# مفردات المقرر

**الفصل الأول :** مقدمة وتعاريف حول البيانات والرسم.

**الفصل الثاني :**تحويل المسح:

- مسح مستقيم.
- مسح دائرة.
- مسح القطع الناقص.
- الأقواس والقطاعات.
- مسح المستطيل.

**الفصل الثالث :**الصفات المميزة للرسومات الأولية:

- عرض الخطوط.
- حركة القلم.
- تعبئة المضلع.
- خوارزميات خط المسح.

**الفصل الرابع :** التحويلات ذات الأبعاد الثنائية:

- الإزاحة.
- الدوران.
- التقييس( تغيير الحجم).
- الانعكاس المرآتي حول المحاور.
- تركيب التحويلات الهندسية.
- التحويلات الهندسية حول جملة إحداثيات.

**الفصل الخامس** :التحويلات الهندسية ثلاثية الأبعاد:

- إزاحة.
- تدوير.

- تقييس.

**الفصل السادس :** العرض والقص ثنائي الأبعاد.

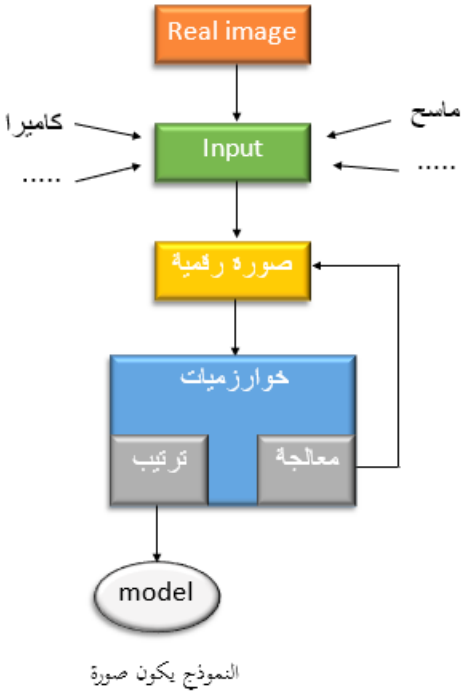
# مقدمة حول المنهاج

يمكننا تقسيم معالجة البيانات إلى مجموعتين:

1 . **Sampled-based graphics :**

وفيه نفكر( تخزين، معالجة، )..على مستوى الصورة، أي إن الدخل هو صورة والخرج هو صورة أيضا

.



النموذج يكون صورة

**الصورة :** هي مصفوفة من البكسيلات تمثل الكثافة اللونية لكل نقطة منها.

**البكسل: pixel (picture elements)**

هو أصغر عنصر في الصورة في بيئة 2D ، نحصل عليه بعملية رقمنه للصورة وهو موجود في الذاكرة، فمثلا:

-إذا كان البكسل مخزن على 1 bit . لكل بكسل إحدى قيمتين فقط هما:

0 و1 فالصورة تكون ذات لونين فقط( أبيض وأسود.)

-إذا كان البكسل مخزن على 8 bit . يمكن للبكسل أن يأخذ أي قيمة في المجال: 0 و 225 فتكون الصورة بالرمادي.

-إذا كان البكسل مخزن على 24 bit . يمكن للبكسل أن يأخذ أي قيمة في المجال $2^{24}$-1,0 :فتكون الصورة بالألوان الحقيقية.

## 2. Geometry-based graphics .

-هنا ننطلق من مشهد مراد تصميمه وصولا إلى الخرج الذي يكون صورة، مع أنه 3D model ولكننا نراه في النهاية كصورة على الشاشة، وهذا هو مجال عملنا

-أي إننا نقوم بإنشاء وتخزين ومعالجة model وليس صورة

-فهو يعتمد على أشكال أساسية( خطوط، مضلعات، )..بالإضافة إلى الألوان والإكساء

-هنا نواجه بعض المشاكل لأن graphics في هذه الحالة مكلفة من حيث زمن المعالجة وزمن الإظهار وكذلك من جهة الذاكرة..، مما يجعل من الأهمية بمكان إيجاد نمذجة صحيحة للأشياء.

