بسم الله الرحمن الرحيم

جامعة السودان المفتوحة

برنامج علوم الحاسوب وتقنية المعلومات

تخطيطات الحاسوب

رمز المقرر ورقمه: حسب1030

تأليف : د. سامي محمد شريف

التحكيم العلمي : أ.د. عزالدين محمد عثمان

التصميم التعليمي : أ. سلمى مأمون عبد النور

التصميم الفني : أماني الأمين مبارك

منشورات جامعة السودان المفتوحة، الطبعة الأولى 2006

جميع الحقوق محفوظة لجامعة السودان المفتوحة، لا يجوز إعادة إنتاج أي جزء من هذا الكتاب، وبأي وجه من الوجوه، إلا بعد الموافقة المكتوبة من الجامعة.

مقدمة المقرر

بسم الله الرحمن الرحيم والصلاة والسلام على خاتم الأنبياء والمرسلين سيدنا محمد وعلى آله وصحبه أجمعين وبعد

عزيزي الدارس تعلمت من خلال دراستك لمقررات الحاسوب أن **الحاسوب يعرف** كنظام معلومات الكتروني Electronic Information System EIS، يؤدى مهام قبول البيانات ومعالجتها ونقلها وعرضها وتخزينها واسترجاعها والتي تمثل معلومات معينة. في هذا الكتاب سوف نتعرض لموضوع **التخطيط بالحاسوب حيث** يعتبر التخطيط بالحاسوب أحد تطبيقات الحاسوب ويهتم بتقنيات قبول البيانات ومعالجتها ونقلها وعرضها وتخزينها واسترجاعها على نسق مرئي. يبدأ هذا الكتاب **بتعريف تخطيط الحاسوب** كعتاد hardware وبرمجيات Software تستخدم لمعالجة البيانات المرئية، **فدور العتاد في التخطيط** دور حيوي في ضبط مقدرات نظم الحاسوب التخطيطية، حيث إنه يحدد درجة الاتصال الفعال بين المصمم والحاسوب. فأكثر نظم تخطيط الحاسوب تستخدم **عتاد نظم الحاسوب المعياري** مع **برامج تخطيط خاصة**، أما **دور الحاسوب في التخطيط** فإن الحواسيب تقوم في الوقت الحالي بدور فعال وهي تعتبر من الأدوات الأساسية في عمليات التصميم و التخطيط البياني.

عزيزي الدارس نجد أن الإنسان قد مارس **منذ القدم** عمليات التعامل مع البيانات المرئية، **وقد يؤرخ لبداية** هذه العمليات في العام 2000 قبل الميلاد حين استخدم **الإسقاط المتعامد** orthographic projection. في العام 1400 قبل الميلاد الذي أعطى **ازدهار الحركة الفنية الرومانية** دافعا قويا لتطوير مجال التخطيط والرسومات عموما. و في العام 1600 قبل الميلاد كان لاستخدام **نظام الإحداثيات وعلم الحسبان والفيزياء والضوء** آثاراً كبيرة في هذا المجال، وقد كان **راسم الإشارة** oscilloscope الذي صنع في العام 1897 أول جهاز تخطيط الكتروني، وخلال الأعوام 1950 وحتى 1970 استخدمت **الحواسيب** مع **العارض المتجه** بكثرة، وفى العام 1966 تم تصنيع أول

عارض "**ماسح خطوط**" ثم تطور مجال التخطيط بالحاسوب منذ ذلك التاريخ هذا وتوجد اليوم كثير من تقنيات التخطيط الفعّالة بالحاسوب.

تغطى تطبيقات الرسم بالحاسوب مجالات كثيرة مثل الأفلام، و العاب الحاسوب، و التصميم الصناعي، و التصوير العلمي، التصوير الطبي و التصميم بالحاسوب و الربط البياني المستخدم وقد كان لكل **هذه المجالات** دور كبير في **تطوير التخطيط بالحاسوب في الفترة الأخيرة.**

عزيزي الدارس **العناصر الرئيسة للتخطيط بالحاسوب هي النمذجة، و عوامل الإضاءة الحركة و التفاعل، فالنمذجة تشتمل على بدائيات primitives التخطيط** : والتي تمكّن المبرمج من رسم النقاط والخطوط والمنحيات والأسطح .... الخ، و **الصفات التي تحدد إشكال المشهد** من الألوان وخواص الإضاءة وغيرها ، و**التحويل الهندسي** والذي يستخدم لنقل حجم الكائن ودورانه وتغيره ،**عوامل الإضاءة وتشتمل على الرؤية** وهى العرض بكفاءة ووضوح ،و **محاكاة انتشار الضوء** من انعكاس وانكسار وانتشار ... الخ،

يعتبر هذا المقرر عزيزي الدارس مدخلا لتخطيط الحاسوب وسنتعرف خلاله على بدائيات الرسم ثنائية الأبعاد، وتحويلات وعمليات مشاهدة الكائن الثنائي الأبعاد و قص الكائن ليناسب حجم الشاشة، وتمثيل عمليات المشاهدة في الفضاء ثلاثي الأبعاد،بالإضافة إلى بعض الموضوعات المتقدمة الأخرى،وسيعطى المقرر أيضا الدارس خلفية كافية **لكتابة تطبيقات تخطيط** الحاسوب باستخدام مكتبة التخطيط OpenGL، لغة البرمجة C.

أخي الدارس نرجو منك المراجعة الدقيقة المتقنة لما درست في مجالات الجبر الخطي، والمصفوفات، وحساب المثلثات، و الهندسة التحليلية، و الحسبان، بالإضافة للمعرفة بتقنيات عتاد وبرامج الحاسوب، ومعرفة عمليات معالجة الإشارة الرقمية، حيث إنها أساسيات لما سوف تدرس في هذا المقرر هذا ونذكرك بأن كل الفصول مزودة بمجموعة من الأمثلة و التدريبات المنَّوعة التي تساعد على فهم المادة وأسئلة

التقويم الذاتي حتى تتأكد من فهمك واستيعابك للمقرر **نتمنى لك** وقتاً طيباً مع هذا الكتاب **الذي تضمن عشر وحدات وهي كما يلي:**

الوحدة الأولى : عتاد الرسم (التخطيط) بالحاسوب

الوحدة الثانية : الدخل والخرج وبرمجيات التخطيط

الوحدة الثالثة : المناظر ثنائية الأبعاد

الوحدة الرابعة: عمليات مشاهدة المناظر ثنائية الأبعاد

الوحدة الخامسة: التحويلات الهندسية ثلاثية الأبعاد

الوحدة السادسة :ألإسقاط والقص

الوحدة السابعة :الرؤية و نماذج الإضاءة

الوحدة الثامنة:المنحنيات و الأسطح

الوحدة التاسعة:مواضيع مختارة

الوحدة العاشرة:أمثلة تطبيقية لاستخدام مكتبة التخطيط

الأهداف العامة للمقرر:

عزيزي الدارس، بعد دراستك لهذا المقرر نتوقع منك أن تكون قادراً على أن:

- **تفرق** بين تقنيات العرض المختلفة.
- **تتعرف** نظم تخطيطات الحاسوب.
- **تتعرف** أهمية أجهزة الدخل والخرج وبرمجيات التخطيط
- **تمثل** المناظر الثنائية الأبعاد.
- **تنجز** عمليات مشاهدة المناظر الثنائية الأبعاد.
- **تجري** التحويلات الهندسية الثلاثية الأبعاد.
- **تستخدم** الإسقاط والقص.
- **تطبق** خوارزميات الرؤية وقوانين نماذج الإضاءة.
- **تكتب** معادلات المنحنيات والأسطح.
- **تصف** الإستراتيجيات المختلفة لأخذ العينات.
- **تطبق** على مكتبة التخطيط OpenGL.

محتويات المقرر

الوحدة	الموضوع	الصفحة
1	عتاد الرسم (التخطيط) بالحاسوب Computer Graphics Hardware	**1**
2	الدخل والخرج وبرمجيات التخطيط Output and Input and Graphics Software	**29**
3	المناظر الثنائية الأبعاد Two Dimensional Views	**55**
4	عمليات مشاهدة المناظر الثنائية الأبعاد Two-Dimensional Viewing Operations	**91**
5	التحويلات الهندسية الثلاثية الأبعاد Three Dimensional Geometric Transformation	**113**
6	ألإسقاط والقص Projection and Clipping	**145**
7	الرؤية و نماذج الإضاءة Visibility and Illumination Models	**173**
8	المنحنيات و الأسطح Curves and Surfaces	**195**
9	مواضيع مختارة Selected Topics	**219**
10	أمثلة تطبيقية لاستخدام مكتبة التخطيط OpenGL	**239**


الوحدة الأولى
عتاد الرسم (التخطيط) بالحاسوب
Computer Graphics
Hardware

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**3**
تمهيد	3
أهداف الوحدة	3
1. عتاد تخطيط الحاسوب:	**4**
1.1.تصنيف عتاد تخطيط الحاسوب:	4
2.1 عمليات التخطيط بالحاسوب	4
3.1 العوامل المستخدمة في التخطيط بالحاسوب:	4
4.1 تطور تقنيات التخطيط :	5
2. تقنية العرض ونظم التخطيط	**5**
1.2 أنبوب الأشِعّة المهبطية	7
2.2 أنبوب التخزين	10
3.2 عارضات تجديد المتجه	12
4.2 عارضات خُطوط المسح	14
5.2 عارضات خُطوط المسح الملونة	17
6.2 عارضات اللوحات المستوية	20
7.2 أجهزة الدخل	21
الخلاصة	**25**
لمحة مسبقةعن الوحدة الدراسية التالية	**25**
إجابات التدريبات	**26**
مسرد المصطلحات	**27**
المراجع	**28**

المقدمة

تمهيد

عزيزي الدارس مرحباً بك ، فلنبدأ معاً بالوحدة الأولى من مقـرر تخطيطـات الحاسوب التى أوردنا فيها تصنيف عتاد تخطـيط الحاسـوب، والعمليـات والعوامـل المستخدمة في التخطيط بالحاسوب وأهميتها، كما تطرقنـا لتطـور تقنيـات التخطـيط، وشرحنا بالتفصيل تقنية العرض ونظم التخطيط من حيـث العوامـل التـي تحـدد أداء العارض و الاستبانة، والمقدرات التفاعلية ونظم العرض الأساسية التى شـملت أنبـوب الأشِعّة المهبطية ،أنبوب التخزين، عارضات تجديد المتجه، وعارضات خُطوط المسح ، عارضات خُطوط المسح الملونة، وعارضات اللوحات المسـتوية، وعـارض البلازمـا وأخيراً استعرضنا أجهزة الدخل وخصصنا الفـأرة بالتفصـيل لـدورها الفاعـل فـي تخطيطات الحاسوب.

وقد أوردنا في هذه الوحدة مجموعة من التدريبات وأسئلة التقويم الذاتي نتمنى أن تقـوم بحلها بعد دراستك لهذه الوحدة لتصقل معرفتك بعتاد الرسم بالحاسوب كمـا نرجـو أن تساهم معنا في نقدها وتقويمها.

أهداف الوحدة

عزيزي الدارس، بعد دراستك هذه الوحدة ينبغي أن تكون قادراً على أن:

(1) **تصنف** عتاد تخطيطات الحاسوب.

(2) **توضح** العوامل المتداخلة والمهمة للتخطيط بالحاسوب

(3) **تصف** نظم العرض الأساسية.

(4) **يتعرف** على أجهزة الدخل لنظم التخطيط بالحاسوب.

1. عتاد تخطيط الحاسوب

1.1 تصنيف عتاد تخطيط الحاسوب

يمكن **تصنيف عتاد تخطيط الحاسوب** بصورة عامة إلى **أربعة نظم فرعية** هي :

- نظم الدخل.
- نظم التخزين.
- نظم الإرسال.
- نظم الخرج.

2.1 عمليات التخطيط بالحاسوب

تتم هذه العمليات على النحو التالي :

أولاً: تحويل معلومات التخطيط إلى الحاسوب باستخدام أحد أجهزة الدخل المتخصصة المختلفة.

ثانياً: تخزين هذه المعلومات في نسق معين يعتمد على نوعية وخواص الجهاز المستخدم.

ثالثاً: إرسال هذه المعلومات لجهاز الخرج المناسب للعرض أو المشاهدة.

3.1 العوامل المستخدمة في التخطيط بالحاسوب:

يوجد الكثير من **العوامل المتداخلة والمهمة للتخطيط بالحاسوب** ومن هذه العوامل :

1) **التقنيات** المختلفة **لإنتاج** المنظر ونوعيته.
2) **عرض** الألوان و**التمثيل** الفضائي.
3) **الاختلافات** في **نوعية الوسائط** المستخدمة.
4) **سرعة** خواص **جهاز الخرج** (العرض).

❖ **أهمية العوامل المستخدمة في التخطيط بالحاسوب:**

وقد تستخدم هذه العوامل المتداخلة **لتميز** عتاد نظم التخطيط بالحاسوب **عن** عتاد الحاسوب المعياري.

4.1 تطور تقنيات التخطيط :

أدى التطور في **نظم المعالج الدقيق** microprocessors وتميزه **بالتكلفة المنخفضة** منذ بدايات **ثمانينيات** القرن الماضي إلى تطور كبير في **تقنيات التخطيط،** حيث مكن من تصميم **نظم عتاد فرعية** يمكن استخدامها **كمسرعات للتخطيط** graphics accelerators، وقد أصبح من الممكن تنفيذ **دوال التخطيط الثلاثية الأبعاد** على مستوى **العتاد.**

2. تقنية العرض ونظم التخطيط

display Technology and Graphics System

تمثل **طرفية العرض** في نظم تخطيط الحاسوب **الرابط الأساسي** بين المستخدم والنظام. **تستخدم** تقنيات مختلفة **لتصميم وتصنيع** أجهزة عرض التخطيط. فالشكل 1.1 يوضح **تصنيف أجهزة العرض المختلفة** هذا وكل واحد من هذه الأجهزة لديه **ميزات وقصور** يجب أن **تؤخذ في الاعتبار عند اختيار** عارض **لتطبيق معين.**

❖ **العوامل التي تحدد أداء العارض:**

يحدد أداء العارض بعوامل مختلفة منها :

1) السعر.
2) الاستبانة resolution.
3) المقدرات التفاعلية.

❖ **الإستبانة:**

تحدد **مقدرة العارض** على **عرض التفاصيل، وتصل استبانة** النظم المستخدمة حاليا إلى حوالي **1280x1024** وتشير هذه الأرقام إلى :

1) **عدد النقاط** على الشاشة.

2) **العدد الكلى للألوان** التي يدعمها العارض.

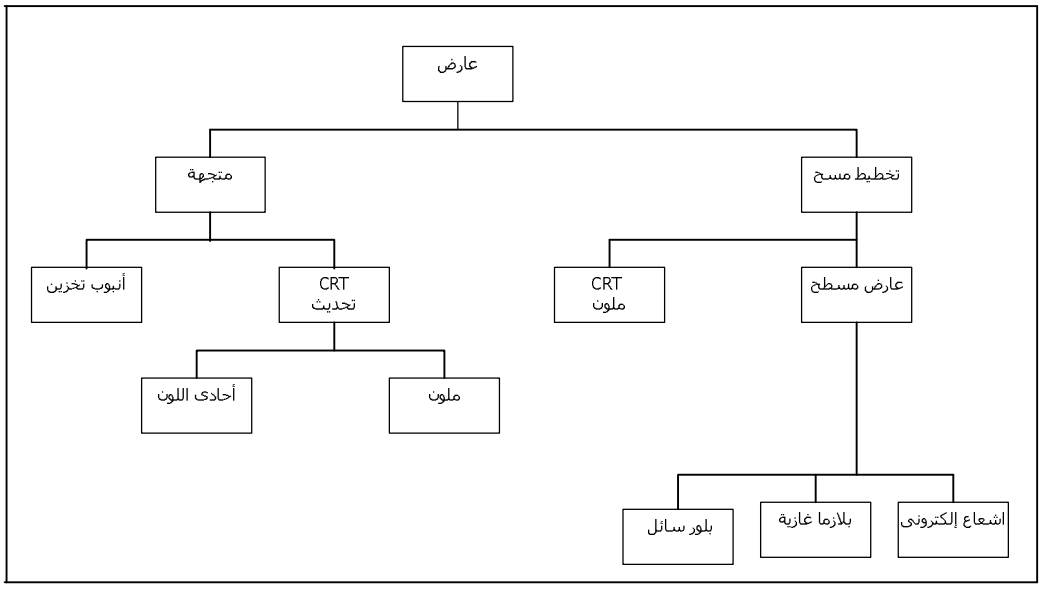
3) **عدد ألألوان** التي يمكن عرضها في نفس الوقت.

أيضاً من **العوامل الهامة** التي يجب أخذها في الاعتبار، **تراوح عدد الألوان** التي **يمكن عرضها** في نفس الوقت على **التوازي من 8 إلى 4096 لون** من **العدد الكلى للألوان** التي قد تصل إلى حوالي **16 مليون لون مختلف**.

❖ **المقدرات التفاعلية:**

تحدد خواص العارض التالية:

- **المقدرة** على **إدارة** جهاز دخل متفاعل.
- **مجال الحركية (مقدرة** النظام لتوفير حركة **مثيرة)**.



