

## Lucrarea de laborator nr. 1

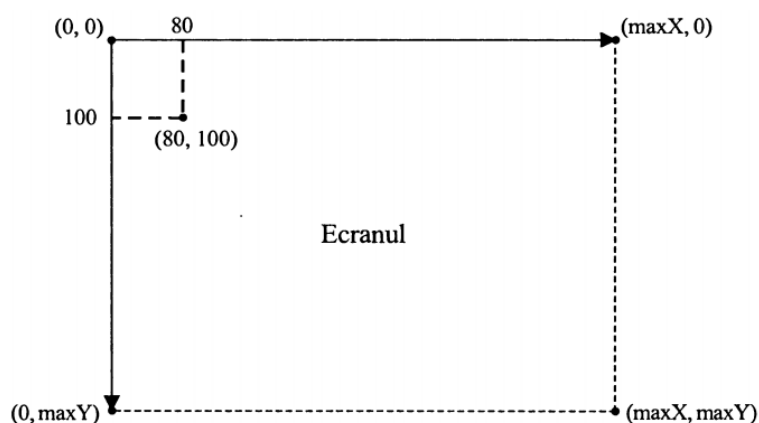
### Tema: Lucrul cu ecranul grafic. Primitivele grafice

#### Repere teoretice:

În regimul grafic, ecranul poate fi considerat ca fiind format dintr-o matrice de puncte numiți pixeli. Un punct de pe ecran poate fi aprins (colorat) într-o anumită culoare, iar poziția sa este definită prin coordonatele sale  $x$  și  $y$  (numere întregi) reprezentând coloana, respectiv linia, unde :

$$0 \leq x < \max X :$$

$$0 \leq y < \max Y.$$



Limitele  $\max X$  și  $\max Y$  depind de placa video, iar valorile acestora sunt returnate de către funcțiile *GetMaxX* și *GetMaxY*. Exemple de programe ce folosesc aceste funcții le găsiți în Anexa 1. De exemplu pentru plăcile video, de tipul *EGAHi* sau *VGAMed*,  $\max X=640$  și  $\max Y=480$ .

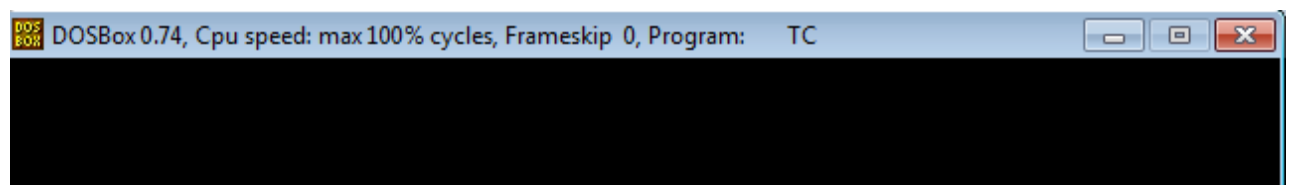
Pentru a utiliza ecranul în mod grafic, va trebui mai întâi să selectăm modul grafic, adică să trecem ecranul din modul text în modul grafic prin *inițializarea* sesiunii grafice (vezi mai sus). La terminarea sesiunii grafice sistemul va reveni în modul text.

#### 4 Primitivele grafice

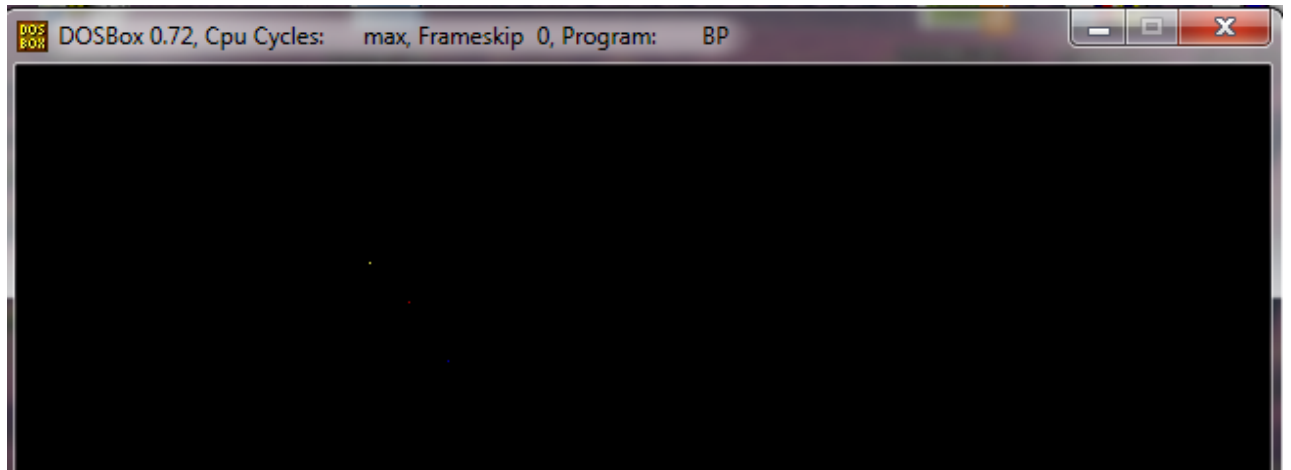
**Rezolvare:** Generarea unui punct se realizează cu ajutorul funcției *Putpixel*, descrierea căreia o găsiți în anexa 1.

