Gestalt principles, are applied to design interfaces, animations, and visualizations that are intuitive and effective in conveying information.

- Human-computer interaction (HCI) research contributes to the development of techniques for user-driven navigation, manipulation, and interaction with 3D graphical interfaces.

- Virtual reality (VR) and augmented reality (AR) technologies heavily rely on understanding human perception and cognition to create immersive and believable virtual experiences.

**CRT is one of the graphic devices, explain how it displays images on the screen using a well labelled diagram. [4Marks]**

A CRT monitor works by using a beam of electrons emitted from a cathode to create images on a phosphorescent screen. Here's how it displays images:

**1. Electron Beam Generation:**

- The CRT monitor contains an electron gun, which emits a focused beam of electrons when a voltage is applied to it.

- The electron beam passes through a series of focusing and deflection coils, which shape and direct it towards the screen.

**2. Screen Coating:**

- The front of the CRT screen is coated with a layer of phosphorescent material. This material emits light when struck by the electron beam.

**3. Pixel Illumination:**

- The electron beam scans across the screen horizontally, from left to right, line by line. This scanning process is called raster scanning.

- As the electron beam moves across the screen, it energizes the phosphorescent material at each point it hits, causing it to emit light.

- By controlling the intensity of the electron beam and the rate of scanning, different levels of brightness and colors can be achieved.

**4. Refreshing the Image:**

- After scanning one line, the electron beam moves back to the left side of the screen and slightly down to start scanning the next line.

- This process repeats for each line of the screen until the entire image is displayed.

- The entire screen is refreshed multiple times per second (usually 60 times per second or more) to maintain a stable image.

**Explain why C language is one of the most preferred programming languages in computer graphics. [4Marks]**

**1. Efficiency**: C is a low-level language that allows direct manipulation of memory and hardware, making it efficient for graphics programming where performance is critical.

**2. Access to Hardware:** C provides access to hardware components and system resources, allowing programmers to control graphics devices directly and optimize performance.

**3. Portability:** C code can be easily ported across different platforms and operating systems, making it suitable for developing graphics applications that need to run on diverse environments.

**4. Rich Library Support:** C has a rich set of libraries and frameworks specifically designed for graphics programming, such as OpenGL and DirectX, providing developers with powerful tools and functionalities for creating advanced graphics applications.

**5. Close to the Metal:** C allows developers to work closely with the underlying hardware, making it easier to implement complex algorithms and optimizations for rendering graphics efficiently.

**6. Legacy Support:** Many existing graphics applications and libraries are written in C, and there is a vast amount of documentation, tutorials, and resources available for graphics programming in C, making it easier for developers to learn and work with.

**Explain why closegraph () method must be invoked after display in C.**

In C graphics programming, `closegraph () ` is a function used to release resources and close the graphics mode that was initiated by the `initgraph () ` function. Here's why `closegraph () ` must be invoked after displaying graphics:

**1. Resource Cleanup:** When `initgraph()` is called to initialize the graphics mode, it allocates resources such as memory buffers, drawing surfaces, and hardware settings. These resources are used to render graphics on the screen.

**2. Preventing Resource Leaks:** If `closegraph()` is not called after displaying graphics, these resources may not be released properly when the program terminates. This can lead to memory leaks and other resource-related issues, as the resources allocated by `initgraph()` will remain in use even after the program has finished executing.

**3. Restoring Previous State:** `closegraph()` restores the system to its previous state before entering the graphics mode. This includes restoring the text mode, resetting display settings, and releasing any resources allocated during graphics mode.

**4. Compatibility:** Some graphics drivers and systems may require `closegraph()` to be called explicitly to ensure proper cleanup and compatibility with the underlying hardware and operating system.

**Determine the new position of a point(x,y) when moved:**

1. **To a point which is at a distance of Tx along x axis**

add Tx to the current x-coordinate:

New x-coordinate = x + Tx

The y-coordinate remains unchanged: New y-coordinate = y

**ii) To a point which is at a distance of Ty along y axis**

add Ty to the current y-coordinate:

New y-coordinate = y + Ty

The x-coordinate remains unchanged: New x-coordinate = x

**QUESTION TWO**

**Define the following terms.**

1. **Persistence**

**Persistence** refers to the ability of a visual display to maintain an image after the source of the image has been removed. In the context of display technologies, persistence is often discussed in relation to cathode ray tube (CRT) monitors. After the electron beam in a CRT monitor scans a point on the screen and illuminates phosphors to create an image, those phosphors continue to emit light for a short period, even after the electron beam has moved on to scan other points. This lingering emission of light is known as persistence. It's measured in milliseconds, indicating how long the phosphors emit light after being excited by the electron beam. Persistence affects the quality of motion portrayal and can cause motion blur if it's too high.

1. **DVST**

**DVST (Direct-View Storage Tube):**

A Direct-View Storage Tube (DVST) is a type of display device used in early computer systems for both display and storage of information. It's based on the cathode ray tube (CRT) technology. Unlike conventional CRT monitors that require constant refreshing to maintain an image, a DVST can retain an image indefinitely without the need for refreshing. This is achieved by using a special phosphor-coated screen that retains the image even after the electron beam has moved away. DVSTs were primarily used in applications where static images needed to be displayed for extended periods, such as radar displays, flight simulators, and medical imaging systems. However, they were eventually superseded by more advanced display technologies such as LCD and LED screens due to their higher resolution, lower power consumption, and smaller form factors.