## تطبيقات رسومات الحاسوب:

يحتوي Computer Graphics على العديد من التطبيقات ، بعضها مدرج أدناه:

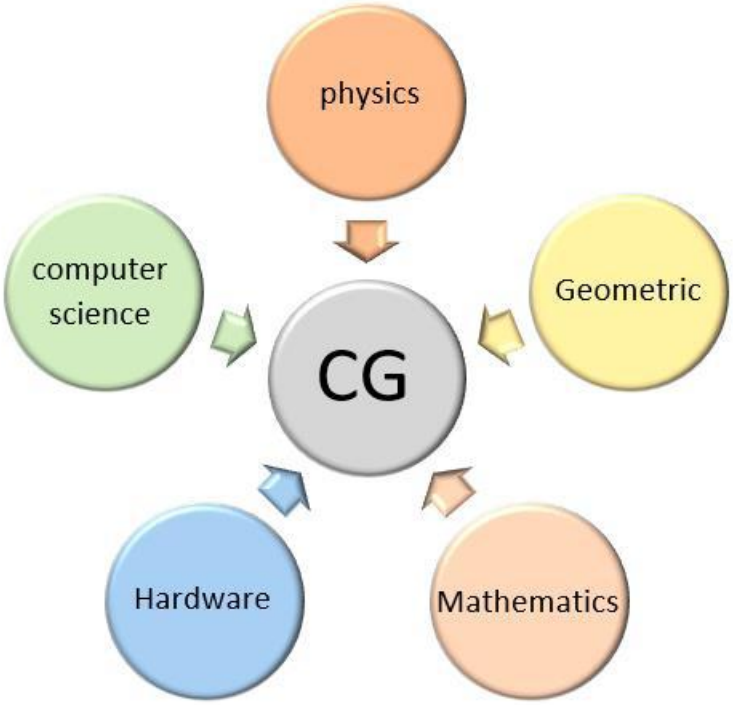
- واجهات مستخدم رسومات الكمبيوتر - ( GUI ) نموذج رسومي موجه بالماوس يسمح للمستخدم بالتفاعل مع جهاز كمبيوتر.
- رسومات عرض الأعمال " - الصورة تساوي ألف كلمة."
- رسم الخرائط - رسم الخرائط

- خرائط الطقس - رسم الخرائط في الوقت الحقيقي ، التمثيل الرمزي.
- تصوير الأقمار الصناعية - الصور الجيوديسية.
- تحسين الصور - شحذ الصور غير واضحة.
- التصوير الطبي - التصوير بالرنين المغناطيسي ، الأشعة المقطعية ، وما إلى ذلك - الفحص الداخلي غير الغازية.

•الرسومات الهندسية - الميكانيكية والكهربائية والمدنية وغيرها - استبدال مخططات الماضي.
•الطباعة - استخدام الصور الشخصية في النشر - استبدال النوع الصعب من الماضي.
•الهندسة المعمارية - خطط البناء ، الرسومات الخارجية - استبدال المخططات والرسومات اليدوية من الماضي.
•الفن - أجهزة الكمبيوتر توفر وسيلة جديدة للفنانين.
•التدريب - أجهزة محاكاة الطيران ، والتعليم بمساعدة الكمبيوتر ، إلخ.
•الترفيه - الأفلام والألعاب.
•المحاكاة والنمذجة - استبدال النمذجة والتشريعات المادية.