الشكل 1.1: التقنيات المستخدمة في أجهزة عرض التخطيط

أسئلة تقويم ذاتي

(1) يمكن تصنيف عتاد تخطيط الحاسوب إلى أربعة نظم فرعية ما هي؟
(2) عرف مسرعات التخطيط graphics accelerators
(3) ما الذي تمثله طرفية العرض في نظم تخطيط الحاسوب ؟
(4) تصل استبانة نظم العرض المستخدمة حاليا إلى حوالي 1280x1024 فإلى ماذا تشير هذه الأرقام ؟
(5) ماذا نعني بالمقدرات التفاعلية ؟

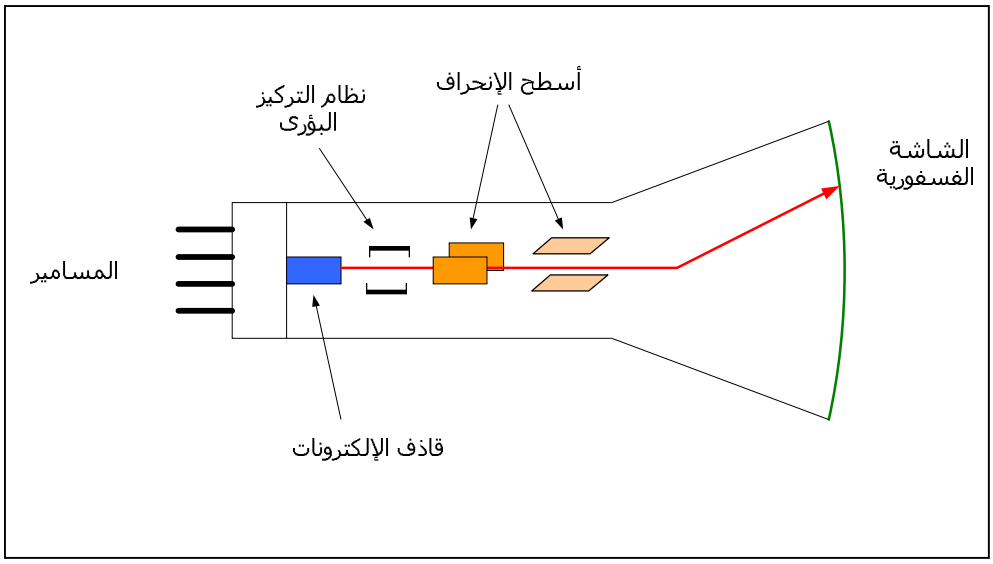
1.2 أنبوب الأشِعّة المهبطية Cathode Ray TubSine

يعتبر أنبوب الأشعة المهبطية أحد نظم العرض الأساسية والمستخدمة بكثرة في الوقت الحالي في نظم الحاسوب التخطيطية.

- **تعريف أنبوب الأشِعّة المهبطية Cathode Ray Tube:**

عبارة عن أنبوب زجاجي مفرغ ينتهي بشاشة مخروطية الشكل.

- **المكونات الأساسية لهذا الأنبوب هي : (أنظر الشكل 1.2)**
  - القاعدة والتي تحتوى على مسامير التوصيل
  - قاذف الإلكترونات
  - نظام تركيز بؤري،
  - أسطح الانحراف الأفقي والرأسي
  - شاشة فسفورية



الشكل 1.2: أنبوب الأشعة المهبطة

❖ **إظهار المنظر على الشاشة:**

لإظهار المنظر يبعث قاذف الإلكترونات **شعاع** beam يمر عبر **نظام الانحراف** و **نظام التركيز البؤري** قبل وصوله **لموقع معين** على **الشاشة** الفسفورية، ونسبة **لعدم وجود وزن للإلكترونات** يمكن **أعادة ضبط** موقع هذا الشعاع **بسرعة** فائقة.

❖ **أثر تغطية شاشة الأنبوب بمادة فسفورية:**

تصنع مادة الفسفور من **مزيج** من مواد مثل الكبريت والسيلكون والفضة، وتغطى شاشة أنبوب الأشعة المهبطية بمادة فسفورية **تتوهج** عندما تضرب بواسعة الشعاع الإلكتروني **وتحدث نقطة إضاءة**، والضوء المبعوث من الفسفور **يتلاشى بسرعة عالية** عندما يتحول الشعاع من موقع لأخر على الشاشة، **ولمشاهد منظر مستمر** على الشعاع الإلكتروني تتم **إعادة تخطيط** (تحديث) الصورة على الشاشة **في نفس الموقع**.

❖ **زمن التوهج الفسفوري:**

الفترة الزمنية التي يكون خلالها الفسفور متوهجاً تسمى **بفترة استمرار الأثر** Persistence **وتساوى** الزمن المطلوب للاضمحلال إلى عشر الشدة الأولية.

وتختلف قيمة هذا العامل باختلاف أنواع الفسفور، كما **يعتمد** على مكونات الفسفور كما هو موضح بالجدول 1.2.

ويلاحظ هنا أنه:

لتخزين المعلومات على الشاشة يستخدم فسفور لدية استمرار أثر طويل مثل P7 من الجدول، **وللحصول على سرعة عرض سريعة** يستخدم فسفور لدية فترة استمرار أثر قصيرة مثل P31 أو P4.

نوع الفسفور	اللون	استمرار الأثر
P1	أصفر – أخضر	24 msec
P2	أصفر – أخضر	35-100 µsec
P4	أبيض	60 µsec
P7	أخضر/أبيض	300 µsec
P22-B	أزرق	22 µsec
P31	أخضر	40 µsec
P39	أصفر – أخضر	150 µsec

جدول 1.1: أنواع الفسفور المستخدمة في أجهزة العرض

للمحافظة على **شدة إضاءة** جزء من الشاشة لفترة طويلة، يجب **تنشيط أو تحديث الفسفور** في هذا الجزء بصورة **دورية** ومستمرة بواسطة **الشعاع الإلكتروني.**

قد ويؤدى **عدم التحديث** في أوقات مناسبة إلى حالة تعرف **بالارتعاش flicker** والتي تجعل **المنظر يضمحل** بين دورات التحديث على الشاشة **بصورة مستمرة.**

2.2 أنبوب التخزين Storage Tube

أنبوب التخزين وان كان لا يصنع في الوقت الحالي فقد كان له أثر كبير في **تطوير نظم تخطيط الحاسوب،** ويستطيع هذا الجهاز **الاحتفاظ** بالمنظر على الشاشة دون حاجة إلى **تحديث.**

❖ **أنواع القاذفات التي يستخدمها أنبوب التخزين :**

يستخدم أنبوب التخزين قاذفات إلكترونيات مختلفة وهى: (الشكل 1.3)

- **قاذفات الفيضان flooding guns** والتي تضرب الشاشة باستمرار بالإلكترونيات لتنشيط الفسفور في حده الأدنى.
- **قاذف كاتب الضربة الواحدة stroke guns** والذي ينشط الفسفور إلى مستوى أعلى من الحد الأدنى.

❖ **فائدة قاذفات الفيضان:**

التشغيل المستمر لقاذفات الفيضان تجعل المنظر يشاهد باستمرار على الشاشة ولوقت طويل دون تحديث، ويتم مسح المشهد من الشاشة برفع الجهد في كل الشاشة.

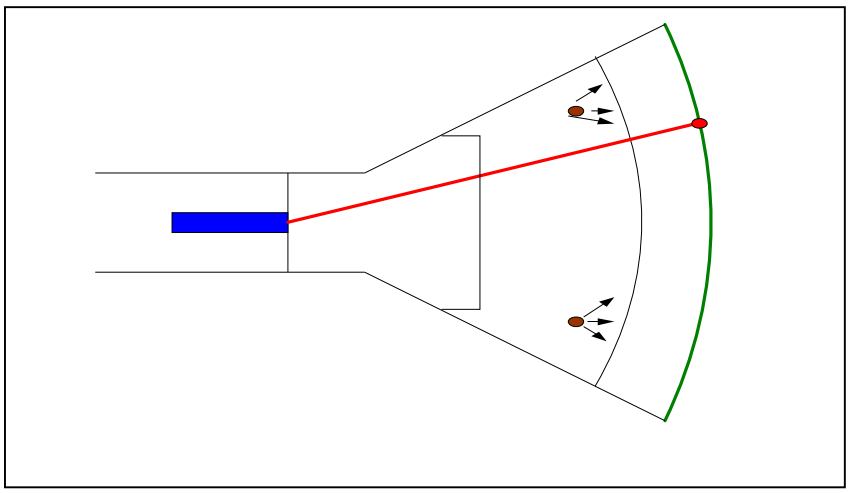
❖ **ميزات أنبوب التخزين وأوجه القصور فيه:**

ومن أهم ميزات هذا النظام:

1) التكلفة المنخفضة الناتجة من عدم استخدام نظام تحديث إلكتروني معقد.
2) يعرض أنبوب التخزين كمية كبيرة من البيانات في نفس الوقت.
3) يعطى درجة استبانة قد تصل إلى 4096x4096.

يعانى هذا النظام من قصور في:

1) مقدرات التلوين.
2) مقدرته على إنتاج عرض ساكن.



الشكل 1.3: أنبوب التخزين

أسئلة تقويم ذاتي

(1) عرف أنبوب الأشِعّة المهبطية Cathode Ray Tube

(2) لماذا تغطى شاشة أنبوب الأشعة المهبطية بمادة فسفورية؟

(3) ما السبب في حدوث حالة الارتعاش flicker ؟

(4) اذكر أنواع القاذفات التي يستخدمها أنبوب التخـــزين Storage Tube ؟

(5) من مزايا أنبوب التخزين مقدرته على إنتاج عرض ســـاكن ( ) ومن عيوبه أنه يعطى درجة استبانة قد تصل إلى 4096x4096 ( ) بين صحة أو خطأ المعلومات السابقة .

تدريب رقم (1)



رتب القائمة (أ) مع ماينا سبها من القائمة (ب)

القائمة (أ)	القائمة (ب)
لتخزين المعلومات على الشاشة	الارتعاش
سرعة عرض سريعة	P7
المحافظة على شدة إضاءة جزء من الشاشة	P_{31} أو P_4
اضمحلال المنظر على الشاشة	تنشيط أو تحديث الفسفور

3.2 عارضات تجديد المتجه Vector Refresh Displays

ينتج عارض تحديث المتجه المشهد **بتوجيه شعاع إلكترونيات** إلى نقاط عشوائية على الشاشة ويتم ربط هذه النقاط على نمط متجهات.

❖ **نوع الفسفور المستخدم :**

يستخدم في هذا العارض عادة **فسفور يتصف بانخفاض** فترة استمرار أثر منخفضا لغطاء شاشة أنبوب الأشعة المهبطية. **ولتجنب الارتعاش** تحدث الشاشة مرات عديدة في الثانية.

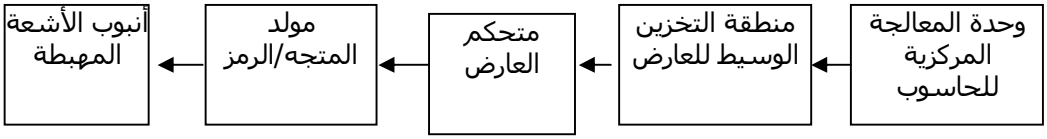
❖ **المتحكم و منطقة التخزين الوسيطة :**

بالإضافة إلى أنبوب الأشعة المهبطية يحتاج هذا النظام إلى **منطقة تخزين وسيطة**، لحفظ المعلومات المطلوبة لتكوين الصورة .

و**متحكم** ليرسل هذه المعلومات إلى **مولد المتجهات** والذي يقوم بدوره بتحويلها إلى موقع الشعاع الإلكتروني على الشاشة.

تعتمد هذه العملية بدرجة كبيرة على **سعة الخازنة** الوسيطة **وسرعة المتحكم.**

(الشكل 1.5 يوضح المخطط الصندوقي لعارض تجديد المتجه).



الشكل 1.5: نظام تحديث المتجه

❖ **تعديل عناصر المشهد في الذاكرة:**

يسمح عارض تجديد المتجه **بتعديل** إي من عناصر المشهد في الذاكرة الوسيطة، وحيث إن الفسفور المستخدم لديه فترة استمرار أثـر قصـيرة، يمكـن اسـتخدام دورة التحديث لتعديل المنظر الأول.

وهذه **الخاصية** تجعل هذا العارض مناسبا في **التطبيقات ذات الحركة المستمرة.**

- **مزايا عارض تجديد المتجه:**

ينتج عارض تحديث المتجه صورة:

أ/ زاهية وواضحة. ب/ ذات استبانه عالية.

- **عيوبه :**

يعانى هذا العارض من:

أ/ التكلفة العالية . ب/ عدم المقدرة على عرض المناطق الكثيرة الألوان.

- **استخدامه:**

استخدم عارض تجديد المتجه حتى منتصف **الثمانينيات** من القرن الماضي عندما بدأ **يفضل** عليها **عارض خطوط المسح.**

4.2 عارضات خُطوط المسح Raster Scan Display

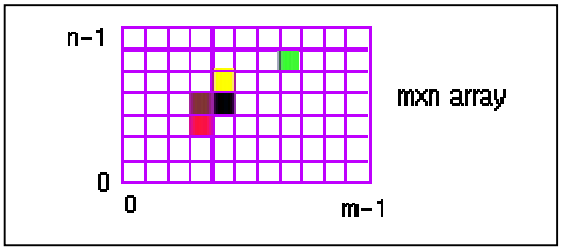
استخدمت عارضات خطوط المسح والمعتمدة على **تقنيات التلفزيون** في **بداية السبعينات** من القرن الماضي، حيث أعطت مجال تخطيط الحاسوب دفعة قوية، وخلافا لنظم العرض التي تم التعرض لها سابقا والتي **تصنف باعتبارها أجهزة راسمة للخطوط** يصنف عارض خطوط المسح **باعتبارها جهازاً راسماً للنقاط.**

❖ **شاشة عارض خطوط المسح :**

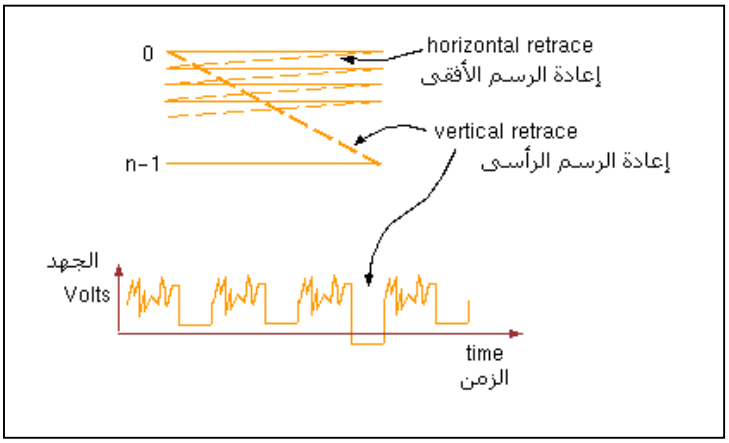
تقسم شاشة عارض خطوط المسح إلى **مصفوفة من النقاط xy** ، تعرف كل منها **بعنصر الصورة** pixel وتتكون الصورة من عناصر الصورة هذه وذلك بجعلها **نشطة أو خاملة**، انظر (الشكل1.6).

❖ **تكون المشهد على الشاشة :**

- ♦ يتحرك شعاع الإلكترونيات في أنبوب الأشعة المهبطة أفقيا بدء من الركن الأيسر في أعلى الشاشة ويستمر من الشمال إلى اليمين في مستويات رأسية أفقية (مائلة)، ثم يعود سريعا من اليمين إلى اليسار. (الشكل 1.7).
- ♦ وتعاد هذه العملية حتى يتم تغطية كل الشاشة، عندها يعود الشعاع إلى الركن الأيسر أعلى الشاشة ويعاد المسح، فيتكون المشهد على الشاشة بتغير شدة الإشعاع إثناء تنقله.
- ♦ إن كان الموقع جزء من المشهد مضاء يمر الشعاع عليه بالشدة القصوى ويعود إلى الحد الأدنى من الشدة إن كان الموقع مظلماً.



الشكل 1.6: مصفوفة عارض خطوط المسح



الشكل 1.7: أشارات المرئي في عارض ماسح الخطوط

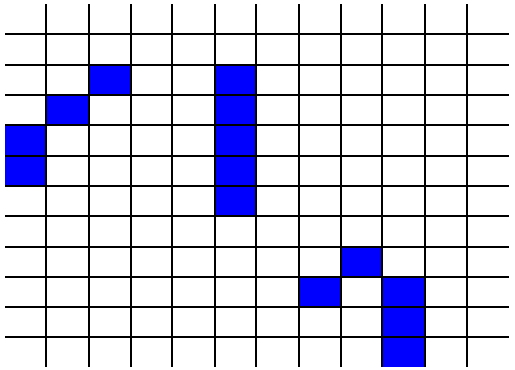
❖ محدودية عارض خطوط المسح :

تتمثل محدودية هذا العارض في الآتي :

- أن كل عنصر صورة يجب التعامل معه منفرداً.
- خلال تحرك الشعاع عبر العارض يتسلم معلومات عن مستوى الشدة المطلوبة في النقاط المختلفة لتكوين الصورة.
- يخزن مستوى التضمين لكل عنصر صورة في ذاكرة رقمية تعرف باسم الخازنة الوسيطة للتجديد أو الأطر (الشكل 1.8).
- عندما يتحرك الشعاع يتسلم متحكم من البيانات هذه الذاكرة ويحولها لجهد تماثلي ثم ترسل إلى مضخم شدة الشعاع.

- كل عنصر صورة يحتاج لموقع خاص به في الذاكرة مما يتطلب حجم ذاكرة كبير.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0



عناصر الصورة الثنائيات في الذاكرة (خازنة التجديد الوسيطة)

الشكل 1.8: تكوين خازنة الأطر الوسيطة

❖ مزايا عارضات خطوط المسح:

تمتاز عارضات خطوط المسح:

- ✓ بالمقدرة على تكوين صورة زاهية وواضحة المعالم وخاصة عندما تستخدم الألوان.
- ✓ تطورت استبانه هذه النظم في السنوات الأخيرة بصورة كبيرة مما جعلها تقترب من أداء عارضات تجديد المتجه.
- ✓ لا تتأثر عارضات خطوط المسح بتعقيدات المشهد ويمكنها عرض المشاهد المتحركة.

أسئلة تقويم ذاتي

(1) ما نوع الفسفور المستخدم في عارض تجديد المتجه ؟
(2) على ماذا تعتمد عملية تكوين الصورة في عارض تجديد المتجه ؟
(3) خاصية تعديل عناصر المشهد في ..... جعلت عارض تجديد المتجه مناسبا في التطبيقات ..... (أكمل المكان الخالي).
(4) اعتمدت عارضات خطوط المسح على: أ/ تقنيات الحاسوب الرقمي. ب/ تقنيات التلفزيون. ج/ تقنيات الحاسوب المعياري.(اختر الإجابة)
(5) في ماذا تتمثل محدودية عارضات خطوط المسح .