**Explain how CRT parts help in achieving the common goal of displaying graphics on the screen. [4Marks]**

The various parts of a Cathode Ray Tube (CRT) work together to achieve the common goal of displaying graphics on the screen:

1. **Electron Gun:** The electron gun is responsible for emitting a focused beam of electrons. It's the starting point of the process. By controlling the intensity of the electron beam, the brightness of the displayed image can be controlled.

2. **Deflection Coils:** The deflection coils are responsible for shaping and directing the electron beam as it moves across the screen. These coils control both the horizontal and vertical movement of the beam, allowing it to scan the entire screen surface in a precise manner.

**3. Phosphor-Coated Screen:** The front surface of the CRT screen is coated with a layer of phosphorescent material. When the electron beam strikes this phosphor coating, it emits light, creating the visible image on the screen. Different phosphor materials produce different colors when excited by the electron beam, allowing for the display of a wide range of colors.

**4. Electrostatic Focusing and Accelerating Anodes:** These components help in focusing and accelerating the electron beam to ensure that it remains sharp and concentrated as it moves across the screen. The focusing anode ensures that the beam stays narrow, while the accelerating anode increases the speed of the electrons, allowing them to reach the phosphor coating with sufficient energy to produce a bright image.

**5. Control Circuits and Electronics:** The control circuits and electronics within the CRT monitor control the timing and intensity of the electron beam, as well as other parameters such as screen refresh rate and resolution. These circuits ensure that the image displayed on the screen is accurate and stable.

**Tablets are perfect in getting input of a two-dimensional picture, how can it be modified to get input from three-dimensional picture? [6Marks]**

**1. Depth Sensing Technology:** Integrate depth sensing technology into the tablet to capture the depth information of objects placed on its surface. Depth sensing cameras or sensors, such as time-of-flight (ToF) cameras or structured light sensors, can be embedded in the tablet to measure the distance to objects. This allows the tablet to capture not only the 2D position of the stylus or finger but also the depth or z-coordinate, enabling interactions with 3D content.

**2. Stereo Cameras:** Equip the tablet with stereo camera setups consisting of two or more cameras positioned to capture images from slightly different perspectives. By analyzing the disparity between corresponding points in the stereo images, the tablet can estimate the depth of objects in view. This approach enables the tablet to perceive the 3D structure of objects and surfaces, facilitating interactions with 3D scenes.

**3. Infrared Depth Sensing:** Incorporate infrared depth sensing technology, such as Microsoft's Kinect sensor or similar systems, into the tablet. These systems emit infrared light and measure the time it takes for the light to bounce back from objects, allowing them to create depth maps of the scene. By integrating such technology into the tablet, it can detect and interpret gestures or interactions in 3D space.

**4. Haptic Feedback:** Integrate haptic feedback mechanisms into the tablet to provide tactile sensations corresponding to interactions with 3D objects or surfaces. For example, the tablet could incorporate vibration motors or force feedback mechanisms to simulate the sensation of touching or manipulating virtual objects in 3D space. This enhances the user's sense of immersion and interaction with 3D content.

**5. Gesture Recognition:** Implement gesture recognition algorithms to interpret hand movements and gestures in 3D space. By analyzing the motion of the user's hands or fingers captured by the tablet's cameras or sensors, the tablet can infer gestures such as swiping, pinching, or rotating in three dimensions. This enables intuitive manipulation of 3D objects or scenes on the tablet's display.

**6. Augmented Reality (AR) Integration:** Combine the tablet with AR technologies to overlay virtual 3D content onto the real-world environment captured by the tablet's camera. This allows users to interact with virtual objects superimposed on their physical surroundings, providing a seamless integration of the digital and physical worlds.

**What is Scan conversion with respect to computer graphics?**

Scan conversion, in the context of computer graphics, refers to the process of converting geometric primitives (such as lines, curves, and polygons) defined in a continuous space into a discrete pixel grid for display on a raster display device, such as a computer monitor.

The process involves determining which pixels on the display grid should be illuminated to accurately represent the shape of the geometric primitive being drawn. This is done by determining the intersection points between the primitive and the grid of pixels, and then determining which pixels fall within the boundaries of the primitive.

There are different algorithms and techniques for scan conversion depending on the type of primitive being drawn. For example:

**1. Line Drawing:** Bresenham's line drawing algorithm is a commonly used technique for scan converting lines. It efficiently determines which pixels to illuminate along a line segment by considering the slope of the line and using integer arithmetic to minimize computational overhead.

**2. Polygon Filling:** Scan conversion of polygons involves determining which pixels lie within the boundaries of the polygon. Various algorithms, such as the scan-line polygon fill algorithm, divide the polygon into horizontal bands and determine the intersections between each scan line and the edges of the polygon to efficiently fill the interior.