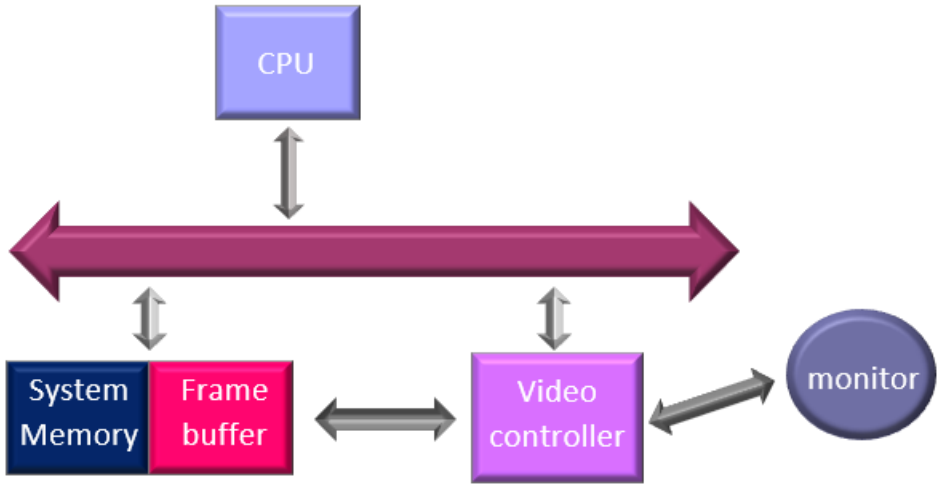
## -أهم البرامج التي تقوم بذلك:

Adobe Illustrator, Claris's MacDraw Autodesk's AutoCAD and Studio 3D, Alias Wavefront Studio and Maya, SoftImage 3D




**ملاحظة:**

لن نقوم في هذا المقرر بتعلم Maya أو3D Studio MAX أو Softimage وإنما مهنتنا في هذا المقرر هي فهم الخلفية التي تعمل بها مثل هذه البرامج.




## تحويل المسح

يتألف العديد من الصور بدءا من الرسومات ثنائية الأبعاد إلى المشاهد للكيانات ثلاثية الأبعاد من عناصر أولية مثل النقاط والدوائر والخطوط ومضلعات مملوءة..
وغالبا ما تعرف هذه المكونات في حيز مستمر( رسم خط أو مثلث )..مقارنة بالبكسلات الفردية في حيز( فضاء )صورة غير مترابط.
مثلا يعرف المستقيم بنقطتين ومعادلة، في حين تعرف الدائرة بنصف قطرها وموقع مركزها ومعادلتها.
إن نظام الرسوميات أو برنامج التطبيق مسؤول عن تحويل كل عنصر أولي من تعريفه الهندسي إلى مجموعة من البكسلات التي تؤلف العنصر الأولي يجب أن نتكلم قليلا عن إحداثيات الشاشة أو الصورة مقارنة بالإحداثيات الحقيقية.

❖ **تحويل مسح النقطة :** هنالك طريقتين لتحويل مسح النقطة (x,y):

**الطريقة الأولى** : نحول النقطة الحقيقية (x,y) إلى البكسل عند النقطة وعند الموضع (x',y') عن طريق أخذ الجزء الصحيح فقط من العددين y و x

أي : $x` = floor(x)$ و $y` = floor(y)$

حيث أن التابع floor يعطي أكبر عدد صحيح أقل أو يساوي المتغير المطلق.

أي نجعل نقطة الأصل لنظام إحداثيات مستمر لـ (x,y) عند الزاوية اليسرى السفلى لشبكة البكسل في فضاء الصورة.

**ملاحظة:**

كل النقاط التي تحقق $x` \le x < x` + 1$ و $y` \le y < y` + 1$ توضع في البكسل $(x\ ,\ y)$ أي مثلاً :

