بيانيات الحاسوب
Computer Graphics

جامعة المنارة
MANARA UNIVERSITY

Dr.-Eng. Samer Sulaiman

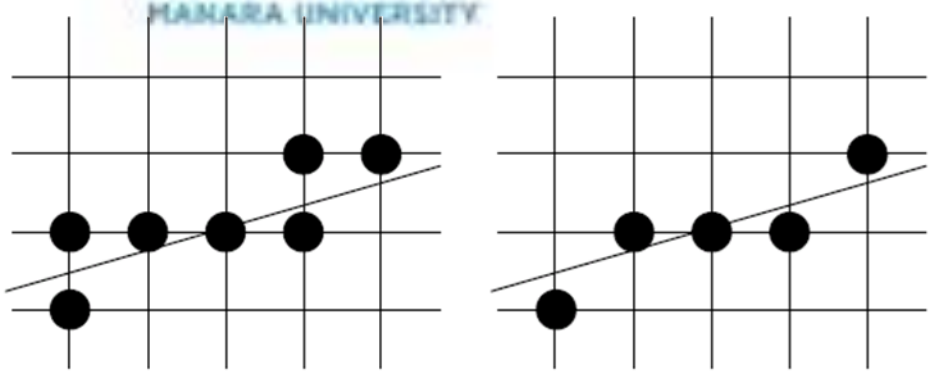
2022-2023

مفردات المنهاج

- **أساسيات بيانيات الحاسوب**
  - مقدمة: مفاهيم أساسية
  - التحويلات ثنائية البعد 2D
  - التحويلات ثلاثية البعد 3D
- **خوارزميات بيانيات الحاسوب**
  - الخوارزميات الهندسية
  - الخوارزميات النقطية الأساسية
  - خوارزميات إزالة الأسطح المخفية
- **نماذج بيانيات الحاسوب**
  - الألوان ونماذج الألوان
  - المنحنيات والأسطح: النمذجة الهندسية
  - نماذج الإضاءة والتظليل
  - عناصر التركيب والنمذجة الإجرائية

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - جميع الرسومات الحديثة التي يتم إنشاؤها بواسطة الكمبيوتر تعتمد على البيانات النقطية
  - يتم تمثيل كل صورة كمصفوفة ثنائية الأبعاد من النقاط (تسمى وحدات البكسل من عناصر الصورة)
  - لكل بكسل إحداثيات تكون عادةً زوجًا من القيم الصحيحة (X,Y) وبعض الألوان
  - تقع جميع وحدات البكسل على ما يسمى بالشبكة النقطية ، والتي تكون عادةً مجموعة من النقاط ذات الإحداثيات الصحيحة x و y
  - يجب أن تختار تنقيط الشكل الرياضي المثالي (مثل القطعة المستقيمة) لوحدات البكسل بحيث تكون
    - الاقتراب من المستقيم الفعلي
    - اعتبارها مجموعة متصلة على الشبكة النقطية