**3. Circle and Ellipse Drawing:** Algorithms for drawing circles and ellipses also involve scan converting the curves by determining which pixels fall within the boundaries of the curves. The midpoint circle algorithm is a popular technique for drawing circles, while the midpoint ellipse algorithm is commonly used for ellipses.

**It is important to be specific about polygons, in your own opinion, why do you think so?**

**[4Marks]**

Being specific about polygons in computer graphics is important for several reasons:

**1. Accuracy and Precision:** Polygons are fundamental geometric shapes used to represent complex objects and surfaces in computer graphics. Being specific about polygons ensures that the boundaries and details of objects are accurately represented on the screen or in the rendered image. This is crucial for creating realistic and visually appealing graphics.

**2. Efficiency:** When dealing with complex scenes containing numerous polygons, being specific about the properties and attributes of each polygon allows for efficient rendering and processing. By providing detailed information about polygons, such as vertex coordinates, edge connectivity, and surface properties, computational resources can be optimized for rendering, collision detection, and other operations.

**3. Interactivity and Manipulation:** Specific information about polygons enables interactive manipulation and editing of objects in computer graphics applications. With precise polygon data, users can perform operations such as scaling, rotation, translation, and deformation accurately and predictably. This enhances the user experience and allows for intuitive interaction with 3D models and scenes.

**4. Compatibility and Standards:** Many computer graphics software and file formats rely on specific representations of polygons to ensure compatibility and interoperability between different systems and applications. Adhering to standard polygon representations facilitates data exchange, collaboration, and integration across various graphics platforms and tools.

**Describe why yx algorithm is called so?**

The "YX algorithm" you mentioned is likely a reference to the Bresenham's line drawing algorithm, often abbreviated as the "Bresenham's algorithm" or simply "Bresenham algorithm." It's a commonly used algorithm for drawing lines on a raster display device, such as a computer screen.

The algorithm is named after its creator, Jack Elton Bresenham, who developed it in 1962 while working at IBM. However, it's worth noting that "YX algorithm" might also refer to some other specific algorithm in a different context. If you're referring to a different algorithm, please provide more context, and I can give you a more accurate explanation.

**QUESTION FOUR [20 MARKS]**

1. **What do you know about the following terms? [4Marks]**
2. **DDA**

**DDA (Digital Differential Analyzer):**

DDA is a line drawing algorithm used in computer graphics to draw lines on a raster display device. It works by calculating the incremental changes in x and y coordinates between two endpoints of a line and then plotting the pixels between these endpoints based on the calculated increments. DDA is a simple and straightforward algorithm but may suffer from precision issues due to rounding errors.

**Transformation:**

Transformation refers to the process of modifying the geometric attributes of objects in a graphics scene. Common transformations include translation (moving objects), rotation (turning objects), scaling (resizing objects), and shearing (distorting objects). Transformations are applied using mathematical operations to manipulate the coordinates of vertices defining the objects.

1. **Explain requirements that a good line drawing algorithm should meet. [4Marks]**

**1. Accuracy:** The algorithm should accurately represent the intended line on the display device without distortion or artifacts.

**2. Efficiency:** The algorithm should be computationally efficient, minimizing the number of calculations and iterations required to draw the line.

**3. Speed:** The algorithm should be fast, especially when drawing lines in real-time applications such as computer games or simulations.

**4. Adaptability:** The algorithm should be adaptable to different display resolutions and orientations, ensuring consistent performance across various devices.

**5. Flexibility:** The algorithm should be flexible enough to handle lines of various slopes and lengths, as well as support anti-aliasing for smoother line rendering.

**6. Ease of Implementation:** The algorithm should be easy to understand, implement, and maintain, with clear and well-documented procedures.

1. **Write a C program to generate a line using Bresenham's algorithm [6Marks]**

#include <stdio.h>

#include <graphics.h>

void drawLineBresenham (int x1, int y1, int x2, int y2) {

int dx = x2 - x1;

int dy = y2 - y1;

int x = x1;

int y = y1;

int steps;

if (dx > dy) {

steps = dx;

} else {

steps = dy;

}

float increment\_x = (float) dx / steps;

float increment\_y = (float) dy / steps;

for (int i = 0; i < steps; i++) {

putpixel(round(x), round(y), WHITE);

x += increment\_x;

y += increment\_y;

}

}

int main() {

int gd = DETECT, gm;

initgraph (&gd, &gm, "");

int x1 = 100, y1 = 100;

int x2 = 300, y2 = 300;

drawLineBresenham(x1, y1, x2, y2);

delay (5000);

closegraph ();

return 0;

}

**d) Describe various difficulties that arise in drawing circles using DDA method with it's**

**differential equation and how to overcome them. [6Marks]**

Drawing circles using the DDA method can pose several difficulties:

- **Rounding Errors:** The DDA method may suffer from rounding errors, leading to inaccuracies in the circle's shape and position.

- **Uneven Pixel Distribution:** Due to the incremental nature of the DDA method, pixels may not be evenly distributed along the circumference of the circle, resulting in a jagged or uneven appearance.

- **Inefficient:** The DDA method requires calculating and plotting many intermediate points along the circumference of the circle, making it computationally inefficient for large circles.