النقطة $p_1(1.7,0.8)$ تتوضع في البكسل $(1,0)$

والنقطة $p_2(2.2,1.3)$ و $p_3(2.8,1.9)$ تمثلان في البكسل $(2,1)$

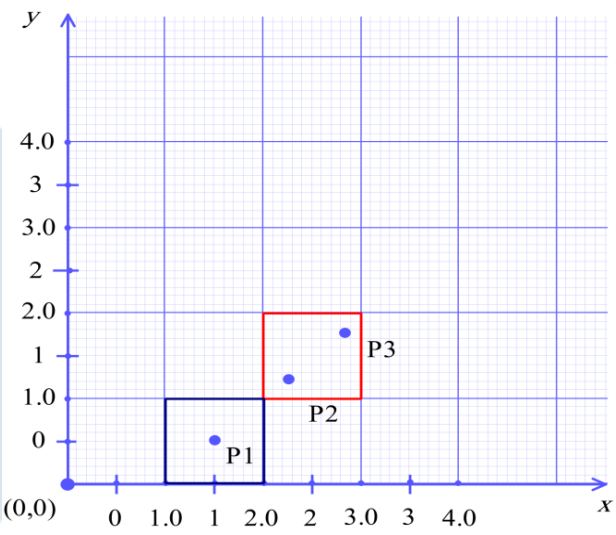
**الطريقة الثانية:** تحول النقطة (x,y) إلى البكسل بالشكل:

$$y` = floor(y + 0.5) \quad \text{و} \quad x` = floor(x + 0.5)$$

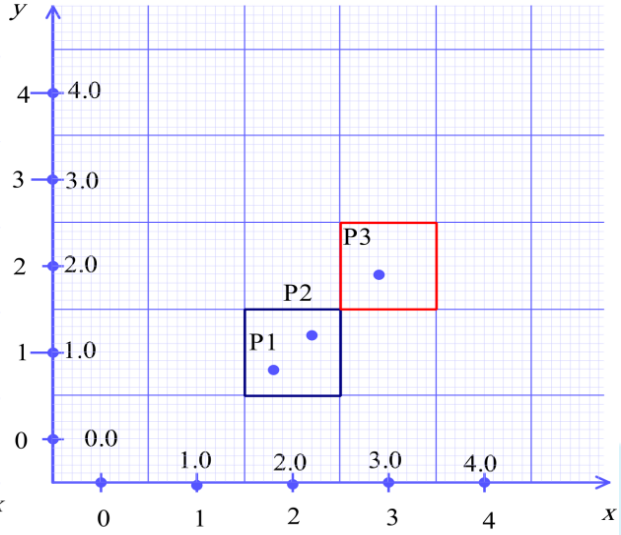
أي يحدث تقريب العدد الحقيقي x و y إلى أقرب عدد صحيح لهما.

وهنا تصبح نقطة الأصل لنظام الإحداثيات $(x\ ,\ y)$ عند مركز البكسل $(0,0)$ وهذا يعني بأن النقطتين $p_1$ و $p_2$ تمثلان ضمن البكسل $(2,1)$ في حين النقطة $p_3$ تقع في البكسل $(3,2)$ أي:

$$x` = Round(x) = floor(x + 0.5) = \text{int}(x + 0.5)$$
$$y` = Round(y) = floor(y + 0.5) = \text{int}(y + 0.5)$$




طريقة 1




طريقة 2

نستخدم هنا الطريقة الثانية ضمن شبكة نظام الإحداثيات وبالتالي كل البكسلات متمركزة عند قيم صحيحة للنظام إحداثيات مستمر.

❖ **تحويل المسح لمستقيم** $y = mx + b$ :

عندما ننقل الإحداثيات الحقيقية لنقط من نظام إلى ما يقابلها في نظام آخر (البكسلات) نحصل على خط درجي.

مثلاً: تصبح النقطة $p\,(10.48, 20.51)$ بعد المسح $(10, 21)$

إن إسقاط القيم العشرية يجعلنا نعاني من الانزياحات عن المواقع الحقيقية، والتدوير إلى رقم صحيح يسبب أن كل الخطوط الأفقية والعمودية يتم إظهارها بشكل درجات السلم ويظهر هذا العيب في الأنظمة ذات دقة التمييز المنخفضة، وإذا رفعنا دقة التمييز نستطيع تحسين هذا المظهر أو عن طريق ضبط كثافة البكسل على طول الخط المستقيم.