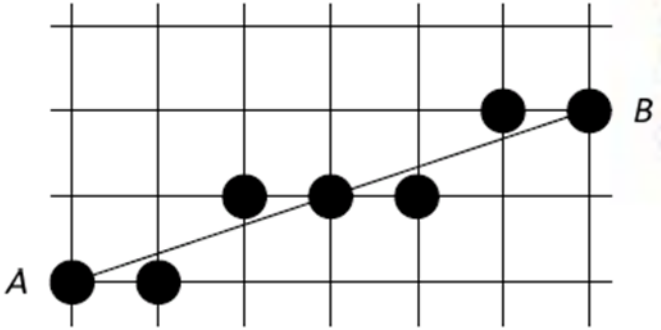
خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - أنواع الاتصال على شبكة البيانات النقطية:
    - 4-التوصيل 4-connectivity:
      - البكسلات (x0، y0) و (x1، y1) متصلة إذا وفقط إذا كانت $|x0 - x1| + |y0 - y1| \leq 1$
        - جميع الإحداثيات عدد صحيح
      - إذا كانت وحدتا بكسل متصلتان بأربع وحدات ، فيمكنهما الاختلاف في إحداثي واحد فقط x أو y
    - 8-التوصيل 8-connectivity:
      - البكسلات (x0، y0) و (x1، y1) متصلة إذا وفقط إذا كانت $|x0 - x1| \leq 1$ و $|y0 - y1| \leq 1$
    - إذا كان بكسلين متصلين بـ 4 ، فهما متصلان بـ 8
      - لكن العكس ليس صحيحًا
      - على سبيل المثال: وحدات البكسل (0 ، 0) و (1 ، 1) متصلة بـ 8 ، لكنها غير متصلة بـ 4
    - مجموعة البكسلات الموجودة على شبكة نقطية هي 4 أو 8 متصلة:
      - إذا كان لأي بكسلين من هذه المجموعة ، يوجد تسلسل من البكسل يبدأ من البكسل الأول وينتهي بالثاني بحيث يكون أي بكسلين لاحقين متصلين بـ 4 أو 8
    - يمكن تنقيط نفس الشكل بشكل مختلف اعتمادًا على نوع الاتصال المستخدم

جامعة المنارة

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - إعطاء التقريب للمستقيمات التي تقلل المسافة إلى المستقيم (خطأ تقريبي)
    - باستخدام مستقيم بـ 8-التوصيل
    - بافتراض أن A(xa, ya) و B(xb, yb) هما نقطتان لهما إحداثيات عدد صحيح
    - إنشاء صورة نقطية للقطعة المستقيمة AB:
      - بدءًا من الحالة حيث $0 \le yb - ya \le xb - xa$
    - بالنظر إلى أن القطعة المستقيمة جزء من المستقيم بالمعادلة $y = kx + b$
      - يقع المعامل k في النطاق من 0 إلى 1
    - لكل قيمة عدد صحيح xi بدءًا من xa وحتى xb
      - أن تكون بالضبط بكسل واحد (xi, yi) مع إحداثي x هذا
    - استخدام تابعين للتبسيط:
      - putPixel(x, y, c)
        - رسم بكسل في الموضع (x, y) باللون C
      - getPixel(x, y)
        - قراءة لون البكسل في الموضع المحدد (x, y)



جامعة المنارة
MANARA UNIVERSITY

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - الكود الأول:

```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  float k = (float)(yb-ya)/(float)(xb - xa);
  float b = ya - k * xa;
  for ( int x = xa; x <= xb; x++ )
    putPixel ( x, round ( k*x + b ), color );
}
```

- تحسين التابع من خلال ملاحظة أن قيمة السطر y سيتم تغييرها بواسطة k في كل تكرار حلقة

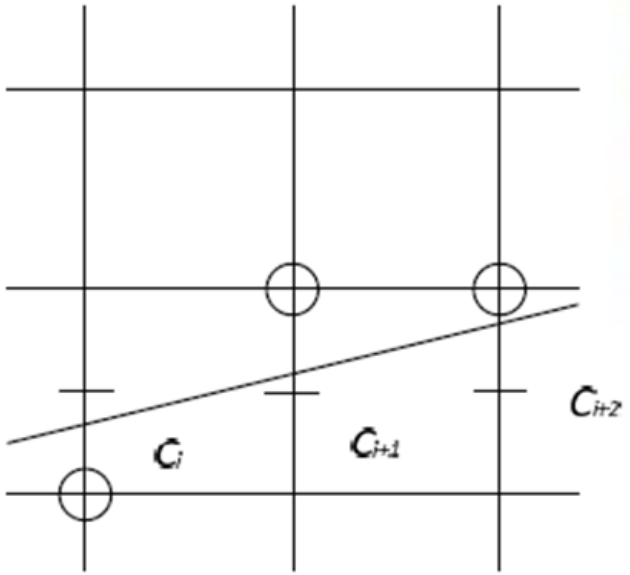
```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  float k = (float)(yb-ya)/(float)(xb - xa);
  float b = ya - k*xa;
  float y = ya;

  for ( int x = xa; x <= xb; x++, y += k )
    putPixel ( x, round ( y ), color );
}
```

- .