5.2 عارضات خُطوط المسح الملونة Color Raster Display

تعمل عارضات خطوط المسح الملونة بصورة **مشابهة** للعارضات الأحادية اللون monochrome، **وتختلف عنها** باستخدامها مجموعة من الألوان الفسفورية لتكوين المشهد، و تستخدم هذه العارضات **ثلاثة إشعاعات** لإضاءة ثلاثة **أنواع** مختلفة من الفسفور وهى **الأحمر والأخضر والأزرق** RGB. و**بتغير شدة** هذه ألألوان الأساسية يمكن إنتاج تتدرج كبير من الألوان.

❖ **قناع الظل:**

عادة تستخدم ثلاث قاذفات الكترونات في عارضات خطوط المسح الملونة وتنظم على هيئة مثلث، وتنظم النقاط الحمراء، والخضراء، والزرقاء في الشاشة على نفس النسق.

وتضع **شبكة معدنية مثقبة** بين قاذفات الإلكترونات والفسفور تعرف **باسم ستار (قناع) الظل**، بحيث **يسمح** لكل قاذف **بإثارة الفسفور** المقابل له **فقط.**

إن الثقوب الموجودة في الستار المعدني لا تسمح للشعاع بالسقوط في أكثر من نقطة فسفورية واحدة، و ينتج التغير في شدة الشعاع مستويات مختلفة من تتدرج اللون الأساسي مما يعطى تدرجاً كبيراً للأوان.

العلاقة بين عدد أسطح الثنائيات وعدد الألوان:

عرض الألوان على شاشة أنبوب الأشعة المهبطية يتطلب تعديل خازنة الأطر الوسيطة، و كما ذكر سابقا، إن كل عنصر صورة لديه موقع في هذه الذاكرة، ففي **النظام الأحادي اللون** يحتوى هذا الموقع على **قيمة الثنائية 0 أو 1**، والتي تحدد **شدة الشعاع** الساقط على عنصر الصورة، ويعرف هذا **بسطح الثنائية bit surface**.

لعرض **الألوان** تستخدم أسطح ثنائيات **إضافية**. إن استخدام عدد من أسطح ثنائيات **تساوى N** يعطى عدداً من **مستويات الشدة** (أو عدد ألوان) **تساوى N2** ، والتي يمكن عرضها في نفس الوقت. الجدول 1.2 يوضح العلاقة بين عدد أسطح الثنائيات وعدد الألوان الممكنة.

الألوان	ثنائيه لكل عنصر صورة
4	2
8	3
16	4
256	8
1024	10
4096	12
≈ 16 مليون	24

جدول 1.2: أسطح الثنائيات

❖ **الحد الأدنى لعدد الأسطح:**

الحد الأدنى لعدد أسطح الثنائيات الضروري **لتكوين تخطيط ملون** هو **ثلاثة أسطح** واحد لكل من الألوان الرئيسية الأحمر، والأخضر، والأزرق.

البيانات المخزنة في هذه الأسطح **تستخدم** لتنشيط قاذفات الالكترونات الثلاثة، **و يحدد** لأي عنصر صورة على الشاشة **ثلاث ثنائيات** لتحديد **مستوى الشدة**، مما يعنى

أن عنصر الصورة يمكن أن يأخذ واحداً من **ثمانية ألوان مختلفة** كما هو موضح في الجدول 1.3.

	أحمر	أخضر	أزرق
أسود	0	0	0
أحمر	1	0	0
أخضر	0	1	0
أزرق	0	0	1
أصفر	1	1	0
Cyan	0	1	1
Magenta (القرمزي)	1	0	1
أبيض	1	1	1

جدول 1.3: الألوان والثنائيات

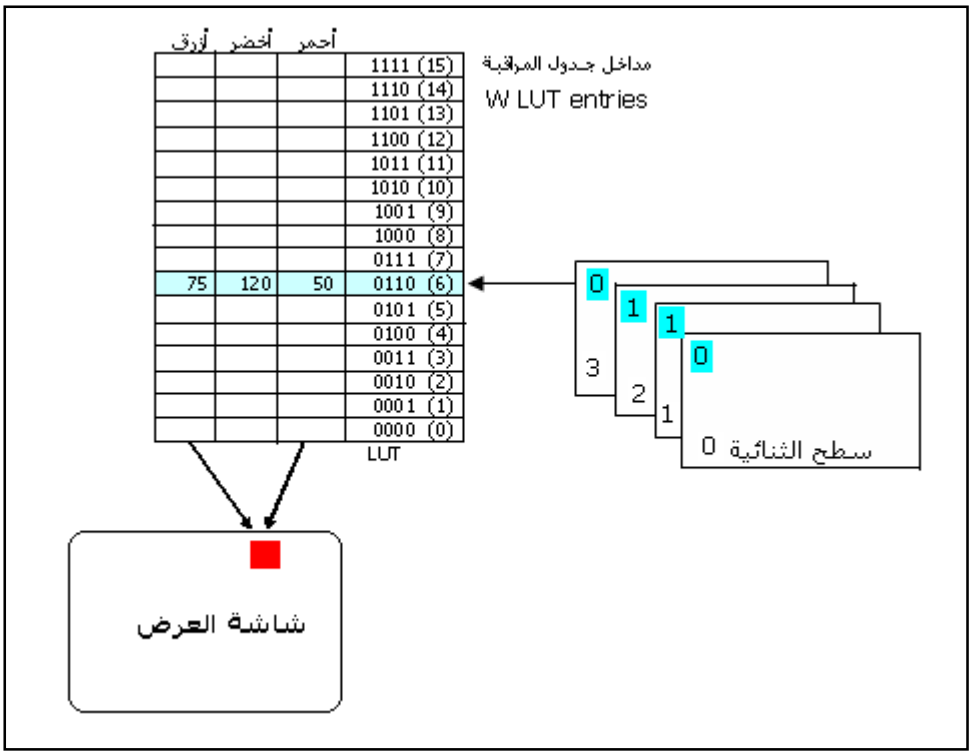
❖ **الأسطح الإضافية:**

استخدام عدد **أضافي** من أسطح الذاكرة (أسطح الثنائيات) يعطى **مجالاً أكبر لتعدد الألوان** وشدة عناصر الصورة ومن الضروري أعطاء المستخدم بعض **المرونة** في اختيار الألوان التي يريد أن يستخدمها.

❖ **اختيار الألوان:**

أحد الطرق التي تستخدم لتحيق هذه المرونة هي استخدام **جدول المراقبة lookup table** و في هذا الجدول يحتوى عنوان الذاكرة على العدد الموسع الذي يمكن وضعه على إحداثيات عنصر الصورة في العارض كما هو موضح في الشكل 1.9.

تحول أسطح الثنائيات إلى خازنة الأطر الوسيطة ويكون الرقم النتاج والمسترد من الذاكرة متوافقاً مع الموقع المبرمج في جدول المراقبة، وفى هذه الحالة **لا تستخدم** قيمة عنصر الصورة **للتحكم** مباشرة في **الشعاع الالكتروني**، بل يستخدم **كمؤشر** في جدول المراقبة فإن كان عدد أسطح الثنائيات يساوى N يجب أن يكون هناك 2N مدخل في جدول المراقبة.



الشكل 1.9: خازنة الأطر الوسيطة مع جدول المراقبة

6.2 عارضات اللوحات المستوية Flat Panel Displays

أكثر العارضات استخداما في الوقت الحالي هي عارضات أنبوب الأشعة المهبطية و هذه العارضات عادة ما تكون ضخمة وكبيرة الحجم ومحدودة المقاس، ممّا زاد الاهتمام بعارضات اللوحة المستوية.

❖ **أنواع عارضات اللوحات المستوية:**

أكثر عارضات اللوحة المستوية استخداما حاليا هي:

1) عارض البلازما Plasma . 2) عارضة الاستضاءة الكهربائية electroluminescent. 3) عارضات البلور السائل liquid crystal.

❖ **عارض البلازما Plasma:**

تتكون عارضة البلازما من صف من **بصيلات نيون** صغيرة التي يمكن تنشيطها أو عدم تنشيطها، وتبقى هذه البصيلات على حالتها حتى يتم أمرها بالتغير؛ و أي نقطة يمكن تنشيطها أو عدم تنشيطها دون الحاجة لتجديد كل الشاشة.

تتكون اللوحة من ثلاثة أسطح زجاجية مع **وجود** بصيلات النيون في **السطح الوسطى** الزجاجي و**خطوط التغذية الكهربائية** الرأسية والأفقية على **الأسطح الأخرى**.

تنشط إي من **البصيلات** النيون **بعنونة** المصفوفة و**ضبط الجهد** في خطوط التغذية المقابلة لها في السطح الأمامي والخلفي.

❖ **عيوب ومزايا عارض البلازما Plasma:**

تعانى عارضات البلازما من:

1) محدودية الاستبانة. 2) التكلفة العالية.

وبالمقابل تمتاز بـ:

1)المتانة. 2) العمل دون الحاجة لدورات تجديد.

7.2 أجهزة الدخل Input Devices

توجد كثير من أجهزة الدخل لنظم التخطيط بالحاسوب مثل:

- الفارة mouse.
- عمود الإدارة Joystick.
- صندوق الأزرار bottom box.
- القلم الضوئي light pen.

.....الخ. وسنتعرض هنا فقط للفارة.

❖ **الفأرة mouse:**

الفارة ببساطة جهاز دخل يمد الحاسوب بثلاثة ثمانيات bytes من المعلومات (كحد ادني) تمثل:

- مسافة الحركة في اتجاه x
- مسافة الحركة في اتجاه y
- حالة الزر

❖ **عرض حركة الفأرة على الشاشة:**

- يتم توفير علامة مرئية على الشاشة بواسطة برنامج التشغيل.
- ينتج حدث الفأرة عندما يتغير شي ما مثل التحرك أو الضغط على الزر.
- ترسل الفأرة إشارة إلى نظام التشغيل لإعلامه بالحدث المعنى ثم ترسل المعلومات الجديدة.
- يقوم نظام التشغيل عندها بتحديث موقع العلامة المرئية على الشاشة، كما يقوم بإعلام برامج التطبيقات بالحدث المتعلق به.
- تتسلم برامج التطبيقات هذه المعلومات بما يعرف بعملية المناداة العكسية callback procedure أو حلقة الحدث event loop.

أدناه مثال بسيط لعملية المناداة العكسية:

```
while (executing) do
{       if (menu event) ProcessMenuRequest();
        if (mouse event)
        {       GetMouseCoordinates();
                GetMouseButtons();
                PerformMouseProcess();
        }
        if (window resize event) RedrawGraphics();
}
```

❖ **استخدام الفأرة:**

تستخدم الفارة عادة لتحقيق أساليب الدخل على النحو التالي:

- **محدد الموقع Locator**

يحدد الموقع على الشاشة باستخدام علامة مرئية.

- **رباط مطاط Rubber Band**

يضبط حجم وموقع كائن التخطيط.

أسئلة تقويم ذاتي



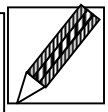
1) وضح كيفية عمل عارضات خطوط المسح الملونة
2) ما وظيفة جدول المراقبة lookup table؟
3) اذكر اثنين من أنواع عارضات اللوحة المستوية
4) عرف الفارة mouse

نشاط



تفحص عتاد نظام التخطيط في الحاسوب الخاص بك ثم أعط قائمة بكل المواصفات الفنية لأجهزة الدخل والخرج.

تدريب(2)



1. قارن بين ميزات وبين قصور كل تقنيات العرض الرئيسة
2. أشرح كيف يمكن لبعض عارضات مسح الخط عرض 256 لون في نفس اللحظة من مجموع الألوان الكلى والذي يساوى 16.7 مليون؟
3. اشرح بنية وتشغيل عارض البلور السائل Liquid Crystal Display .LCD

الخلاصة

لقد تعرفت عزيزي الدارس في ثنايا دراستك لهذه الوحدة على عمليات التخطيط بالحاسوب من نقل البيانات إلى الحاسوب وتخزينها وإرسالها بعد المعالجة إلى جهاز الخرج المناسب،و العوامل المؤثرة في التخطيط من تقنيات إنتاج المنظر وأساليب العرض وأنواع الوسائط المستخدمة في العرض وخواص أجهزة العرض، والتطور الذي حدث في التخطيط بظهور المعالجات الدقيقة وتصميم نظم العتاد الفرعية واستخدام دوال التخطيط الثلاثية الأبعاد على مستوي العتاد. كما تعرضت على تقنيات العرض المختلفة مثل أنبوب الأشعة المهبطية وعارضات خطوط المسح ومزايا وعيوب كل منها ، وأخيراً تعرفت على التطبيقات المختلفة على عناصر المشهد من إمكانية تعديلها والتأثير عليها باستخدام الألوان أو بإضافة الحركة لها أوالتحكم في مدة عرضها على الشاشة.

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، الوحدة الثانية التي تلي هذه الوحدة سنشرح فيها تبعية الجهاز، و رابط المبرمج البيني، و نظام الإحداثيات العالمي، والمعيارية و دوال الخـــرج، ودوال الـــدخل وجميعها من متطلبات برمجة نظم تخطيط الحاسوب االتى نتمنى لك التوفيق في فهمهـــا واستيعابها.

إجابات التدريبات

تدريب (1)

1.ميزات وقصور تقنيات العرض الرئيسة

نوع العارض	المقدرة التفاعلية	الإستبانة	حجم الذاكرة	التكلفة	القصور
عارض تجديد المتجه	إمكانية عرض الصور المتحركة بكفاءة عالية	عالية	متوسطة	عالية	عدم القدرة على عرض المناظر ذات التنوع اللوني الكبير
عارض خطوط المسح	إمكانية عرض بكفاءة عالية	تكاد تكون عالية	عالية	منخفضة	الحاجة لذاكرة أكبر مع زيادة القدرة اللونية
عارض اللوحات المستوية	إمكانية عرض مقبولة	منخفضة	منخفض	عالية	محدودية الإستبانة

2. يمكن عرض 256 لون في نفس اللحظة باستخدام ثمانية أسطح ثنائية لتخزين مستوى شدة الإضاءة، حيث تقوم قاذفات الإلكترونات باستخدام هذه المعلومات في إعطاء كل عنصر صورة لون محدد و يزداد عدد الألوان مع زيادة الأسطح لأن ذلك يزيد عدد الدرجات اللونية للون الأساسي التي يمكن استخدامها.

مسرد المصطلحات

عتاد الحاسوب المعياري:هو العتاد المستخدم في أجهزة الحواسيب الشخصية.

العارض أو طرفية العرض:يقصد به الشاشة.

الاستبانة: شدة وضوح المنظر على الشاشة.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة الثانية
الدخل والخرج وبرمجيات التخطيط
Output and Input and Graphics Software

محتويات الوحدة

الموضوع	الصفحة
المقدمة	31
تمهيد	31
أهداف الوحدة	31
1. تبعية الجهاز Device Dependence	32
2. رابط المبرمج البيني applications programmer interface (API)	36
3. نظام الإحداثيات العالمي World Coordinate System	38
4.المعيارية Normalization	41
5. دوال الخرج Output Functions	45
6.دوال الدخل Input Functions	46
الخلاصة	48
لمحة مسبقة عن الوحدة الدراسية التالية	48
إجابات التدريبات	49
مسرد المصطلحات	52
المراجع	53

المقدمة

تمهيد

عزيزي الدارس نرحب بك في هذه الوحدة والتي هي بعنوان الدخل والخرج وبرمجيات التخطيط **فبعد** الفهم الجيد للوحدة الأولى سيرد في هذه الوحدة شرح. تبعيـــة الجهـــاز، و رابط المبرمج البيني، نظام الإحداثيات العالمي، ومعيارية دوال الخرج، ودوال الـــدخل، وسوف تجد في ثنايا هذه الوحدة أمثلة لصقل المعرفة وأسئلة تقويم ذاتي وتدريبات لكـــي تساعدك في المراجعة والاستذكار نرجو أن توليها اهتمامك وأن تساهم معنا في نقد هـــذه الوحدة وتقويمها.

أهداف الوحدة



عزيزي الدارس ، بعد فراغك من دراسة هذه الوحدة يجب أن تكون قادراً على أن :

(1) **تصنف** أجهزة الخرج وأجهزة الدخل.

(2) **توضح** أهمية مكتبة التخطيطات.

(3) **تطبق** نظام الإحداثيات العالمي.

(4) **تستخدم** عملية المعايرة.

(5) **تصف** دوال الخرج ودوال الدخل.

1. تبعية الجهاز Device Dependence

ورد في الوحدة السابقة أن العناصر الرئيسة لعتاد نظام تخطيط الحاسوب هي:

- المعالج processor
- الذاكرة memory
- خازنة الأطر الوسيطة frames buffer
- أجهزة الخرج
  - العارض CRT, LCD
  - الطابعة
- أجهزة الدخل
  - لوحة المفاتيح، الفأرة، الخ.

تتطلب برمجة نظم تخطيط الحاسوب أن تكون الأجهزة **غير تابعة**، وبمعنى آخر، يجب أن نكتب البرامج بالبعد قدر المستطاع على صفات عتاد معين.

أكثر أجهزة خرج التخطيط استخداما في مجال الحاسوب تتدرج تحت الأنواع التالية:

- شبكة خطوط المسح *Raster* type
- ورسم بياني النقاط plot points

❖ **عناصر الصورة pixel:**

في هذه الأجهزة توضع كل **النقاط dot** (عناصر الصورة pixel) التي تكون الصورة مباشرة **في ذاكرة** وصول عشوائي random access memory. **وتتصل** وحدة المعالجة المركزية CPU بهذه الذاكرة **مباشرة أو عبر مسجلات تحكم control registers**.