## معادلة المستقيم:

$$y = m\,x + b$$

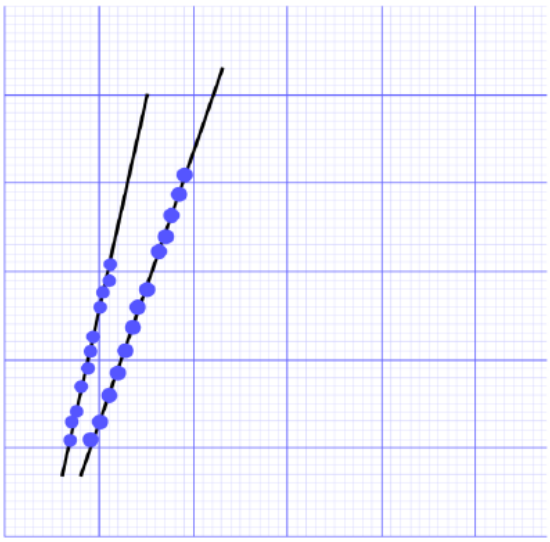
نفرض النقطة الأولى $(x_0, y_0)$ والنقطة الأخيرة $(x_{end}, y_{end})$

$$y_0 = m\,x_0 + b$$

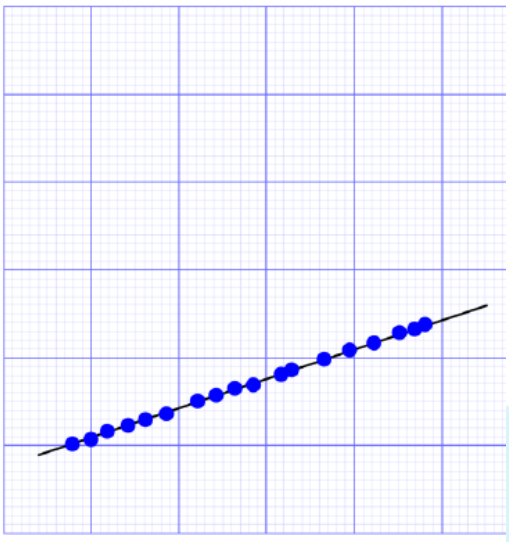
$$y_{end} = m\,x_{end} + b \longrightarrow y_{end} - y_0 = m\,(x_{end} - x_0)$$

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{end} - y_0}{x_{end} - x_0}$$

$$b = y_0 - m\,x_0$$



slope Greater than 1
$|m| \geq 1$



slope less than 1
$|m| < 1$

1. ننطلق من النقطة الأولى $p_1(x_0, y_0)$

2. نحسب كل من $d_x$ و $d_y$ ونرسم البكسل الأول $p_1$

3. إذا كان $d_x$ لا يساوي الصفر، نحسب الميل $m = \frac{\Delta y}{\Delta x}$

ثم نحسب $b = y_0 - m\, x_0$

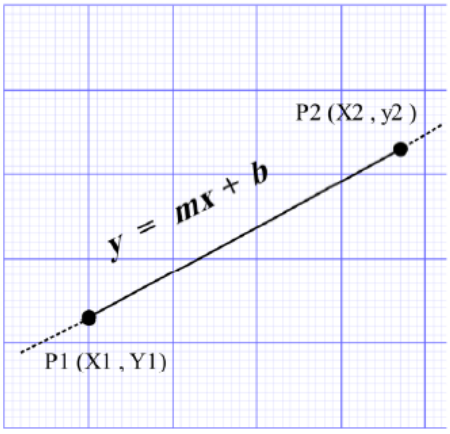
4. نكرر عمليات الحساب طالما أن $x_0 < x_1$

$$x_0 = x_0 + d_x \quad .I$$

$$y_0 = Round(m\, x_0 + b) \quad .II$$

**أبسط خوارزمية لرسم المستقيم:**

إن الطريقة البسيطة لتحويل المسح لمستقيم تبدأ بتحويل مسح $p_1$ و $p_2$ إلى إحداثيات البكسل $(x_1, y_1)$ و $(x_2, y_2)$ على التوالي.