جامعة المنارة
MANARA UNIVERSITY

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - تتبع كيفية تغير الجزء الكسري من معادلة المستقيم عندما نصل إلى قيمة x التالية
      - $x_i = x_a + i$،
      - $c_i = \{kx_i + b\}$
      - إذا كان للبكسل من المرتبة $c_i \le 1/2$ i نقرب القيمة $kx_i + b$ للأسفل وزنجعل $y_i = kx_i + b$
      - وإلا فإننا نضع $y_i = kx_i + b + 1$
    - تغيير ci من خلال التكرار $x_0 = x_a, x_1, x_2, \cdots$
    - من خلال معرفة إحداثيات (x_i, y_i) للبكسل الحالي (جنبًا إلى جنب مع ci)
    - يمكن العثور على قيم البكسل التالي بالطريقة التالية:
      - $c_{i+1} = c_i + k$
      - $y_{i+1} = y_i$ (يتم تقريب y لأسفل) إذا $c_i \le 1/2$
    - خلاف ذلك:
      - $c_{i+1} = c_i + k - 1$
      - $y_{i+1} = y_{i+1}$ إذا $c_i > 1/2$
    - بالنسبة إلى $x_0 = x_a$ و $y_0 = y_a$، لدينا $c_0 = 0$



جامعة المنارة
MANARA UNIVERSITY

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)

```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  float k = (float)(yb-ya)/(float)(xb - xa);
  float y = ya;
  float c = 0;
  int   y = ya;

  putPixel ( xa, ya, color );
  for ( int x = xa+1; x <= xb; x++ )
  {
    if ( (c += k) > 0.5f )
    {
      c--;
      y++;
    }

    putPixel ( x, y, color );
  }
}
```

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - المقارنة بين c_i والصفر أسهل بكثير من المقارنة مع 1/2
      - باستخدام $di = 2c_i - 1$ بدلاً من c_i ومقارنتها مع الصفر
      - بما أن $c_0 = 0$ و $c_1 = k$ لدينا $d_0 = -1$ و $d_1 = 2k - 1$

```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  float k = (float)(yb-ya)/(float)(xb - xa);
  float y = ya;
  float d = 2*k - 1;
  int   y = ya;

  putPixel ( xa, ya, color );
  for ( int x = xa+1; x <= xb; x++ )
  {
    if ( d > 0 )
    {
      d += 2*k + 2;
      y++;
    }
    else
      d += 2*k;

    putPixel ( x, y, color );
  }
}
```

- .

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - تستخدم جميع التوابع بشكل كبير عمليات النقطة العائمة والتي يمكن أن تكون مكلفة
    - الخطوة الأخيرة هي التخلص تمامًا من جميع أرقام الفاصلة العائمة
      - جميع معلمات الإدخال هي عدد صحيح وبالنسبة لوحدات المعالجة المركزية البسيطة التي تستخدم عددًا صحيحًا فقط ضمن العمليات الحسابية فهي ذات كفاءة عالية
      - بما أن $k = (y_b - y_a)/(x_b - x_a)$
      - يتم حساب جميع قيم الفاصلة العائمة الأخرى عن طريق إضافة مضاعفات k وقيم عدد صحيح
      - كل di يمكن تمثيلها على شكل كسور $n / (x_b - x_a)$
        - إذا تم ضرب di بـ $x_b - x_a$ سنحصل على عدد صحيح فقط ضمن العمليات الحسابية
        - يجب مضاعفة الـ Deltas المستخدمة في هذه الخوارزمية أيضًا

جامعة المنارة

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)

```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  int dx = xb - xa;
  int dy = yb - ya;
  int d  = (dy<<1) - dx;
  int d1 = dy << 1;
  int d2 = (dy - dx) << 1;
  int y  = ya;

  putPixel ( xa, ya, color );
  for ( int x = xa+1; x <= xb; x++ )
  {
    if ( d > 0 )
    {
      d += d2;
      y++;
    }
    else
      d += d1;

    putPixel ( x, y, color );
  }
}
```

- .