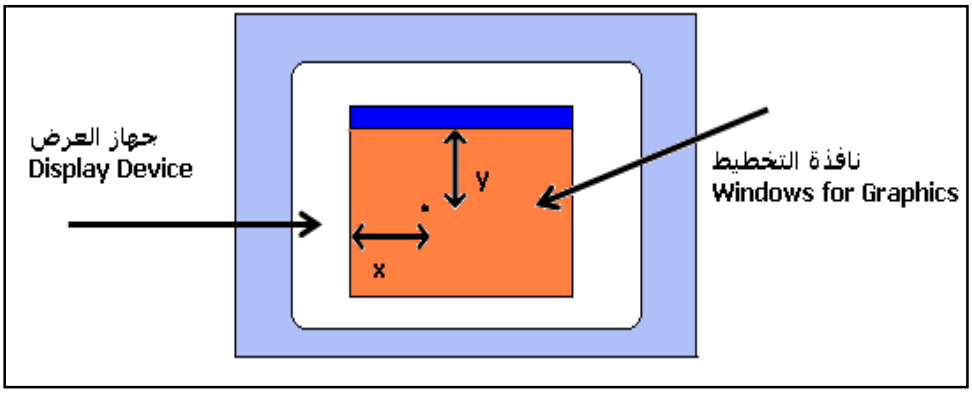
إن عدد الثنائيات bits الذاكرة المستخدمة لكل عنصر صورة **يحدد مدى الشدة intensities** التي يمكن استخدمها، فمثلا ثنائية **واحدة** تعني أن عنصر الصورة **أبيض**

أو أسود وثمانية ثنائيات تسمح: **255 شدة** (أو لون) مختلف لعنصر الصورة، و من العادة حاليا استخدام **24 أو 32 ثنائية** لتمثيل عنصر الصورة و هذا التمثيل **بملايين الألوان** (انظر الوحدة الأولى).

تسمى عدد عناصر الصورة في الشاشة **بالاستبانة** (درجة الوضوح) resolution و**تعطى** عادة **بدلالة مكونات x و y [XRes,YRes]**، وفى حالة **الطابعات الليزرية** تحدد الإستبانة **بعدد النقاط في كل بوصة.**

تغيير لون العنصر:

أكثر أجهزة عرض التخطيط استخداما هي أجهزة **شبكة خطوط المسح** حيث يمكننا استخدام **إجراء procedure** مثل **SetPixel(x,y,colour) لتغير لون** عنصر صورة معين، وهنا **الإحداثيات** المستخدمة تمثل العنوان **الحقيقي** لعنصر الصورة. (أنظر الشكل 2.1)

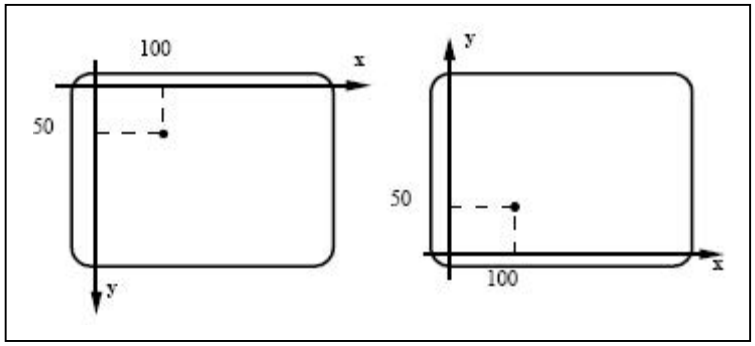


الشكل 1.2: معنى الإجراء SetPixel(x,y,colour)

❖ **الإحداثيات الفعلية:**

تتمثل مشكلة استخدام الإحداثيات الفعلية لعنصر الصورة في أن **أجهزة العرض** المختلفة قد **تستخدم أساليب مختلفة** لتحديد عنوان عنصر الصورة على الشاشة. ففي بعض هذه الأجهزة تكون **نقطة تقاطع محاور الإحداثيات** origin في **الركن الشمالي**

الأعلى وفى البعض الأخر تكون هذه النقطة **في الركن الأسفل على يمين الشاشة** كمـــا هو موضح في الشكل 2.2.



الشكل 2.2 تغير أسلوب عنونة عنصر الصورة من نظام إلى أخر

أسئلة تقويم ذاتي

1. ما هي أكثر أجهزة خرج التخطيط استخداما في مجال الحاسوب؟
2. ما ناتج استخدم الإجراء التالي (SetPixel(x,y,colour؟
3. متي تكون الإحداثيات المستخدمة تمثل العنوان الحقيقي لعنصر الصورة؟

تدريب (1)

1. "علل" (يجب أن نكتب البرامج بالبعد قدر المستطاع من صفات عتاد معين).
2. كيف يمكن تحديد الاستبانة (درجة الوضوح) بالنسبة لـــ :
   أ/ الشاشة. ب/ الطابعات الليزرية.
3. فيم تتمثل مشكلة استخدام الإحداثيات الفعلية لعنصر الصورة؟

❖ **نظم التشغيل وإمكانية رسم التخطيط:**

يوفر لنا نظام **التشغيل** إمكانيات **رسم التخطيط** على مستوى **عنصر الصورة،** فمثلا في **رابط المبرمج البيني** Windows 32 يمكن **استخدام الإجراءات** التالية لتخطيط:

MoveToEx(hdc xpix, ypix);
LineTo(hdc, xpix, ypix);
TextOut(hdc, xpix, ypix, message, length);

حيث تحدد hdc نوع النافذة وكل من xpix و ypix نظام الإحداثيات.

ونوضح هنا أن رابط المبرمج البيني **لا يوفر لنا عدم تبعية مطلقة للجهاز،** فمثلا إن كتبنا برنامج يحتوى على كل القيم في عنصر الصورة، فإنه لا يحدث تغير إن غيرنا حجم النافذة.

برامج التخطيط وجهاز العرض:

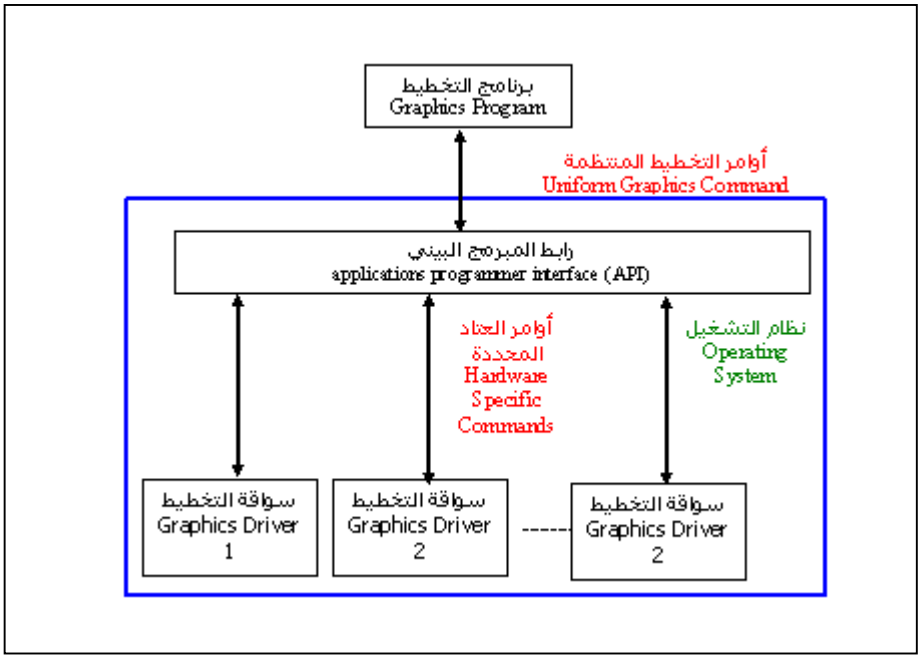
إن الحاجة لكتابة برامج تخطيط غير معتمدة على جهاز العرض تتمثل فيما يلي:

في التطبيقات العادية هناك حاجة لضبط حجم الصورة حتى إن تغير حجم النافذة

- يجب أن تكون الصورة غير معتمدة على الاستبانة
- السماح بنقل الصورة بين نظم مختلفة (حواسيب شخصية ومحطات عمل وحواسيب فائقة القدرة الخ)

2. رابط المبرمج البيني applications programmer interface (API)

في أغلب ألأحيان لا نهتم كثراً **بتفاصيل** الجهاز، حيث **يقوم نظام التشغيل** operating system بالاهتمام بذلك **بواسطة** ما نسميه **رابط المبرمج البيني**.applications Programmer interface or API أو **مكتبة التخطيطات** graphics library. والذي يوفر للمبرمج مجموعة من الإجراءات المنتظمة لرسم المشهد بغض النظر عن الجهاز المستخدم (أنظر الشكل 2.3)



الشكل 2.3: رابط المبرمج البيني

- **أكثر روابط المبرمج البينية (مكتبات التخطيط) مستخدمة حاليا وأكثرها شيوعا:**
  - OpenGL
  - Direct3D
  - Java3D
  - VRML

- Win32 API ♦
- GKS ♦
- PHIGS ♦

سنتعرض في هذا المقرر ببعض التفصيل لاستخدام **مكتبة التخطـيط OpenGL**. أما في بقية هذه الوحدة فسنتعرف بعض الأمثلـة مـن **مكتبـة التخطـيط GKS و PHIGS**بالإضافة إلى OpenGL.

خواص مكتبة التخطيط OpenGL

إن مكتبة التخطيط OpenGL لديها الخواص التالية:

- ✓ **لا تعتمد على** العتاد حيث يمكن استخدامه على الحواسـيب الشخصـية PC ومحطـات العمـل workstations والحواسـيب الفائقـة القـدرة Supercomputer.
- ✓ **تعتمد على** نظام التشغيل وتدعم نظم التشـغيل التاليـة: Windows NT, Windows 2000, Windows XP Linux and Unix.
- ✓ **يمكن أن تنفذ** الإجراءات باستخدام البرامج مثال المعالج أو العتاد مثل لوحة تسريع التخطيط accelerated graphics card إن وجدت.
- ✓ **يمكن استخدامها مع لغات البرمجة التاليـة:** C, C++, Ada, Fortran, Java.
- ✓ **تدعم** أداء المضلع polygon rendering و التخطـيط البنيـوي texture mapping وعدم استخدام الاسم المستعار anti-aliasing.
- ✓ **لا تـدعم** متابعـة الإشـعاع ray tracing و أداء الحجـم volume rendering.

❖ **البرنامج المتنقل portable:**

لكتابة برنامج متنقل portable لايعتمد على جهاز التخطيط، علينا **إزالة التبعية** للجهاز الموضحة أعلاه من أغلب برامج التخطيط. **ففي** بعض التطبيقات مثـل ألعـاب

الحاسوب computer games، **نحتاج لاستغلال** كل مساحة الشاشة (العـــارض)، وفـــى مثل هذه الحالة **نحتاج أن يتواءم** المشهد مع التغيرات في الاستبانة.

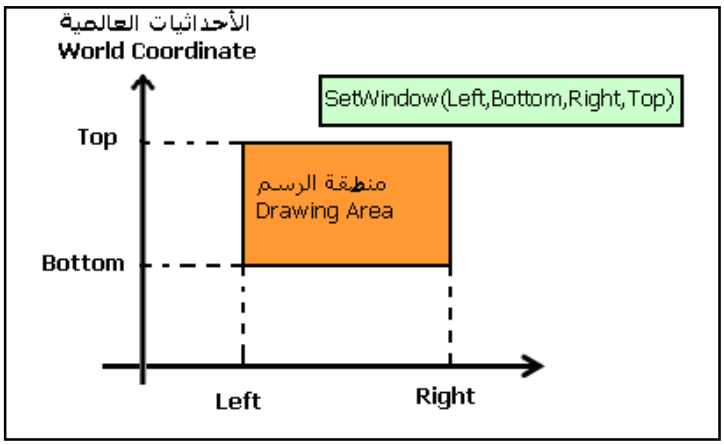
ومن كل ما ذكر أعلاه نخلص إلى أنه لا يمكننا كتابة برنامج بكفاءة عالية باستخدام عنوان عنصر الصورة.

3. نظام الإحداثيات العالمي World Coordinate System:

تتطلب تطبيقات التخطيط الفعلية أن يكون كل مشهد يرسم في النافذة غير معتمد على موقع النافذة على الشاشة وحجمها. إن **استخدام نظام الإحداثيات العالمي** يحقق عدم الاعتمادية هذه؛ حيث **تحدد مساحة المشهد** باستخدام وحدات قيـــاس مختلفـــة ومناســـبة للتطبيق المعنى مثل المتر والسنة الضوئية والميكرون الخ، وفى العادة **يســـتخدم الآمـــر** التالي لتحديد الإحداثيات العالمية

SetWindow(left,bottom,right,top)

يمكن اعتبار هذا نافذة في العالم تتوافق مع النافذة في الشاشة (أنظر الشكل 2.4)



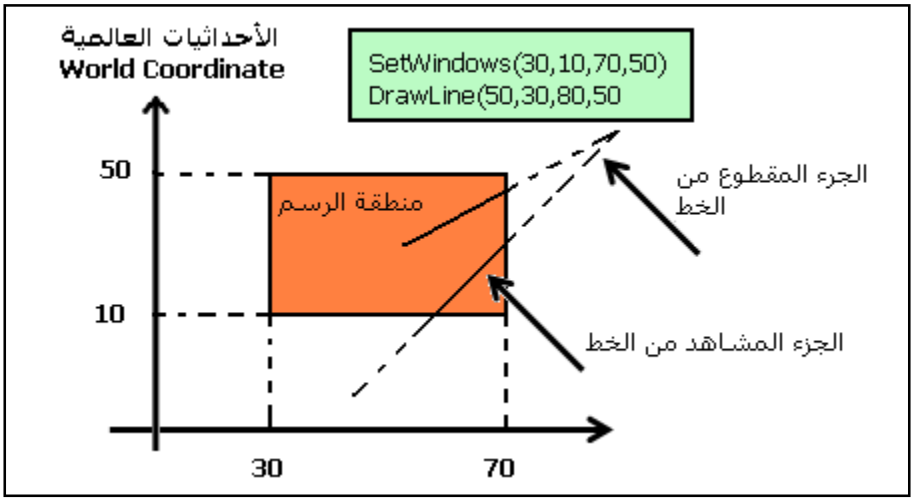
الشكل 2.4: النظام العالمي للإحداثيات

بعد تحديدنا لنظام الإحداثيات العالمي يمكننا تحقيق **بدائيات primitives** الرسم التي تستخدم معه مثل

```
DrawLine(x1,y1,x2,y2);
DrawCircle(x1,y1,r);
DrawPolygon(PointArray);
DrawText(x1,y1,"A Message");
```

في العادة أي جزء من التخطيط **يقع** خارج النافذة يتم **قصه**، وهـذا قـد **يمثـل مشكلة** في بعض الأحيان؛ فمثلا في الشكل 2.5 يعطـى التخطـيط المتحصـل عليـه باستخدام التعليمات التالية:

```
SetWindow(30,10,70,50);
DrawLine(50,30,80,50);
DrawLine(80,50,50,5);
```



الشكل 2.5: قص جزء من الرسم خارج النافذة

❖ **الرسم غير المعتمد على جهاز العرض:**

لا تستخدم بدائيات الرسم غير المعتمدة على جهاز العرض عوامل شاملة، بـــل **تستخدم** عدداً من **الصفات attributes** لتحديد الكائن (المشهد). فمـــثلا **الخـــط يحـــدد بالصفات** التالية:

- ❖ ألأسلوب(مستمر أو متقطع)
- ❖ السمك (عدد النقاط)
- ❖ اللون،

أما النص text فيحدد باستخدام

- ❖ الحروف
- ❖ الحجم
- ❖ اللون.

أسئلة تقويم ذاتي

(1) يتيح نظام التشغيل إمكانيات رسم التخطيط على مستوى عنصر الصورة اكتب الإجراءات اللازمة لهذا النوع من التخطيط

(2) اذكر اثنين من خواص مكتبة التخطيط OpenGL

(3) إلى ماذا نحتاج في استغلال كل مساحة الشاشة في تطبيقات ألعاب الحاسوب؟

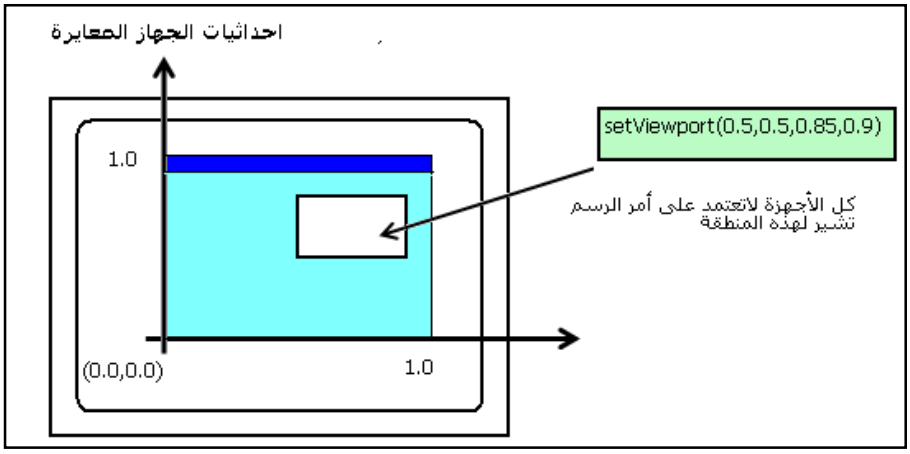
(4) تستخدم الصفات لتحديد الكائن (المشهد)، إلا أن هنالك صفة يشترك فيها الخط والنص هي

أ/ الاستمرارية. ب/ اللون. ج/ التدرج.

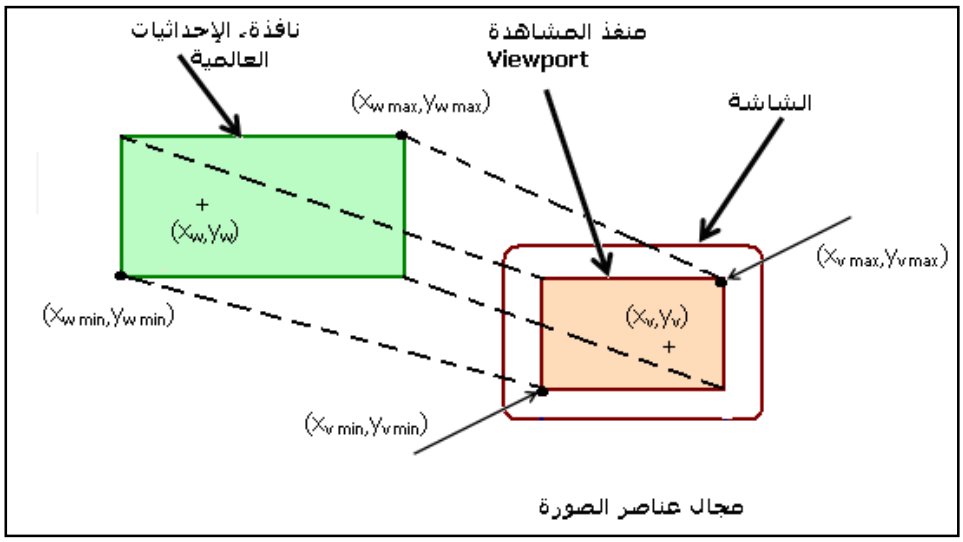
(5) 5. رابط المبرمج البيني لا يوفر عدم تبعية مطلقة للجهاز ،برهن على ذلك بمثال

4. المعيارية Normalization

حتى نتمكن من **مشاهدة المخطط** على الشاشة لابد من **ربط** بـــدائيات التخطـــيط غير المعتمدة على الجهاز **مع** أوامر الرسم المعتمدة على الجهـــاز **ويـــتم** هـــذا **الـــربط** باستخدام **عملية المعايرة.** (انظر الشكل 2.6)



الشكل 2.6: المعايرة



الشكل 2.7: العلاقة بين إحداثيات النافذة ومنفذ المشاهدة

نحتاج في المعايرة إلى **ترجمة** الإحداثيات العالمية إلـــى مجموعـــة إحـــداثيات مناسبة للرسم **باستخدام** رابط المبرمج البيني، فلابد في البداية من مناداة رابط المبـــرمج البيني من نظام التشغيل حتى نتمكن من تحديد عناوين عناصر الصورة لأركان المساحة التي سنستخدمها، باعتبار الشكل 2.7 **يتم الربط** بين **نظام** الإحداثيات العـــالمي **ونظـــام** إحداثيات جهاز العرض **باستخدام العلاقات** التالية :

$$\frac{x_v - x_{v\min}}{x_{v\max} - x_{v\min}} = \frac{x_w - x_{w\min}}{x_{w\max} - x_{w\min}} \qquad 3.1a$$

$$\frac{y_v - y_{v\min}}{y_{v\max} - y_{v\min}} = \frac{y_w - y_{w\min}}{y_{w\max} - y_{w\min}} \qquad 3.1b$$

وبإعادة ترتيب هذه المعادلة تتحصل على

$$x_v = (x_w - x_{w\min})\left(\frac{x_{v\max} - x_{v\min}}{x_{w\max} - x_{w\min}}\right) + x_{v\min} \qquad 3.2a$$

$$y_v = (y_w - y_{w\min})\left(\frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}}\right) + y_{v\min} \qquad 3.2b$$

الحدود

$S_x = \left(\frac{x_{v\max} - x_{v\min}}{x_{w\max} - x_{w\min}}\right)$ و $S_y = \left(\frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}}\right)$ **ثابتـــة constant** لكـــل النقـــاط وتمثـــل ببساطة عامل **ضبط** مقاس (الحجم) في اتجاه x و y. **إذا كان** $S_x \neq S_y$ لا يحدث **تشـــويش** distortion في الصورة. **مفهوم نسبة الواجهة** aspect ratio **يشير** إلى هذه الحالـــة، **حيث** تعطى نسبة الواجهة AR **بقسمة العرض على الارتفاع**

$$AR = \frac{x_{mav} - x_{\min}}{y_{\max} - y_{\min}} \qquad 3.3$$

إذا كانت نسبة الواجهة لكل من النافذة ومنفذ المشاهدة **متســـاوية** عليـــه يكـــون $S_x = S_y$ و يحدث **تشويش** للصورة.

ويمكن دمج المعادلات أعلاه في المعادلات الخطية البسيطة التالية:

$$x_v := x_w * A + B;$$
$$y_v := y_w * C + D; \qquad 3.4$$

❖ **منافذ المشهد viewports:**

تمثل منافذ المشهد viewports **أصغر جزء** من النافذة يمكن عرضه، وعند اختيارنا لمنفذ المشهد، **فإن** الأسلوب المتعارف علية **لنقل** (تخطيط) الإحداثيات العالمية **يكون إلى المنفذ** وليس لكل مساحة الرسم. **تعرف** منافذ المشهد في إحداثيات الجهاز المعيارية **ويكون** لكل نافذة رسم إحداثيات أركان [0.0,0.0] و [1.0,1.0]

طريقة المعايرة عند استخدام منافذ المشهد viewports:

إن استخدام منافذ المشهد يغير من طريقتنا للمعايرة، حيث نحتاج في هذه الحالة لما يلي:

- **منادية** رابط المبرمج البيني من نظام التشغيل لتحديد عناوين عناصر الصورة في أركان النافذة.
- **استخدام** محيط منفذ المشهد لحساب عناوين عناصر الصورة للمساحة التي سيظهر عليها الرسم.
- **حساب** عوامل المعايرة A و B و C و D.

مثال

أوجد مصفوفة التحويل التي تنقل النقاط في النافذة التي لديها الركن الأسفل الأيسر عند (2,2) والركن الأعلى ألأيمن عند (6,5) إلى منفذ مشاهدة معاير لديه الركن الأسفل الأيسر عند (0.5,0.5) والركن الأعلى ألأيمن عند (1,1).

الحل:

عوامل النافذة ومنفذ المشاهدة:

$x_{wmin} = 2$ $\quad$ $x_{vmin} = 0.5$

$x_{wmax} = 6$ $\quad$ $x_{vmax} = 1$

$y_{wmin} = 2$ $\quad$ $x_{vmin} = 0.5$

$y_{wmax} = 5$ $\quad$ $y_{vmax} = 1$

باستخدام المعادلة 3.2 تتحصل على

$$S_x = \left(\frac{1-0.5}{6-2}\right) = \frac{1}{8}$$

$$S_y = \left(\frac{1-0.5}{5-2}\right) = \frac{1}{6}$$

عليه تكتب المعادلة 3.2 على الهيئة التالية:

$$x_v = (x_w - x_{w\min}) + x_{v\min}$$

$$y_v = (y_w - y_{w\min}) + y_{v\min}$$

في هيئة المصفوفة

$$[x_v \quad y_v \quad 1] = [x_w \quad y_w \quad 1]\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ (-S_x.x_{w\min} + x_{v\min}) & (-S_y.y_{w\min} + y_{v\min}) & 1 \end{bmatrix}$$

بالتعويض تتحصل على مصفوفة النقل التالية:

$$\begin{bmatrix} \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ \frac{1}{4} & \frac{1}{6} & 1 \end{bmatrix}$$

5. دوال الخرج Output Functions

كل معايير التخطيط تعطى **متواليات من بدائيات الخرج** output primitives، يمكن باستخدامها **إنتاج مشاهد معقدة**، و **هذه البدائيات** قد تكون **خطاً** line أو **راسمة** marker **ثنائي** أو **ثلاثي** الأبعاد. الجدول 2.1 يوضح بعض البدائيات في مكتبة PHIGS.

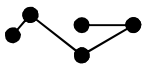
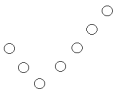
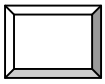
تدعم مكتبة التخطيط مبرمج التطبيقات بدوال خرج ابتدائية مثل:

DRAW_POLYINE(SetOfPoints)

ترسم هذه الابتدائية مقطع خط يصل مجموعة النقاط المحددة.

توفر الابتدائية معلومات عن **الموقع والهندسة، لتحديد** عوامل أخرى مثل اللون ونوع الخط الخ، **عرفت** صفات رسم drawing attributes متصلة بكل ابتدائية **عبر دالة** SET <attribute>. الجدول 2.2 يوضح صفات الابتدائية polyline.

الجدول 2.1. ابتدائيات خرج PHIGS

الابتدائية	الوصف	مثال
polyline	متسلسلة نقاط تحدد مجموعة من الخطوط المتصلة	
Polymarker	رموز تضع فى مواقع محددة	
Text	رموز متصلة تضع في أماكن محددة	**تخطيط - graphics**
Fill area	تعبئة المساحة الداخلية للمضلع ولا تشمل الحافات بأنماط مختلفة.	
Fill area set	تعبئة مساحات مضلعات معقدة وقد تشمل الحافات بأنماط مختلفة	

الجدول 2.2. صفات الابتدائية polyline

الصفة atributte	الوصف
Linetype	للاختيار من: صلب: —————————— مقطع: - - -

أسئلة تقويم ذاتي



(1) عرف المعايرة؟

(2) كيف تتم ترجمة الإحداثيات العالمية إلى مجموعة إحداثيات مناسبة للرسم؟

(3) وضح مفهوم نسبة الواجهة aspect ratio

(4) عند قياس نسبة الواجهة لكل من النافذة ومنفذ المشاهدة متى يحدث تشوش للصورة ومتى لا يحدث؟

تدريب (3)



(1) يعمل مستخدم في نظام إحداثياته العالمي الخاص والذي يمثل مساحة مربعة لديها أحديثات الأركان التالية [(80,50) , (50,-20)]. المشهد يتكون من مثلث لديه إحداثيات الرؤوس vertices التالية:{5,30-] و [50,30] و [10,-5-]. نقلت مساحة المستخدم إلى منفذ مشهد مربع يمثل جزء من نافذة الشاشة وتستخدم إحداثيات الجهاز معرفة بوحدات عناصر الصورة. إحداثيات أركان الجهاز هي: [(613,818) ,(409,613)]. كل الشاشة لديها عنوانين عناصر صورة من 0 إلى 1023 لكل من x و y.

أ. خطط بالتقريب ما سيظهر على الشاشة (تجاهل كل شي قد يوفره نظام التشغيل)

ب. ما هي عناصر الصورة المقابلة لنقطة تقاطع الإحداثيات [0,0] في نظام إحداثيات المستخدم؟

ج. النقل (التخطيط) بين نظام إحداثيات المستخدم ونظام إحداثيات الجهاز يعطي بالتالي

$x' = Ax + B$

$y' = Cy + D$

أوجد قيم A و B و C و D.

د. إذا حرك المستخدم النافذة بمقدار 50 عنصر صورة في اتجاه y الموجب و بمقدار 100 عنصر صورة في اتجاه x الموجب، أوجد القيم الجديدة A و B و C و D. وأعطى مخططاً لم سيظهر في النافذة على الشاشة.

ه. إن أرجع المستخدم النافذة لموقعها السابق ثم قام بعملية تكبير "zoom in" بتغير نظام الإحداثيات العالمي، بحيث كبرت الصورة أربع مرات مقارنة بحجمها الأول (إي مرتين في كل اتجاه) مع بقاء نقطة تقاطع محاور الإحداثيات في مكانها. أعد حساب إحداثيات النافذة والثوابت أعلاه A و B و C و D.

و. افترض إن كل ما يتخطى نافذة المستخدم يتم قصه من المنظر، أعط مخططاً لما سوف يظهر على الشاشة.

الخلاصة

بعد دراستك لهذه الوحدة ينبغي أن تكون قد حصَّلت المعارف التالية:

أن مدى الشدة(أي الألوان) التي يمكن استخدامها لكل عنصر صورة يحدد بواسطة عدد الثنائيات، فمثلا ثنائية واحدة تعنى أن عنصر الصورة أبيض أو أسود، وثمانية ثنائيات تسمح بـــ255 لوناً، و 24 أو 32 ثنائية تسمح بملايين الألوان.

أن رابط المبرمج البيني يوفر الإجراءات اللازمة لرسم المشهد بغض النظر عن الجهاز المستخدم، كما توفر مكتبات التخطيط خواص عديدة يمكن الإفادة منها في رسم التخطيط، وأن نظام الإحداثيات العالمي يحقق عدم الاعتمادية على أجهزة العرض فلا يؤثر موقع النافذة ولا حجمها في عملية التخطيط.

كما تستخدم عملية المعايرة للربط بين بدائيات التخطيط غير المعتمدة على الجهاز مع أوامر الرسم المعتمدة على الجهاز وذلك بترجمة الإحداثيات العالمية إلى مجموعة إحداثيات مناسبة للرسم باستخدام رابط المبرمج البيني.

تعطي معايير التخطيط متواليات من بدائيات الخرج تستخدم في إنتاج مشاهد معقدة، بينما تصنف دوال الدخل إلى مجموعات حسب قيم البيانات التي يرجعها الجهاز لبرنامج التطبيقات.

لمحة مسبقة عن الوحدة التالية:

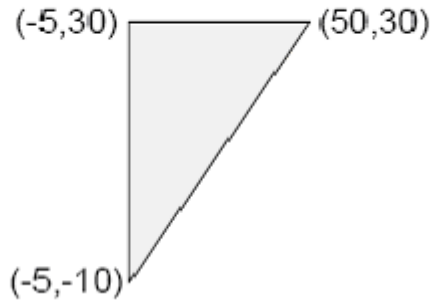
في **الوحدة التالية** نشرح كيفية تمثيل الإشكال الثنائية الأبعاد، وتمثيل الخطـــوط والمنحنيات، والرسم الأمثل للخط ، وتمثيـــل المضـــلعات، و تحويـــل المســـح، نظـــام الإحداثيات المتجانس، و التحويلات الثنائية الأبعاد والتي لهـــا أهميـــة فـــي تخطيطـــات الحاسوب.

إجابات التدريبات

تدريب(1)

تدريب(2)
(أ)

ما أن نظام التشغيل يوفر رهبط البرامج البيني API ، اذاً فالمطلوب هو الشكل العام التالي:



(ب)

أولاً يتم حساب مصفوفة النقل حسب القانون التالي:

$$[x_v \quad y_v \quad 1] = [x_w \quad y_w \quad 1]\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ (-S_x.x_{wmin}+x_{vmin}) & (-S_y.y_{wmin}+y_{vmin}) & 1 \end{bmatrix}$$

حيث:

$$S_x = \left(\frac{x_{vmax}-x_{vmin}}{x_{wmax}-x_{wmin}}\right) \qquad S_y = \left(\frac{y_{vmax}-y_{vmin}}{y_{wmax}-y_{wmin}}\right)$$

و عليه تصبح المصفوفة هى:

$$[x_v \quad y_v \quad 1] = [x_w \quad y_w \quad 1]\begin{bmatrix} 2.04 & 0 & 0 \\ 0 & 2.05 & 0 \\ 449.8 & 715.5 & 1 \end{bmatrix}$$

و عليه فإن الإحداثيات المقابلة لنقطة التقاطع(0,0) هي ≈ (450,716) . و لايجاد رقم عنصر الصورة يمكن استخدام القانون التالي:

رقم عنصر الصورة =

[(الإحداثي السيني للنقطة – 1)*العدد الكلي للأعمدة +(الإحداثي الصادي للنقطة-1)]

اذا" رقم عنصر الصورة للنقطة (0,0) = 460491

(ج)

من مصفوفة النقل في المسألة السابقة نجد ان:

$X' = 2.04x + 449.8$

$y' = 2.05y + 715.5$

و عليه:

$A = 2.04 \quad B = 449.8 \quad C = 2.05 \quad D = 715.5$

(د)

في النافذة على الشاشة.

نتيجة للتحريك ستصبح إحداثيات نافذة المستخدم هي [(509,663),(713,868)]

و عليه تصبح مصفوفة النقل الجديدة هي:

$$[x \quad y \quad 1] = [x \quad y \quad 1] \begin{bmatrix} 2.04 & 0 & 0 \\ 0 & 2.05 & 0 \\ 549.8 & 765.5 & 1 \end{bmatrix}$$

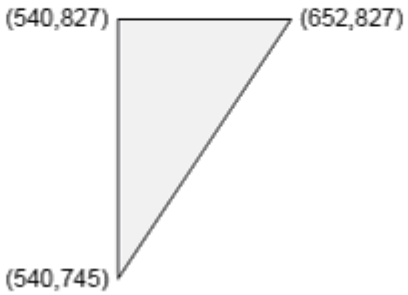
و منها:

$X' = 2.04x + 549.8$

$y' = 2.05y + 765.5$

$A = 2.04 \quad B = 549.8 \quad C = 2.05 \quad D = 765.5$

و بتحويل الإحداثيات العالمية إلى نافذة المستخدم سيظهر الشكل التالي:



(٥)

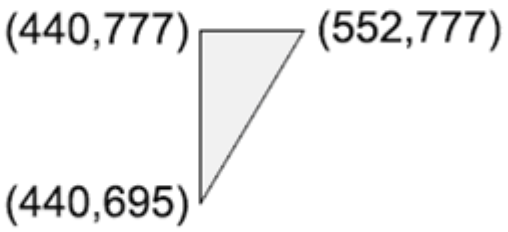
ستصبح إحداثيات نافذة المستخدم هي [(160,100),(-40,-100)]
و باستخدام مصفوفة النقل سنجد ان:

$$X' = 1.02x + 449.8$$
$$Y' = 1.025y + 715.5$$

A = 1.02 B = 449.8 C = 1.025 D = 715.5

و)

لن يحدث تخطي للشكل خارج نافذة المستخدم و بالتالي لن يحدث قص، و سيكون الشكل كالتالي:



مسرد المصطلحات

عناصر الصورة pixel:مجموعة النقاط التي تتكون منها الصورة

رابط المبرمج البيني(API):جزء من برنامج نظام التشغيل يوفر للمبرمج مجموعة من الإجراءات المنتظمة لرسم المشهد بغض النظر عن الجهاز المستخدم.

نظام الإحداثيات العالمي World Coordinate System: هو النظام الذي يحقق عدم اعتمادية رسم التخطيط على موقع وحجم النافذة باستخدام مقاييس مناسبة كالمتر والسنة الضوئية والميكرون.

المعيارية Normalization:عبارة عن ربط بين بدائيات التخطيط غير المعتمدة على الجهاز مع أوامر الرسم المعتمدة على الجهاز.

مفهوم نسبة الواجهة(AR) aspect rati :عبارة عن علاقة تحسب بقسمة العرض على الارتفاع.

منافذ المشهد viewports: أصغر جزء من النافذة يمكن عرضه.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة الثالثة
المناظر الثنائية الأبعاد
Two Dimensional Views

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**57**
تمهيد	**57**
أهداف الوحدة	**57**
1.تمثيل الكائن ثنائي الأبعاد:	**58**
1.1 تمثيل الأشكال الثنائية الأبعاد Representation of Two Dimensional Geometry	58
2.1 تمثيل الخطوط والمنحنيات Lines and Curves Representation	60
3.1 الرسم الأمثل للخط Optimum Line Drawing	61
4.1 تمثيل المضلعات Polygons	67
5.1 تحويل المسح Scan Conversion	68
6.1 نظام الإحداثيات المتجانس homogenous coordinate system	72
7.1 التحويلات الثنائية الأبعاد Two Dimensional Transformation	73
الخلاصة	**85**
لمحة مسبقة عن الوحدة الدراسية التالية	**85**
إجابات التدريبات	**86**
مسرد المصطلحات	**89**
المراجع	**90**

المقدمة

تمهيد

عزيزي الدارس! مرحباً بك في الوحدة الثالثة والتي هي بعنوان المناظر الثنائية الأبعاد وسوف نتطرق فيها إلى كيفية تحديد شكل الكائن ثنائي الأبعاد وحجمه وتمثيل الخطوط والمنحنيات، وسوف نشرح الرسم الأمثل للخط، وتمثيل المضلعات، وتحويل المسح، ونظام الإحداثيات المتجانس، وأخيراً سنتناول التحويلات ثنائية الأبعاد.

أهداف الوحدة

عزيزي الدارس، بعد دراستك هذه الوحدة ينبغي أن تكون قادراً على أن:

- **تمثل** الأشكال الثنائية والخطوط والمنحنيات.
- **تشرح** الرسم الأمثل للخط.
- **تصف** المضلعات.
- **تُعرّف** عملية تحويل المسح والتحويلات الثنائية الأبعاد .

1. تمثيل الكائن ثنائي الأبعاد

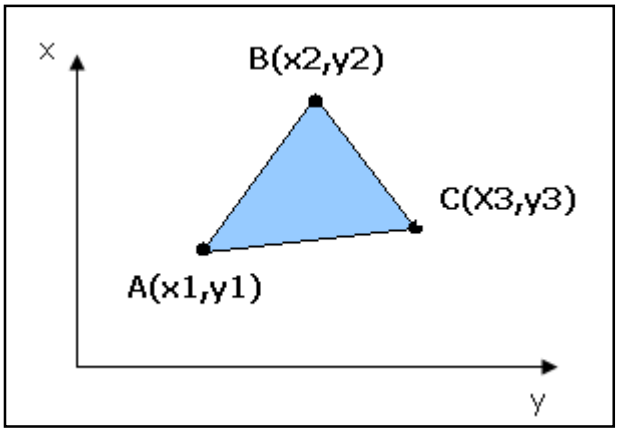
1.1 تمثيل الأشكال الثنائية الأبعاد

Representation of Two Dimensional Geometry

في التخطيط بالحاسوب يحدد شكل وحجم الكائن الثنائي الأبعاد **باستخدام واصفات عددية** ثنائية الأبعاد مرتبطة بنظام إحداثيات معين، عادة ما يكون **نظام إحداثيات ديكارت** Cartesian coordinate، يمكن **استخدام** عدد من التحويلات الهندسية **لإزاحة** وإعادة حجم ودوران الشكل أو الكائن.

❖ **إحداثيات النقطة:**

العنصر الأساسي في أي نموذج ثنائي الأبعاد هي **النقطة**. فالخط مثلا يمثل بنقطتي النهاية والسطح يمثل بعدد من النقاط، و**أي تمثيل ثنائي الأبعاد** يحدد بمجموعة من إحداثيات x و y، التي تعتبر عناصر النموذج. الشكل 3.1 يوضح تمثيل مثلث triangle باستخدام إحداثيات رؤوسه vertices.



الشكل 3.1: تمثيل المثلث باستخدام نظام إحداثيات xy

❖ **التمثيل باستخدام المصفوفة:**

يمكن تمثيل المثلث في (الشكل 3.1) أيضا استخدام مصفوفة ترتيبها 3×2 كالآتي:

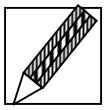
$$[P]_{TRIANGLE} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \qquad 3.1$$

حيث إن أي زوج من xy يمثل **متجه موقع** position vector نسبة إلى نظام إحداثيات معين.

❖ **فائدة استخدام المصفوفات:**

- ✓ إن استخدام المصفوفات مفيد في تحديد الشكل والتعامل معه في التخطيط بالحاسوب.
- ✓ إن بعض التحويلات يتم تحقيقها باستخدام عمليات المصفوفات وبعض التحويلات الأخرى تتم بجمع المتجهات.
- ✓ يتطلب ذلك توحيد نظام الإحداثيات باستخدام **نظام الإحداثيات المتجانس** homogenous coordinate الذي نوقش في الوحدة السابقة.

تدريب (1)



معطى مثلث لديه إحداثيات الرؤوس التالية: A(3,1), B(1,3), C(3,3). لف (دور) المثلث بزاوية مقدارها 90^o على المحور الذي يمر عبر النقطة P(2,2) ثم حدد الإحداثيات الجديدة لكل من A و B و C.

2.1 تمثيل الخطوط والمنحنيات

Lines and Curves Representation

أن الخطوة الابتدائية لرسم الخطوط والمنحنيات هي **اختيار تمثيل مناسب لهـــا.**

توجد ثلاثة طرق لتمثيل الخطوط والمنحنيات يمكن استخدامها وهي:

أ. التمثيل باستخدام **المعادلات الصريح** Explicit: y=f(x)

وهنا يمثل الخط بالمعادلة

$$y = \frac{dy}{dx}(x - x_0) + y_0 \qquad 3.2$$

وتمثل الدائرة بالمعادلة

$$y = \pm\sqrt{r^2 - x^2} \qquad |x| \le r \qquad 3.3$$

ب. التمثيل باستخدام **المعادلات الوسيطة** (البارامترية) parametric

x=(ft), y=f(t) : equation

وهنا يمثل الخط بالمعادلة

$$x(t) = x_0 + t(x_1 - x_0)$$
$$y(t) = y_0 + t(y_1 - y_0)$$
$$t \in [0,1] \qquad 3.4$$

$$P(t) = P_0 + t(P_1 - P_0), \text{ or}$$
$$P(t) = (1-t))P_0 + tP_1$$

وتمثل الدائرة بالمعادلات التالية

$$x(\theta) = r\cos(\theta)$$
$$y(\theta) = r\sin(\theta) \qquad 3.3$$
$$\theta \in [0,2\pi]$$

ج. التمثيل **بالمعادلات الضمنية** Implicit: f(x,y)=0

حيث يمثل الخط بالمعادلات التالية

$$F(x,y) = (x - x_0)dy - (y - y_0)dx \qquad 3.4$$

أذا كان F(x,y) =0 تكون (x,y) على الخط

F(x,y) >0 تكون (x,y) أعلى الخط

F(x,y) <0 تكون (x,y) أسفل الخط

ونمثل الدائرة باستخدام

$$x^2 + y^2 = r^2$$

$$F(x,y) = x^2 + y^2 - r^2 \qquad 3.5$$

أذا كان F(x,y) =0 تكون (x,y) على االدائرة

F(x,y) 0 تكون (x,y) خارج الدائرة

F(x,y)<0 تكون (x,y) داخل الدائرة

3.1 الرسم الأمثل للخط Optimum Line Drawing

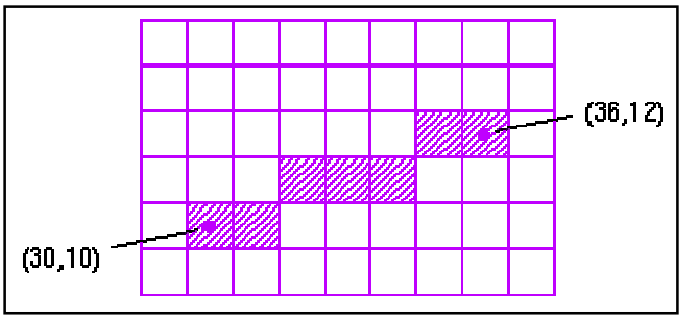
رسم الخط على **شبكة مسح الخطوط** raster grid يتطلب بعض التقديرات والمعالجة، **وتسمى** هذه العملية **بمسح الخطوط** rasterization أو **تحويل المسح** scan-conversion.

❖ **خصائص الرسم الأمثل للخط**

إن الخط المرسوم من عنصر الصورة (x1,y1) إلى عنصر الصورة (x2,y2) لابد إن تكون له الخصائص التالية (انظر الشكل 3.1):

- مستقيم straight
- يمر عبر النقاط النهائية
- سلس smooth

- لا يعتمد على ترتيب النقاط النهائية
- لديه سطوع منتظم
- سطوع غير معتمد على الانحدار



الشكل 3.2: تخطيط الخط

❖ **خوارزميات رسم الخط المستقيم:**

في المثال أدناه استخدمت

خوارزمية مباشرة لرسم الخط المستقيم

```
DrawLine(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
  float y;
  int x;

  for (x=x1; x<=x2; x++) {
    y = y1 +  (x-x1)*(y2-y1)/(x2-x1)
    SetPixel(x, Round(y) );
  }
}
```

يوضح المثال أدناه استخدام

خوارزمية أفضل لرسم الخط المستقيم

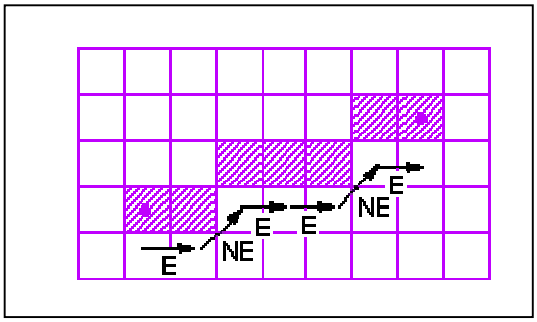
```
  DrawLine(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
```

```
 float m,y;
 int dx,dy,x;
dx = x2 – x1;
dy = y2 – y1;
 m = dy/dx;
y = y1 + 0.5;
for (x=x1; x<=x2; x++) {
 SetPixel(x, Floor(y) );
y = y + m;
 }
}
```

❖ **خوارزمية النقطة الوسيطة midpoint algorithm:**

يمكن استخدام **خوارزمية النقطة الوسيطة** midpoint algorithm لرسم الخط المستقيم.

إن خوارزمية النقطة الوسيطة أفضل من الخوارزميات المذكورة أعلاه والتي **تستخدم** حسابات الأعداد الصحيحة. لكل عنصر صورة يكون عنصر الصورة التالي E أو NE كما هو موضح بالشكل 3.2.



الشكل 3.2: خوارزمية النقطة الوسيطة

وتستخدم خوارزمية النقطة الوسيطة هذا **التعريف غير المباشر للخط، F(x,y)=0**.

❖ **تحديد عنصر الصورة:**

إن تحديد عنصر الصورة التالي E أو EN يتم كالآتي (انظر الشكل 3.3):

حدد

$$d = F(x_p + 1,\ y_p + 0.5$$

أذا كان $d < 0$ يكون الخط أسفل النقطة الوسيطة، اختر E

$d > 0$ يكون الخط أعلى النقطة الوسيطة، اختر NE

$F(x,y)<0$ يكون الخط على النقطة الوسيطة

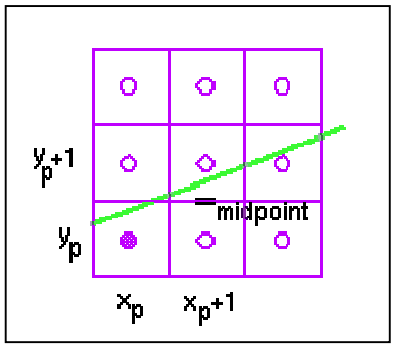
إن تم اختيار E

$$d_{new} = F(x_p + 2,\ y_p + 0.5)$$
$$d_{new} - d_{old} = F(x_p + 2,\ y_p + 0.5) - F(x_p + 1,\ y_p + 0.5)$$
$$\Delta d = d_{new} - d_{old} = dy$$

إن تم اختيار NE

$$d_{new} = F(x_p + 2,\ y_p + 1.5)$$
$$\Delta d = dy - dx$$

تكرر هذه العملية في خطوات على طول x لتحديد E أو NE

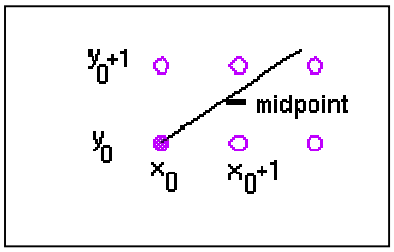


الشكل 3.3: تحديد عنصر الصورة التالي في خوارزمية النقطة الوسيطة

❖ **القيمة الابتدائية للخط:**

تحدد القيمة الابتدائية للخط d_{start} باستخدام المعادلة التالية (انظر الشكل 3.4)

$$\begin{aligned} d_{start} &= F(x_0 + 1,\ y_0 + 0.5) \\ &= (x_0 + 1 - x_0)dy - (y_0 + 0.5 - y_0)dy \\ &= dy - dx/2 \end{aligned}$$



الشكل 3.4: تحديد القيمة الابتدائية

❖ **خوارزمية الأعداد الصحيحة:**

لاستخدام خوارزمية أعداد صحيحة فقط، **عرف**

$$F'(x, y) = 2F(x, y)$$
$$d' = 2d$$
$$d'_{start} = 2dy - dx$$
$$\Delta d' = 2\Delta d$$

مثال:

الخوارزمية أدناه تنتج خطاً مستقيماً يكون فيه x1 اقل من x2 والانحدار يساوى أو يقل عن 1:

```
drawline(x1, y1, x2, y2, colour)
int x1, y1, x2, y2, colour;
{
  int dx, dy, d, incE, incNE, x, y;
  dx = x2 - x1;
  dy = y2 - y1;
```

```
d = 2*dy - dx;
incE = 2*dy;
incNE = 2*(dy - dx);
y = y1;
for (x=x1; x<=x2; x++) {
  setpixel(x, y, colour);
  if (d>0) {
    d = d + incNE;
    y = y + 1;
  } else {
    d = d + incE;
  }
 }
}
```

أسئلة تقويم ذاتي

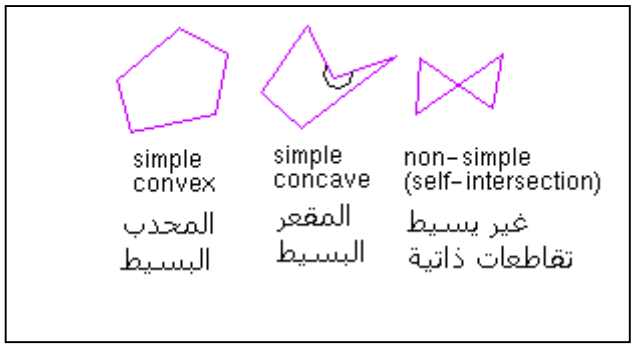


1. ما هو العنصر الأساسي في أي نموذج ثنائي الأبعاد؟
2. ما الخطوة الابتدائية لتمثيل الخطوط والمنحنيات؟
3. عدد خصائص الرسم الأمثل للخط.
4. إلى ماذا نحتاج لاستخدام خوارزمية الأعداد الصحيحة لرسم الخط؟
5. كيف يمكن تمثيل مثلث باستخدام مصفوفة؟ هات مثالاً

4.1 تمثيل المضلعات Polygons

يوجد أنواع مختلفة من المضلعات مثل (انظر الشكل 3.6)

- المحدب البسيط simple convex.
- المقعر البسيط Simple concave.
- غير بسيط (التقاطع الذاتي) non-simple (self-intersection).



الشكل 3.5: أنواع المضلعات

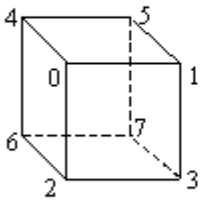
❖ **تمثيل المثلث triangle:**

من السهولة عمليا التعامل مع المثلث triangle حيث انه يكون دائما مستوى السطح ومحدب بسيط.

❖ **طريقة قائمة المراجع list of references لتمثيل المضلع:**

يمكن استخدام العديد من الطرق لتمثيل المضلع، أكثر الطرق شيوعا واستخداما هي **قائمة المراجع** list of references التي ترتب قائمة الرؤوس vertex.