1. نحسب الميل $m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$

2. نحسب $b$ لأجل النقطة الأولى المعطاة

$$:b = y_1 - m\, x_1$$

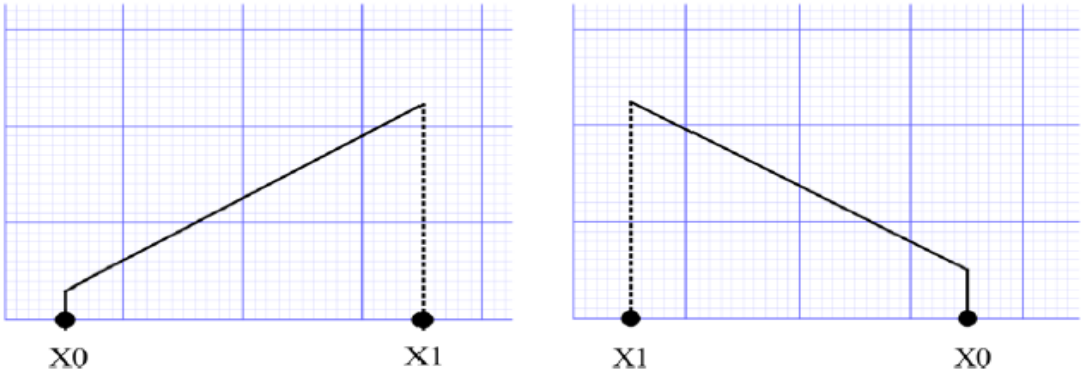
- فإذا كان الميل $m \leq 1$ نحسب القيمة المقابلة لـ $y$ عن طريق زيادة $x$ بواحد، وطالما أن $x$ عدد صحيح بين $x_1$ و $x_2$ ولا يساويهما أي $x \in ]x_1, x_2[$

- أما إذا كان الميل $m \geq 1$ نحسب $x$ المقابلة لـ $y$ باستخدام المعادلة وتحويل مسح النقطة التي نحصل عليها عن طريق $x_0 = \frac{(y_0 - b)}{m}$

أو $x_0 = Round\ (y_0 - b) * \frac{1}{m}$

ونكتب الخوارزمية كما يلي:

```
void  Line( int x0 , int y0 , int x1 , int y1 )
{ int dx=x1-x0;
  int dy=y1-y0;
  writePixel (x0 , y0);
  if ( dx != 0 )
  { float m = (float) dy / (float) dx;
     float b =  y0 – m * x0;
     dx = ( x1 > x0 ) ? 1 : -1;
     while ( x0 != x1 )
      { x0 += dx;
        y0 = Round (m * x +b);   writePixel(x0 , y0);}}}
```

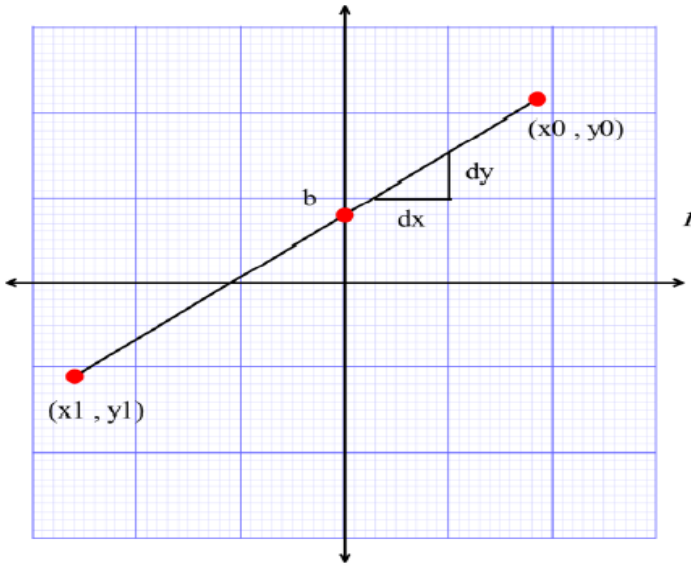



السيئات:

- القيام بعملية جداء وعملية قسمة على أعداد عشرية من أجل كل نقطة من نقاط المستقيم.
- استدعاء التابع Round لتحويل القيمة إلى عدد صحيح.

من نقاط المستقيم.

- استدعاء التابع Round لتحويل القيمة إلى عدد صحيح.




$$m = \frac{dy}{dx}$$