Codul sursă al programului scris în Turbo Pascal	Codul sursă al programului scris în Turbo C++
<pre> Program puncte;  uses crt, graph;  var dr, dm:integer;  begin  dr:=detect;  initgraph(dr,dm,' ');  putpixel(200,120,red);  putpixel(180,100,yellow);  putpixel(220,150,blue);  readln;  closegraph;  end. </pre>	<pre> #include&lt;stdio.h&gt;  #include&lt;conio.h&gt;  #include&lt;graphics.h&gt;  void main()  {  int gd=DETECT, gm;  initgraph(&amp;gd,&amp;gm," ");  putpixel(220,120, RED);  putpixel(150, 180, YELLOW);  putpixel(200, 150, BLUE);  getch();  closegraph;  } </pre>

**Rezultatul execuției programului Turbo C++:**



### Rezultatul execuției programului Borland Pascal:



### Problema 2: Generarea unor segmente de dreaptă

**Rezolvare:** generarea unui segment de dreaptă se realizează cu ajutorul funcției *Line*, descrierea căreia o găsiți în anexa 1.

Codul sursă al programului scris în Turbo Pascal	Codul sursă al programului scris în Turbo C++
<pre>Program linii;  uses crt, graph;  var dr, dm:integer;  begin  dr:=detect;  initgraph(dr,dm,' ');  setcolor(6);  line(200,100,350,200);  setcolor(14);</pre>	<pre>#include&lt;stdio.h&gt;  #include&lt;conio.h&gt;  #include&lt;graphics.h&gt;  #include&lt;dos.h&gt;  void main() {  int gd=DETECT, gm;  initgraph(&amp;gd,&amp;gm," ");  setbkcolor(15);</pre>

```
closegraph;
```

```
end.
```

```
setcolor(10);
```

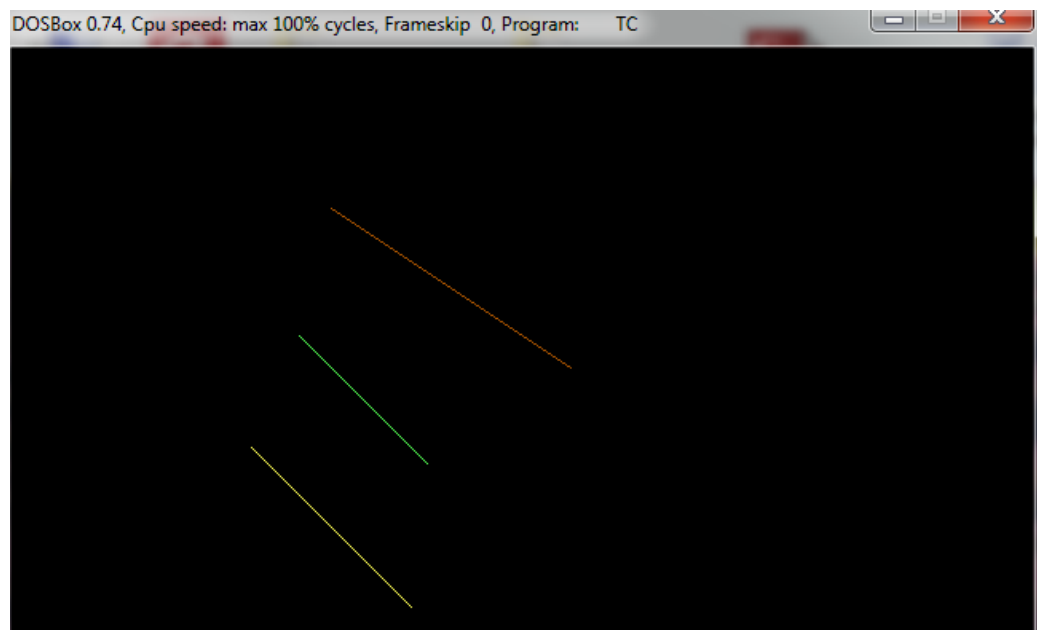
```
line(180,180,260,260);
```

```
getch();
```

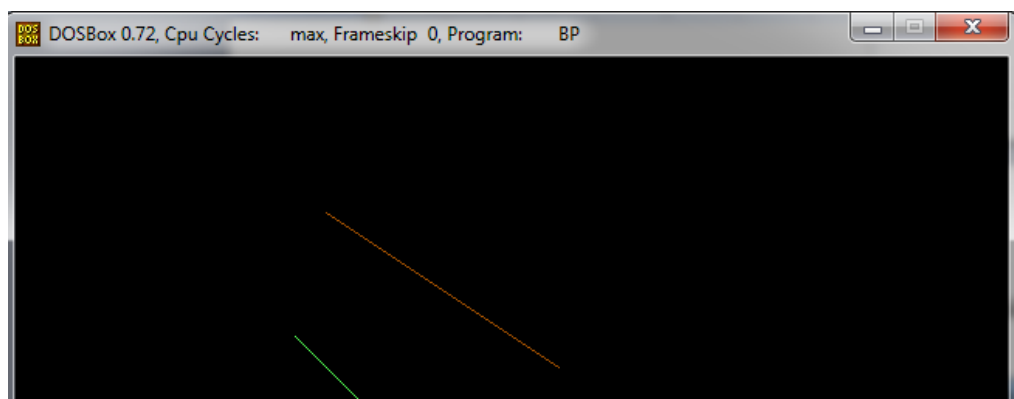
```
closegraph();
```

```
}
```

### Rezultatul execuției programului Turbo C++:



### Rezultatul execuției programului Borland Pascal:



**Lista sarcinilor de lucru individual:**

1. Scrieți programul care va genera 3 puncte ce se află pe o dreaptă paralelă la axa Oy.
2. Scrieți programul care va genera 3 puncte ce se află pe o dreaptă paralelă la axa Ox.
3. Scrieți programul care va genera 3 drepte paralele.
4. Scrieți programul care va genera 5 drepte ce se intersectează într-un punct.
5. Scrieți programul care va genera 3 circumferințe ce se intersectează cu razele diferite.
6. Scrieți programul care va genera două dreptunghiuri perpendiculare.