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - تنقيط مستقيم باستخدام 4-التوصيل

```
void drawLine4 ( int xa, int ya, int xb, int yb,
                 int color )
{
  int dx = xb - xa;
  int dy = yb - ya;
  int d  = (dy<<1) - dx;
  int d1 = dy << 1;
  int d2 = (dy - dx) << 1;
  int x  = xa;
  int y  = ya;

  putPixel ( xa, ya, color );
  for ( int i = 1; i <= dy + dx; i++ )
  {
    if ( d > 0 )
    {
      d += d2;
      y++;
    }
    else
    {
      d += d1;
      x++;
    }

    putPixel ( x, y, color );
  }
}
```

- .

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - الحالة العامة
      - الحالة $0 \le -(yb - ya) \le xb - xa$ متماثلة مع الحالة التي تم النظر فيها بالفعل
        - الحاجة إلى تغيير y بمقدار -1 بدلاً من 1
      - يتم إرجاع الحالة $0 \le yb - ya > xb - xa$ إلى الحالة السابقة عن طريق تبديل متغيري x و y
      - قد يكون من الأفضل اكتشاف المستقيمات الأفقية والعمودية ومعالجتها بشكل منفصل
      - في النهاية ، سيحتوي الكود على عدد صحيح فقط من العمليات الحسابية ولا يحتوي على عمليات ضرب وتقسيم
        - يمكن تنفيذ الكود بسهولة حتى على المعالجات البسيطة جدًا

جامعة المنارة
MANARA UNIVERSITY

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - الحالة العامة

```
void drawLine ( int xa, int ya, int xb, int yb,
                int color )
{
  int dx = abs(xb - xa);
  int dy = abs(yb - ya);
  int sx = xb >= xa ? 1 : -1;     // sign of xb-xa
  int sy = yb >= xy ? 1 : -1;     // sign of yb-ya
  if ( dy <= dx )
  {
    int d  = (dy<<1) - dx;
    int d1 = dy << 1;
    int d2 = (dy - dx) << 1;
    int x  = xa + sx;
    int y  = ya;

    putPixel ( xa, ya, color );
    for ( int i = 1; i <= dx; i++, x += sx )
    {
      if ( d > 0 )
      {
        d += d2;
        y += sy;
      }
      else
        d += d1;

      putPixel ( x, y, color );
    }
  }
  else
  {
    int d  = (dx<<1) - dy;
    int d1 = dx << 1;
    int d2 = (dx - dy) << 1;
    int x  = xa;
    int y  = ya + sy;
```

