

## 2D Chair Drawing in Mode 13h — Report

### 1) Objective

Implement a simple 2D chair on the DOS graphics screen using only the MS-DOS/BIOS services taught in class. The program:

- Switches to **320×200, 256-color graphics (Mode 13h)**
- Draws a chair using only **line segments** (rendered as sequences of pixels)
- Waits for a keypress
- Returns safely to **text mode**

The chair includes a seat with perspective edges, **two front legs**, **one horizontal connector** between those legs, and a **backrest** (outer frame + two inner vertical bars). Everything is drawn in **one color** for clarity.

---

### 2) Development Environment

- **Assembler/Emulator:** EMU8086 (16-bit real mode)
  - **Model:** .model small, .stack 100h
  - **Interrupts used:**
    - INT 10h (BIOS Video): set video mode and plot pixels
    - INT 21h (MS-DOS): wait for key and exit to DOS
- 

### 3) Code Overview

#### 3.1 Program setup

```
mov ax,@data
```

```
mov ds,ax
```

Initializes the **data segment** to allow access to variables and messages.

---

#### 3.2 Enter graphics mode

```
mov ax,0013h
```

```
int 10h
```

This BIOS call activates **Mode 13h** — a 320×200-pixel, 256-color graphics mode. Each pixel can be directly addressed by coordinates (**CX**, **DX**), and the color is given in **AL**.

---

### 3.3 Choose drawing color

```
mov al,14
```

Sets the color to **bright yellow** for all pixels.

---

### 3.4 Drawing the seat

The seat is designed with front, back, and diagonal lines to simulate a simple 3D look.

1. **Front edge (x=100..180, y=130):** A horizontal line at the front of the chair.
2. **Back edge (x=120..200, y=115):** A slightly raised line to show depth.
3. **Left side (100,130 → 120,115):** A diagonal connection between front and back.
4. **Right side (180,130 → 200,115):** Another diagonal connection on the opposite side.

Each line is made using a pixel plotting loop with INT 10h function **AH=0Ch**. Every iteration draws one pixel, then moves **CX** and/or **DX** toward the next position.

---

### 3.5 Drawing the legs

The legs are straight vertical lines:

- **Front-left leg:** at x=110, from y=130 to y=180
- **Front-right leg:** at x=170, from y=130 to y=180
- **Back-right leg:** slightly slanted backward (x=200→195, y=115→180)

These loops increment **DX** (the Y position) each time to draw vertically downwards.

---

### 3.6 Horizontal connector

A **horizontal bar** is added between the two front legs at y=165 (from x=110 to x=170). This gives the chair structural stability and improves realism.

---

### 3.7 Backrest

The backrest is built as a rectangle with two inner supports:

- **Left vertical bar:** (115,70) to (115,115)
- **Right vertical bar:** (195,70) to (195,115)
- **Top bar:** horizontal line (115,70) to (195,70)
- **Two inner bars:** vertical lines at x=140 and x=160, from y=70 to y=115

These are again vertical or horizontal pixel loops, keeping the design simple but realistic.

---

### 3.8 Wait and exit

```
mov ah,07h
```

```
int 21h
```

Waits for any key press without showing the character.

Then the program safely returns to text mode and exits:

```
mov ax,0003h ; switch to 80×25 text mode
```

```
int 10h
```

```
mov ax,4C00h ; exit to DOS
```

```
int 21h
```

---

## 4) Algorithm Details

### 4.1 Pixel plotting primitive

- **Function:** INT 10h / AH=0Ch
- **Parameters:**
  - AL = color

- BH = page number (always 0 here)
- CX = X coordinate
- DX = Y coordinate

Each pixel is plotted one by one to form straight lines.

---

## 4.2 Line drawing technique

This program uses **simple incremental line drawing**, similar to the basic **DDA (Digital Differential Analyzer)** idea, but limited to:

- Horizontal (change CX)
- Vertical (change DX)
- 45° diagonal (change both)

Since the lines are only 0°, 45°, or 90°, we don't need Bresenham's algorithm — loops and increments are sufficient.

Example for horizontal:

HLine:

```
int 10h
inc cx
cmp cx,x2
jbe HLine
```

Example for vertical:

VLine:

```
int 10h
inc dx
cmp dx,y2
jbe VLine
```

---

## 4.3 Control flow

Each section runs sequentially:

1. Draw seat
2. Draw legs
3. Draw connector
4. Draw backrest
5. Wait for key
6. Restore text mode and exit

There are **no subroutines**, keeping logic simple and easy to trace.

---

## 5) Coordinate Plan & Design

Part	X Range	Y Range	Description
Seat front edge	100–180	130	Main sitting surface
Seat back edge	120–200	115	Slightly higher back for perspective
Legs	~110–200	130–180	Vertical supports
Connector	110–170	165	Horizontal brace
Backrest	115–195	70–115	Upright frame
Inner bars	140,160	70–115	Decorative vertical bars

The coordinates are chosen so that all lines meet properly, forming a proportionate 2D chair.

---

## 10) Conclusion

This assembly program successfully demonstrates **basic computer graphics drawing** using only low-level DOS BIOS interrupts.

By combining **horizontal, vertical, and diagonal pixel loops**, it constructs a clear, symmetrical 2D chair entirely with line segments.

It uses simple but effective control logic, adheres strictly to the course's interrupt rules (INT

10h and INT 21h), and shows how structured pixel plotting can form meaningful shapes on a graphics screen.