إن **استخدام** هذه الطريقة **يقلل** من **سعة التخزين** و**الحسابات الزائدة** عن الحاجة. كما يمكننا **ربط وتوضيح** كثير من المعلومات الأخرى مع كل رأس **مثل المتعمدات** normals و**الألوان** colors و**إحداثيات النسيج** texture coordinates. الشكل 3.6



الأوجه
قائمة الرؤوس
faces
vertex list

#	vertex list
0	0,2,3,1
1	1,3,7,5
2	5,7,6,4
3	4,6,2,0
4	4,0,1,5
5	2,6,7,3

قائمة الرؤوس
x, y,z
vertex list

#	x,y,z
0	0,1,1
1	1,1,1
2	0,0,1
3	1,0,1
4	0,1,0
5	1,1,0
6	0,0,0
7	1,0,0

الشكل 3.6: طريقة قائمة المراجع

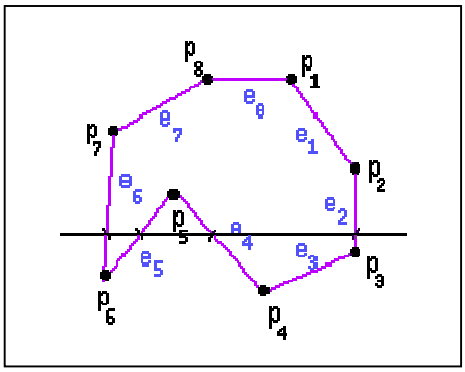
5.1 تحويل المسح Scan Conversion

المهمة **الأساسية** لتحويل المسح هي **تظليل** كل عناصر الصورة الموجودة داخل مضلع مقفول، و**يعتمد لون التعبئة** في أغلب الأحيان على **إضـــاءة** lighting **ونســـيج** texture **ورؤية** visibility المضلع الذي حول مسحه، هذه العوامل سيتم تجاهلها فـــي الوقت الحالي وسنتعرض لها في الوحدات القادمة.

❖ **خوارزمية تحويل المسح:**

لنفترض إن المضلع مغلق ولديه حواف edges مرتبة، كما هو موضح بالشكل 3.7. فإن خوارزمية تحويل المسح تكون على النحو التالي:

- ادخل خطاً مع كل حافة.
- أفرز التقاطعات مع x.
- احسب تكافؤ parity التقاطعات لتحديد الدخل والخرج.
- عبئ عناصر الصورة.

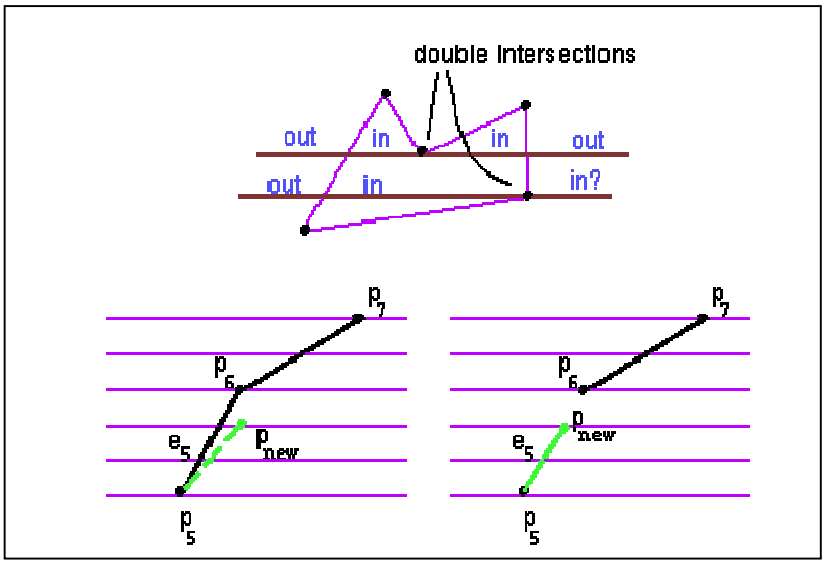


الشكل 3.7: تحويل مسح المضلع

❖ **الحالات الخاصة لخوارزمية تحويل مسح المضلع:**

توجد حالات خاصة لهذه الخوارزمية وهي:

- الحافات الأفقية يمكن استثناؤها.
- بالنسبة للرؤوس الواقعة على خط المسح.
- غير علامة $y_i - y_{i+1}$: تحسب مرتين (الشكل 3.7a).
- لا تغير قصر الحافة بخط مسح واحد (الشكل 3.7b).



الشكل 3.6: الحالات الخاصة لخوارزمية مسح المضلع

❖ **الترابط المنطقي بين الخطوط:**

كثير من اختبارات التقاطع يمكن التخلص منها بالاستفادة من الترابط المنطقي بـــين الخطوط الممسوحة وهى:

- كل الحافات التي تقاطع الخط الممسوح y تقاطع أيضا الخط y+1.
- تغير قيمة x يمكن التنبؤ بها من خط المسح y إلى y+1.

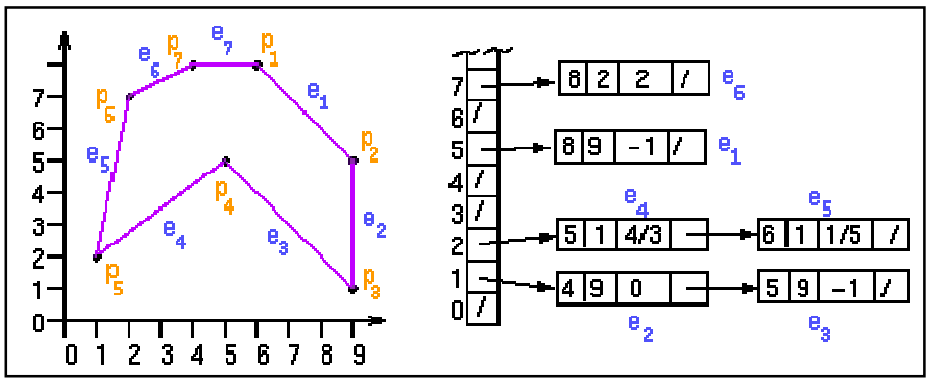
❖ **هياكل البيانات:**

هياكل البيانات التالية يمكن استخدامها بواسطة خوارزمية تحويل مسح ذات كفاءة عالية:

1) جدول الحافة:

ويحدد هيكل البيانات هذا بالتالي (انظر الشكل 3.8)

- اعتبر حافات الانحراف traverse edge.
- استثني الحافات الأفقية.
- إن لم يكن حافة قصوى، قصر الرأس الأعلى.
- أضف الحافة إلى قائمة ربط خط المسح المقابل للرأس الأدنى، وفرز:
  1. قيمة y العليا: أخر خط مسح يأخذ في الاعتبار.
  2. قيمة x الدنيا: بداية إحداثيات x للحافة.
  3. 1/m. لزيادة قيمة x، تحسب قبل التقصير.



الشكل 3.7: جداول الحافة

2) قائمة الحافة النشطة Active Edge List (AEL):

تمثل قائمة الحافة النشطة قائمة ربط للحواف النشطة في خط المسح الحـــالي y. أي حافة نشطة لديها المعلومات التالية:

- قيمة y العليا: آخر خط مسح يأخذ في الاعتبار.
- قيمة x الدنيا: بداية إحداثيات x للحافة.
- 1/m لزيادة قيمة x، تحسب قبل التقصير.

ويتم فرز الحواف النشطة باستخدام x.

❖ **مكونات خوارزمية تحويل المسح:**

إن خوارزمية تحويل المسح تتكون من التالي:

- لكل خط مسح.
- أضف الحافات في جدول الحافة المحدد في قائمة الحافة النشطة.
- إذا كانت قائمة الحافة النشطة غير فارغة؛
  - **صنف** القائمة باستخدام x.
  - **عبئ** عناصر الصورة بين زوج الحافات.
  - **امسح** الحافات التي انتهى التعامل معها.
  - حدث قيمة x لكل حافة.

أسئلة تقويم ذاتي

1. للمضلعات أنواع اذكرها.
2. لماذا كان من السهل التعامل مع المثلث؟
3. على ماذا يعتمد لون التعبئة في عملية تحويل المسح؟
4. ما فائدة استخدام طريقة قائمة المراجع لتمثيل المضلع؟
5. بالاستفادة من الترابط المنطقي بين الخطوط الممسوحة أي اختبارات التقاطع يمكن التخلص منها؟
6. كيف يتم فرز الحواف النشطة؟

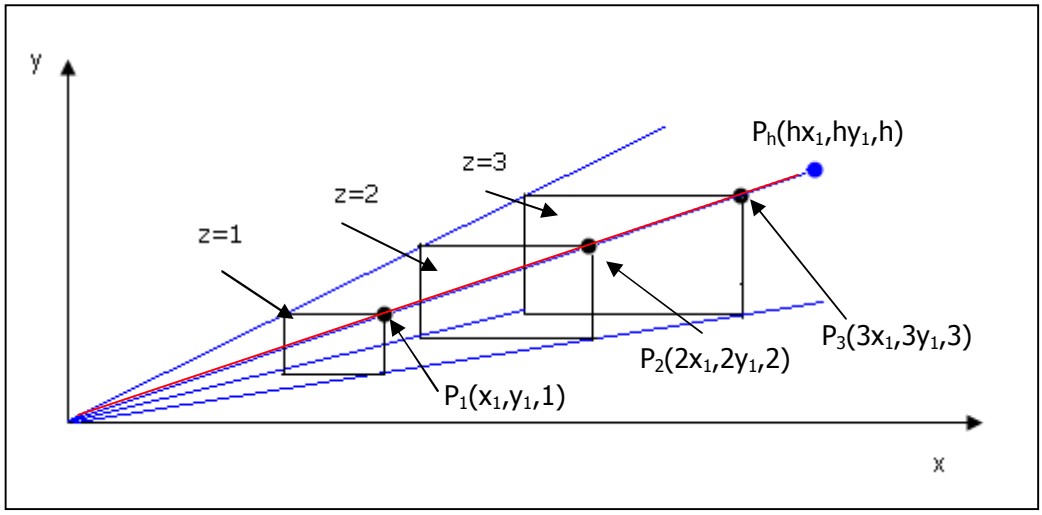
6.1 نظام الإحداثيات المتجانس

homogenous coordinate system

حتى نتمكن من **استخدام المصفوفات** في كل عمليـــات **التحـــويلات الهندســـية** نستخدم **نظام الإحداثيات المتجانس** homogenous coordinate system.

إن **استخدام** نظام الإحداثيات المتجانس **يعطى** طريقة موحدة لوصف التحـــويلات الهندسية.

ولتوضيح مفهوم نظام الإحداثيات المتجانس، اعتبر نقطة النظام الثنائي الأبعـــاد $P_1(x_1,y_1)$ التي تقع في النظام الثلاثي الأبعاد كما هو موضح في الشكل 5.3.



الشكل 3.8. التمثيل الثلاثي الأبعاد في الفضاء المتجانس

❖ **الفضاء المتجانس homogenous space:**

النقاط في الإشعاع الذي يصل النقطة P_1 إلى نقطة تقاطع محـــاور الإحـــداثيات يمكن التعبير عنه باستخدام العامل h على النحو التالي:

$$P(x,y,z) = P(hx, hy,h)$$

أي نقطة ثنائية الأبعاد ما عدا **نقطة تقاطع محاور الإحداثيات** (h=0) يمكن تمثيلها بأحد النقاط في الإشعاع في فضاء ثلاثي الأبعاد (**يسمى الفضاء المتجانس homogenous space**).

بصورة عامة في نظام ألإحداثيات المتجانس، أي **نقطة ديكارتية** يمكن تمثيلها بهيئة زائدة باستخدام **أربع إحداثيات** (hx,hy,hz,h).

❖ **الإحداثيات العادية:**

الإحداثيات العادية تناظر النقطة عندما يعبر الإشعاع السطح z=1. وعند استخدام نظام الإحداثيات المتجانس النقاط في المجال الثنائي الأبعاد تمثل بمصفوفة مرتبتها nx3، عندما تكون n عدد الرؤوس في الكائن. فمثلا المثلث يوصف بالمصفوفة التالية:

$$[P]_{TRIANGLE} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

7.1 التحويلات الثنائية الأبعاد Two Dimensional Transformation

تشتمل التحويلات الهندسية **حساب** الإحداثيات الجديدة للنقاط **التي** تشكل الكائن **من** نقطة تقاطع محاور الإحداثيات **إلى** موقع التحويل.

❖ **ضبط الحجم scaling:**

يسمح تحويل إعادة الحجم **بتغير** الكائن سواء **بالتمديد** أو **الانكماش، رياضيا** يعبر عن تحويل إعادة للحجم للنقطة P(x,y) إلى النقطة P'(x',y') **بالعلاقة** التالية:

$$x' = x \cdot S_x \qquad 3.6a$$

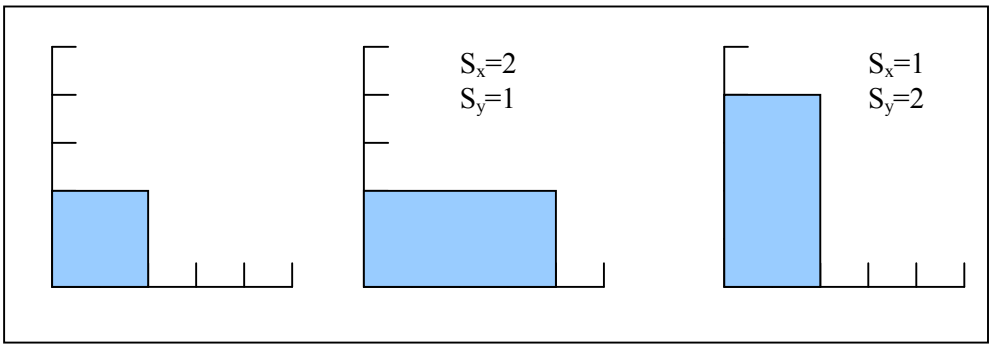
$$y' = y \cdot S_y \qquad 3.6b$$

حيث s_x و s_y ثوابت تغير الحجم في اتجاه x و y على التوالي.

يمكن كتابة هذه المعادلة على **نسق المصفوفة** كالآتي:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.7$$

الشكل 3.9 يوضح أمثلة **لتأثير** تحويل الحجم على **مربع**، و**يلاحظ** أنــه عنــدما تكون عوامل إعادة الحجم **متخلفة** في اتجاه x و y **يحدث تغير** في حجم وشكل الكائن.

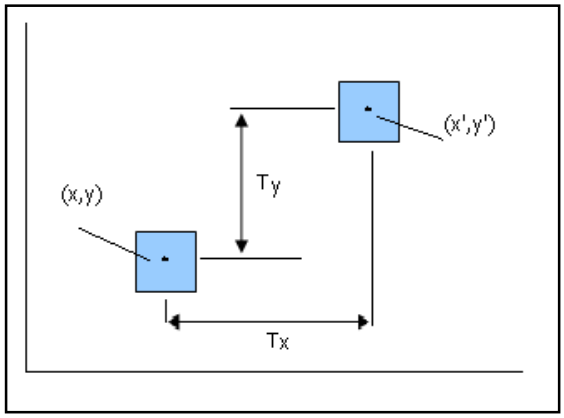


الشكل 3.9: تحويل ضبط الحجم

يسمى هذا التحويل **ضبط الحجم** حول نقطة تقاطع محاور الإحــداثيات، **ويســمى التحويل منتظم** uniform **إذا كان** $S_x=S_y$.

❖ **تحويل الإزاحة Translation:**

تطبيق الإزاحة **تغير موقع** الكائن على العارض **دون أن تغير** حجمه أو شــكله كما هو موضح بالشكل 3.10.



الشكل 3.10: تحويل الإزاحة

يعبر عن هذا التحويل **رياضيا باستخدام**

$$x' = x \cdot T_x \qquad 3.8a$$
$$y' = y \cdot T_y \qquad 3.8b$$

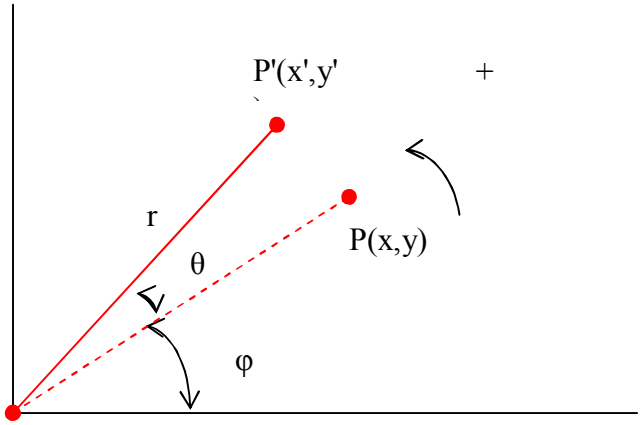
يمكن كتابة هذه المعادلة على **نسق المصفوفة** كالآتي:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \qquad 3.9$$

❖ **تحويل الدوران Rotation:**

بتطبيق تحويل **الدوران** يتم لف الكائن **بزاوية مقدارها θ** حول نقطة تقاطع المحاور. الشكل 3.9 يوضح دورات النقطة P(x,y) بالزاوية إلى النقطة P'(x',y').

من المتعارف عليه إن كان **الدوران** في **عكس اتجاه** الساعة يكون الدوران **موجباً** وإنا كان **في اتجاه الساعة** يكون **سالباً**.



يمكن حساب موقع النقطة P'(x',y') **باستخدام** علاقات حساب المثلثات trigonometric التالية:

$$x = r\cos\varphi$$
$$y = r\sin\varphi \qquad 3.10$$

حيث إن العوامل r و φ موضحة في الشكل 3.9. و**أيضا:**

$$x' = r\cos(\varphi+\theta) = r\cos\varphi\cos\theta - r\sin\varphi\sin\theta$$
$$y' = r\sin(\varphi+\theta) = r\sin\varphi\cos\theta + r\cos\varphi\sin\theta \qquad 3.11$$

عوض المعادلة 3.10 **في** 3.11 **تتحصل** على:

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \qquad 3.12$$

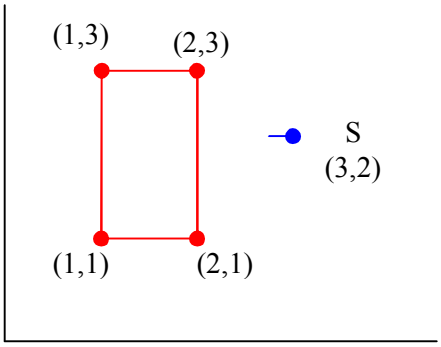
يمكن كتابة هذه المعادلة على **نسق المصفوفة** كالآتي:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1]\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.13$$

من العادة قد يحتاج الشكل لكثير من التحولات لمشاهدته، وفي مثل هذه الحالات يتم تركيب التحويل من التحويلات الموضحة أعلاه كما سيوضح في الأمثلة التالية.