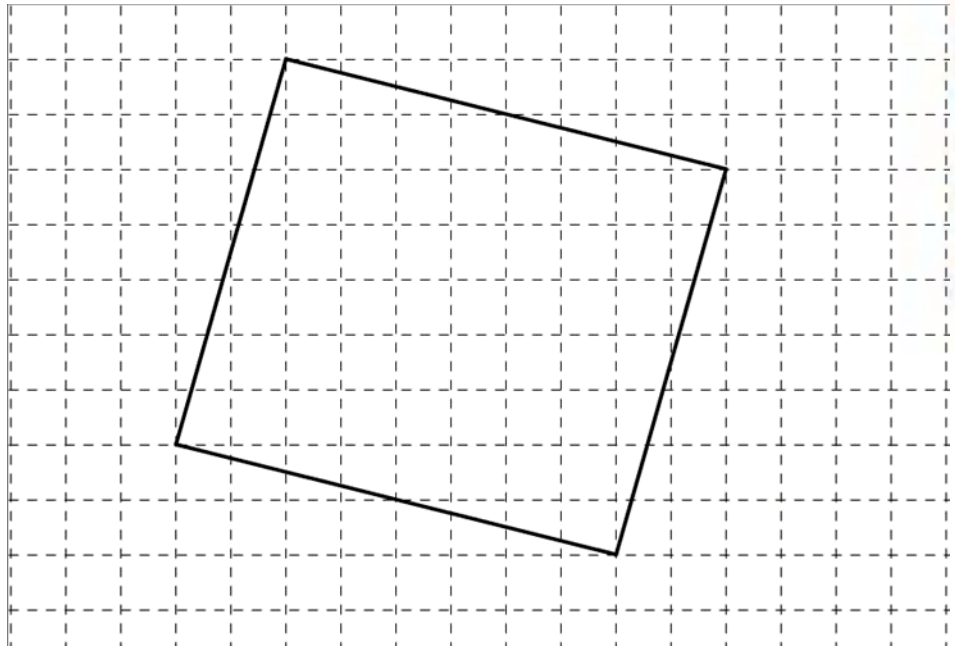
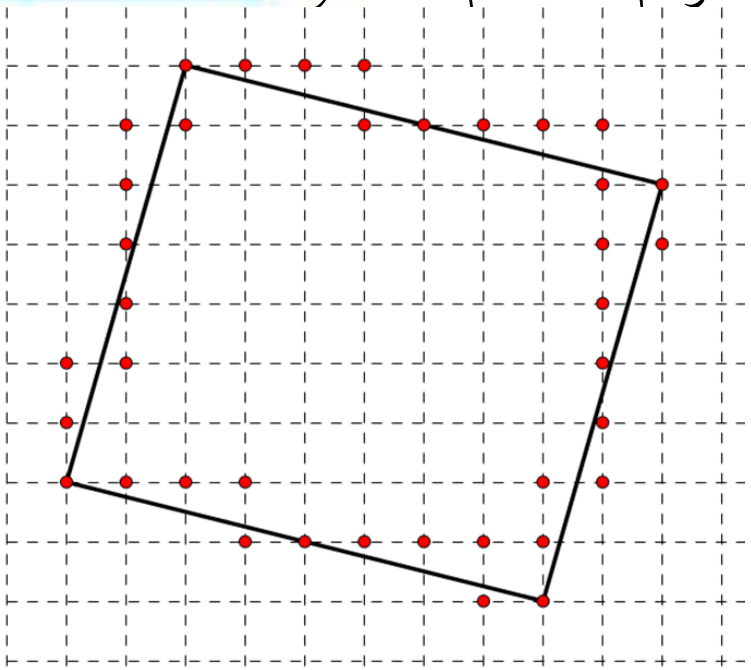
جامعة المنارة
MANARA UNIVERSITY

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
    - الحالة العامة

```
    putPixel ( xa, ya, color );
    for ( int i = 1; i <= dy; i++, y += sy )
    {
      if ( d > 0 )
      {
        d += d2;
        x += sx;
      }
      else
        d += d1;

      putPixel ( x, y, color );
    }
  }
}
```

- .

خوارزميات بيانيات الحاسوب

- خوارزميات البيانات النقطية الأساسية Basic raster algorithms :
  - خوارزمية Bresenheim (Bresenheim's line algorithm)
  - مثال:
    - ليكن لدينا الشكل التالي والذي يوضح توضع مربع على شاشة العرض والتي يمكن تمثيلها كشبكة نقاط تقاطعها تمثل بيكسلات الشاشة والمطلوب: حدد النقاط (البكسلات) التي ستضيء ضمن الشاشة ليتم رسم المربع الموضح بالشكل دون ملئه فيما لو تم استخدام النمط 4-Connectivity!
    - الحل: