النقاط الضوئية والصور النقطية والخطوط والصور

الصور النقطية والخطوط:

الصور النقطية عبارة عن مصفوفة مستطيلة من الأصفار والواحدات تعمل كقناع رسم للجزء المستطيل الموافق لها من الإطار. لنفرض أنك رسمت صورة نقطية وكان اللون الموافق لها أحمر، عندها كل قيمة 1 في الصورة النقطية ستستبدل النقطة الضوئية الموافقة لها بنقطة ضوئية حمراء، وأما إذا كانت القيمة 0 في الصورة النقطية فلن تتأثر محتويات النقطة الضوئية.

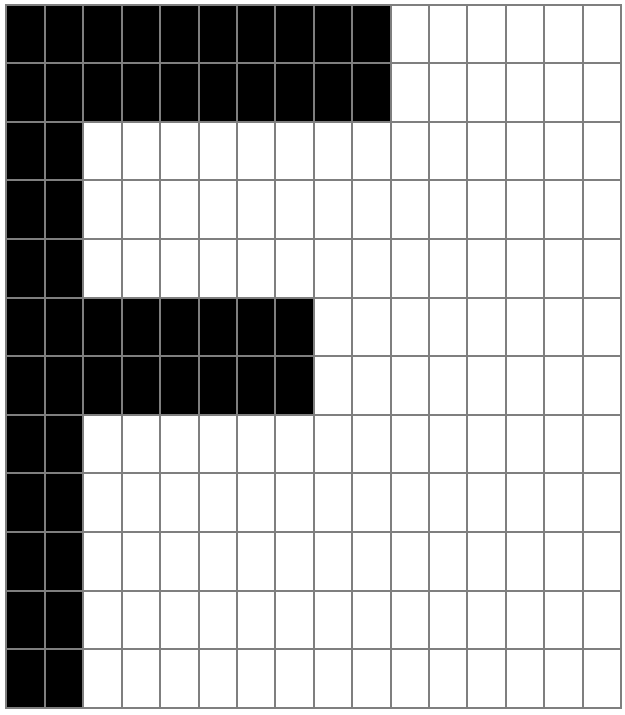
أكثر الاستخدامات الشائعة للصورة النقطية تتمثل في رسم الرموز على الشاشة.

تؤمن OpenGl أدنى مستوى من الدعم لرسم سلسلة من الرموز ومعالجة الخطوط.

يُستخدم الأمران glRasterPos*() و glBitmap() لتوضيع ورسم صورة نقطية وحيدة على الشاشة.

مثال:

سنعمل ضمن هذا المثال على رسم الرمز F ثلاث مرات على الشاشة. يُظهر الشكل التالي الرمز F كصورة نقطية مؤلفة من 12 سطر و 16 عمود، بالإضافة إلى بيانات الصورة النقطية الموافقة لهذا الرمز.



موقع المسح الحالي:

يمثل موقع المسح Raster الحالي المنطقة التي سترسم فيها الصورة النقطة التالية. في مثال الحرف F، يُعيَّن موقع المسح باستخدام الأمر glRasterPos*() إلى الموقع (20,20) الذي ترسم فيه الزاوية اليسارية السفلية من الحرف F.

الأمر المستخدم في المثال:

```
glRasterPos2i(20,20);
```

الشكل العام لأمر تعيين موقع المسح:

```
Void glRasterPos2i{234}{sifd}{v}(TYPE x, TYPE y, TYPE z, TYPE w);
```

تحددالمعاملات x,y,z,w إحداثيات موقع المسح، والقيم الافتراضية لـ z و w هي z=0 و w=1.

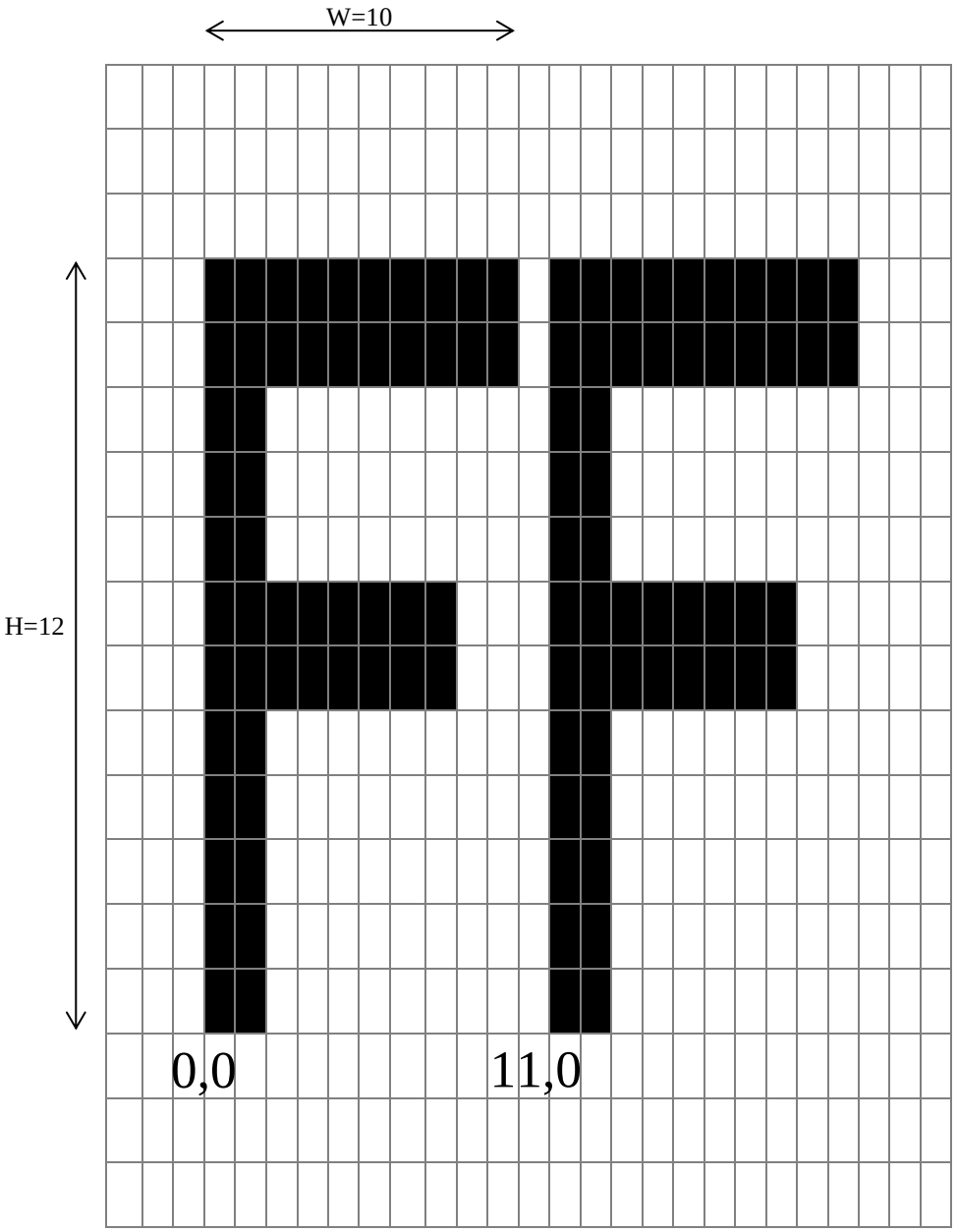
رسم الصورة النقطية:

بعد ان تحدد موقع المسح المطلوب، يمكنك استخدام الأمر glBimap() لرسم البيانات.

الشكل العام لأمر رسم الصورة النقطية:

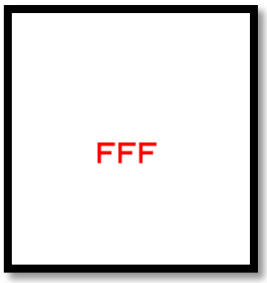
```
Void glBitmap(GLsizei width, GLsizei height, GLfloat xbo, GLfloat ybo, GLfloat xbi,
GLfloat ybi, const GLubyte *bitmap);
```

يرسم الصورة المحددة بالمؤشر bitmap الذي يشير إلى الصورة النقطية، وتوضع الصورة النقطية بدءاً من موقع المسح المعرف حديثاً. تشير معملات width و height إلى عرض وارتفاع الصورة النقطية بالنقطة الضوئية. ليس من الضروري أن يكون العرض من مضاعفات 8، على الرغم من أن البيانات تخزن على شكل رموز غير مؤشرة بحجم 8 خانة. استخدم xbo و ybo لتحديد موقع الصورة النقطية (القيم الموجبة تزيح الموقع إلى الأعلى واليمين، أما القيم السالبة فتزيح موقع الصورة إلى الأسفل واليسار). تشير xbi و ybi إلى الزيادة في x و y التي يجب إضافتها إلى موقع المسح بعد مسح الصورة، كما هو مبين في الشكل التالي:



بعد الانتهاء من الرسم، يزداد موقع المسح قيمة xbi و ybi بالاتجاهين x وy بشكل متتالي. من أجل الخطوط الأجنبية، تكون قيمة ybi مساوية لـ 0,0 وقيمة xbi موجبة. أما في حالة الخطوط العربية فتكون قيم xbi سالبة.

تطبيق1:




```
#include <GL/glut.h>
#include <stdlib.h>
#include <math.h>

GLubyte rasters[24]={0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,0xc0, 0x00,
                     0xff, 0x00, 0xff, 0x00, 0xc0, 0x00, 0xc0, 0x00, 0xc0, 0x00,
                     0xff, 0xc0, 0xff, 0xc0};

static void redraw(void);
void myinit(void)
{
   glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
   glClearColor(1.0, 1.0, 1.0, 0.0);
}

int main(int argc, char **argv)
{
        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
        glutInitWindowPosition(100,100);
        glutInitWindowSize(400,400);
       glutCreateWindow("Font Bitmap");
        myinit();
        glutDisplayFunc(redraw);
        glMatrixMode(GL_PROJECTION);
        gluPerspective(45,1.0,10.0,200.0);
        glMatrixMode(GL_MODELVIEW);
        glutMainLoop();
        return 0 ;
}
```

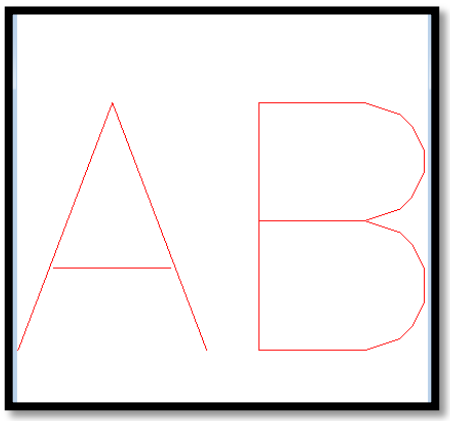
```
static void redraw(void)
}
      glLoadIdentity();
      glColor3f(1.0,0.0,0.0);
      glTranslatef(0.0,0.0,-40);
   glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
      glRasterPos2i (0, 0);
   glBitmap(10, 12, 0.0, 0.0, 12.0, 0.0, rasters);
   glBitmap(10, 12, 0.0, 0.0, 12.0, 0.0, rasters);
   glBitmap(10, 12, 0.0, 0.0, 12.0, 0.0, rasters);
      glutSwapBuffers();

}
```

تعديلات:

- اجعل كل حرف بلون مختلف.
- غير الحرف إلى الحرف E.
- غير ما يلزم للحصول على الناتج الموضح جانباً.
- عدل البرنامج لكتابة حرف الهاء كما هو موضح جانباً.
- عدل لكتابة حرف اللام مرتين الأول بلون أحمر والثاني بلون أخضر مع مراعاة اتجاه الكتابة باللغة العربية من اليمين إلى اليسار كما هو موضح.

تطبيق2:



```
#include <GL/glut.h>
#include <stdlib.h>
#include<math.h >
static void redraw(void);
int main(int argc, char **argv);

int main(int argc, char **argv)
{
      glutInit(&argc,argv);
      glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
```

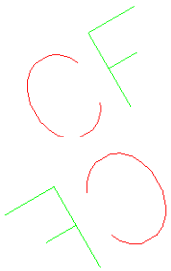
```
        glutInitWindowPosition(100,100);
        glutInitWindowSize(400,400);
        glutCreateWindow("GLUT Font");
        glutDisplayFunc(redraw);

        glMatrixMode(GL_PROJECTION);
        gluPerspective(45,1.0,10.0,300.0);
        glMatrixMode(GL_MODELVIEW);

        glutMainLoop();

        return 0 ;
}


static void redraw(void)
{
        glClearColor(1.0,1.0,1.0,0.0);
        glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();
        glTranslatef(-85.0f,-60.0f,-200.0f);
        glColor3f(1.0, 0.0, 0.0);
        glutStrokeCharacter(GLUT_STROKE_ROMAN,65);
        glutStrokeCharacter(GLUT_STROKE_MONO_ROMAN,66);
                                glutSwapBuffers();
}
```



تعديلات:

- كتابة الحرفين : CF.
- تغيير لون كل حرف.
- تطبيق تحويل الدوران للحصول على الشكل الموضح جانباً.

تطبيق تحويل تغيير الحجم للحصول على الشكل الموضح جانباً.