مثال(1)

دور (لف) **المستطيل** المكون من النقاط $P_1(1,1)$ و $P_2(2,1)$ و $P_3(2,3)$ و $P_4(1,3)$ **30 درجة** في **عكس اتجاه الساعة** حول **النقطة** S(3,2).



الحل:

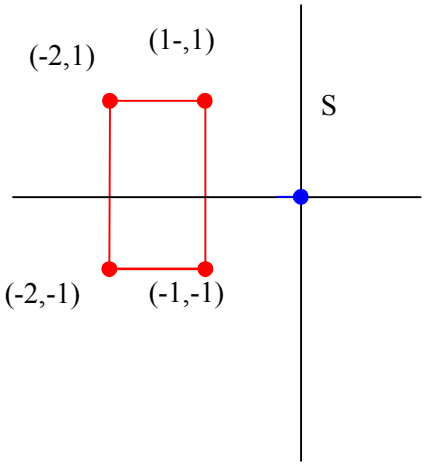
يشتمل الحل على التحويلات المتتالية التالية:

1. **إزاحة** النقطة S إلى نقطة تقاطع محاور الإحداثيات

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & -2 & 1 \end{bmatrix}$$

- **مصفوفة التحويل**

إزاحة النقطة S **ينتج** مباشرة إزاحة المستطيل لموقع جديد كما هو موضح أدناه

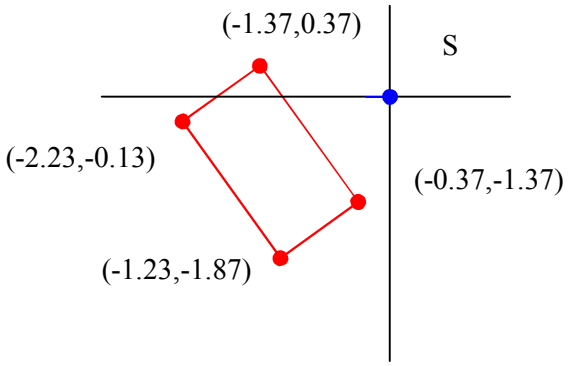


2. **الدوران** بثلاثين درجة عكس عقارب الساعة

- **مصفوفة التحويل**

$$\begin{bmatrix} 0.8660 & 0.50 & 0 \\ -0.50 & 0.8660 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- الشكل أدناه يوضح **موقع المستطيل بعد الدوران**

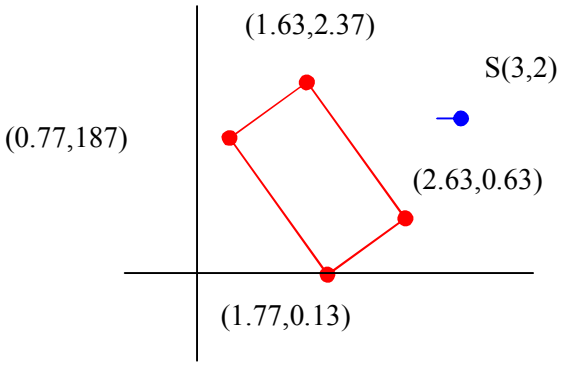


3. **أزاحة** S مرة أخرى لموقعها الأول

- **مصفوفة التحويل**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

- الشكل أدناه يوضح **الموقع الأخير المستطيل**



مثال(2)

للنماذج التي **توصف** بعدد كبير من النقاط، يمكن **استخدام** طريقة حساب أكثر كفاءة للحصول على **التحويل** وذلك **بضرب المصفوفات** أولا ثم **نستخدم مصفوفة النقطة.**

المثال أعلاه **يحل** كالآتي:

$$[T]=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -3 & -2 & 1\end{bmatrix}\begin{bmatrix}0.866 & 0.5 & 0\\ -0.5 & 0.866 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 3 & 2 & 1\end{bmatrix}$$

$$=\begin{bmatrix}0.866 & 0.5 & 0\\ -0.5 & 0.866 & 0\\ 1.4 & -1.23 & 1\end{bmatrix}$$

لإيجاد النقاط [P'] **نستخدم**

$$[P']=[P][T]=\begin{bmatrix}1 & 1 & 1\\ 1 & 3 & 1\\ 2 & 3 & 1\\ 2 & 1 & 1\end{bmatrix}\begin{bmatrix}0.866 & 0.5 & 0\\ -0.5 & 0.866 & 0\\ 1.4 & -1.23 & 1\end{bmatrix}$$

$$=\begin{bmatrix}1.77 & 0.13 & 1\\ 0.77 & 1.87 & 1\\ 1.63 & 2.37 & 1\\ 2.63 & 0.63 & 1\end{bmatrix}$$

❖ **الانعكاس Reflection:**

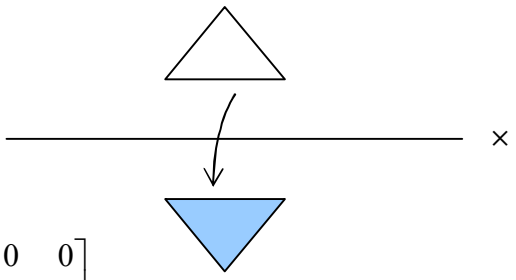
يمكن فهم الانعكاس باعتبار الصورة في المرآة، **ويستخدم تحويل الانعكاس لإنشاء العناصر المتشابهة،** حيث يمكن رسم نصف الشكل ثم يستخدم تحويل الانعكاس لإتمام المشهدأ قد يتم الانعكاس على نقطة أو خط.

مصفوفة الانعكاس بالنسبة محور x أو y أو نقطة تقاطع المحاور يمكن كتابتهـــا علـــى النحو التالي:

$$[T_{REL}] = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.14$$

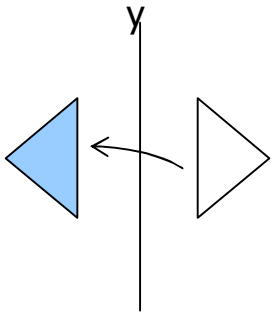
الأمثلة أدناه توضح حالات **مختلفة** لعكس مثلث

- **على محور x:** قيمة x **لا تتغير** و**تتغير** قيمة y



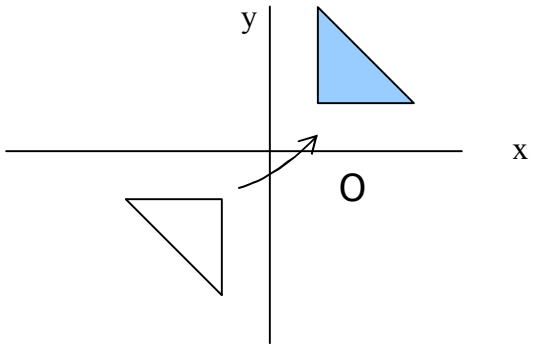
$$[T_{REL}]_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.15$$

- **على محور y:** قيمة y **لا تتغير** و**تتغير** قيمة x



$$[T_{REL}]_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.16$$

- **على نقطة تقاطع المحاور: تتغير** قيمة كل من x و y



$$[T_{REL}]_O = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.17$$

❖ تحويل القص Shearing:

يغير تحويل القص **قيمة** الأحداث **بإضافة دالة خطية** للأحداث الأخر إليه.

المصفوفة العامة للقص تكون على الهيئة التالية:

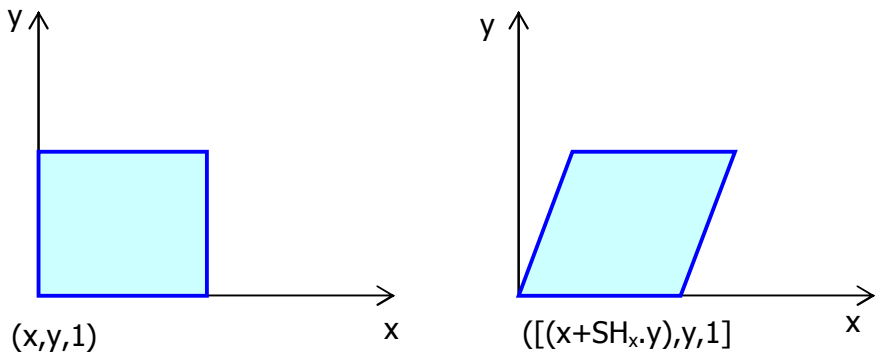
$$[T_{SH}] = \begin{bmatrix} 1 & b & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

مثال:

يوضح قص المربع لبعض الحالات.

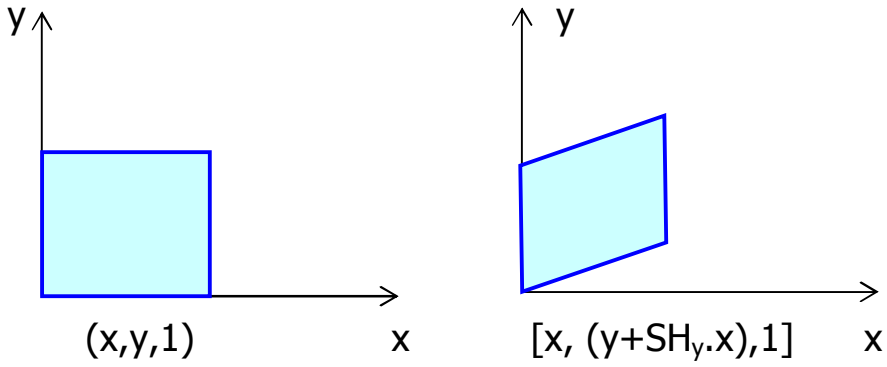
- **قص اتجاه-x يؤثر فقط على إحداثيات x**

$$[T_{SH}]_x = \begin{bmatrix} 1 & 1 & 0 \\ SH_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- **قص اتجاه–y يؤثر فقط على إحداثيات y**

$$[T_{SH}]_y = \begin{bmatrix} 1 & SH_y & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

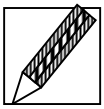


أسئلة تقويم ذاتي



1. لماذا نستخدم نظام الإحداثي المتجانس في التحويلات الهندسية؟
2. ما هي النقطة ثنائية الأبعاد التي لا يمكن تمثيلها في الفضاء المتجانس؟
3. ما الفرق بين تحويل الإزاحة وضبط الحجم؟
4. لتحويل الانعكاس فائدة اذكرها.

تدريب (2)

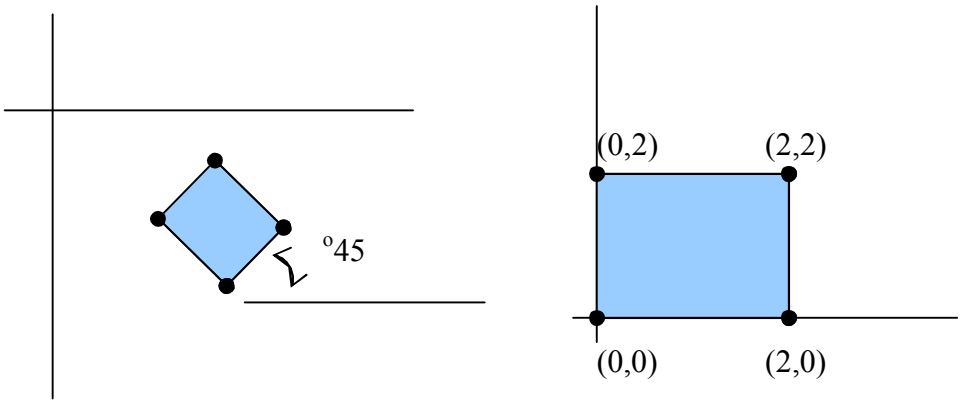


1.وضح أن الدائرة التي تتعرض التحويل الذي يعطى بالمصفوفة

$$[T]=\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

لا تبقى دائرة بعد التحويل، ما هو الشكل بعد التحويل.

2. كبر المثلث المعطى في المسألة الأولى إلى مرتين حجمه مع بقاء النقطة C في موقعها. أحسب إحداثيات النقاط الثلاثة بعد التحويل.
3. أشتق مصفوفة للانعكاس على الخط y=x.
4. اكتب متوالية من التحويلات على نسق المصفوفات التي تحتاجها لوضع المربع الموضح في الشكل 3.11 في الوضع الموضح في الشكل 12 مع تصغير حجمه للنصف، مركز المربع بعد التحويل عند النقطة (2,-2).



الشكل 3.11

الخلاصة

عزيزي الدارس، لتمثيل الأشكال ثنائية الأبعاد نستخدم واصفات عددية ثنائية الأبعاد و لإزاحة حجم الشكل أو الكائن ودورانه وإعادته نستخدم عدداً من التحولات الهندسية، وتمثل النقطة العنصر الأساسي في أي نموذج ثنائي الأبعاد.

يمكن استخدام المصفوفة في التمثيل ثنائي الأبعاد، وتدعم المصفوفة عدداً من الميزات كتحديد الشكل وكيفية التعامل معه في التخطيط بالحاسوب، كما يمكن تمثيل الخطوط تمثيلا أفضل ويمكن استخدام عدد من المعادلات لتمثيلها، أما تمثيل المضلعات فيتم بطرق مختلفة أشهرها طريقة قائمة المراجع حيث يتم ترتيب قائمة الرؤوس، ويمكن استخدام تحويل المسح لتظليل كل عناصر الصورة الموجودة داخل مضلع مقفول.

إن استخدام نظام الإحداثيات المتجانس يعطي طريقة موحدة لوصف التحويلات الهندسية؛ فأي نقطة ديكارتية يمكن تمثيلها بهيئة زائدة باستخدام أربع إحداثيات (hx,hy,hz,h). تشمل التحويلات ثنائية الأبعاد حساب الإحداثيات الجديدة للنقاط التي تشكل الكائن من نقطة تقاطع محاور الإحداثيات إلى موقع التحويل، حسب طريقة التحويل المتبعة من ضبط الحجم أو الإزاحة أو غيرها ...

لمحة مسبقة عن الوحدة التالية

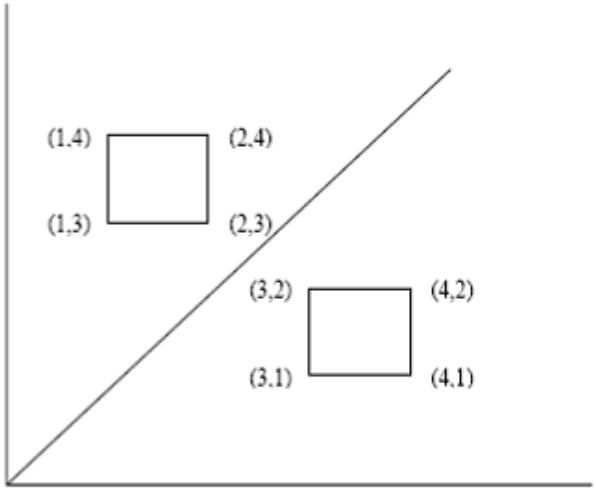
في الوحدة الرابعة التالية سوف تتعلم عمليات القص الثنائي الأبعاد، وقص النقطة، وقص الخط وخوارزمية كوهين– سذرلاند التي تمكنك من مشاهدة المناظر الثنائية الأبعاد.

إجابات التدريبات

تدريب(1)

أشتق مصفوفة للانعكاس على الخط y=x.

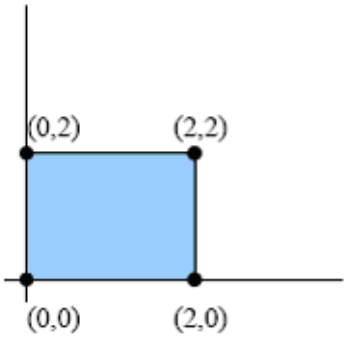
من الشكل ادناه يمكننا ان نلاحظ ان كل نقطة في الشكل المنعكس هي عبارة عن نفس النقطة في الشكل الأصلي مع إبدال الإحداثي السيني بالصادي و العكس صحيح.



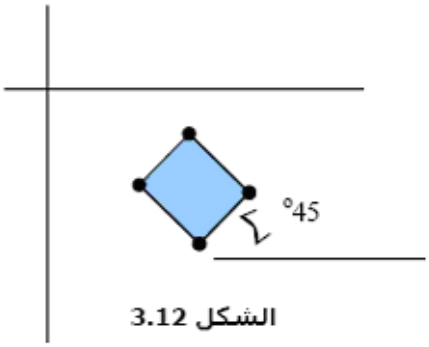
و عليه تصبح مصفوفة النقل هي:-

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. اكتب متوالية من التحويلات على نسق مصفوفات التي تحتاجها لوضع المربع الموضح في الشكل 3.11 في الوضع الموضح في الشكل 12 مع تصغير حجمه للنصف. مركز المربع بعد التحويل عند النقطة (2-,2).



الشكل 3.11



الشكل 3.12

-تقليل حجم المربع إلى النصف:

$$\begin{bmatrix} 0.7071 & 0 & 0 \\ 0 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-عكس المربع بالنسبة للمحور السينى:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-عمل دوران للمربع بزاوية 45 درجة:

$$\begin{bmatrix} 0.7071 & 0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-المركز الآن هو (1,0) و يجب نقله إلى (2-،2) :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

***الدالة التي تستخدم طريقة النقطة الوسيطة لرسم الخط المستقيم مع تحديد خصائص اللون و السمك و الصلابة هي كالآتي:

```
void draw_line(int x1,int y1,int x2,int y2,int color,int thick,int solid)
{
 int dx,dy,d,incE,incNE,x,y;
 dx = x2 - x1;
 dy = y2 - y1;
 d = 2*dy - dx;
 incE = 2*dy;
 incNE = 2*(dy - dx);
 for(int i=0; i<thick; i++)
 {
  y = y1+i;
  for(x=x1; x<=x2; x=x+solid+1)
  {
   putpixel(x,y,color);
   if(d>0)
   { d = d+incNE;
     y = y+solid+1; }
   else
   {  d = d+incE;   }
  }
 }
}
```

مسرد المصطلحات

الرسم الأمثل للخط : هو رسم الخط على شبكة مسح الخطوط raster grid.

خوارزميات رسم الخط المستقيم: هي الطرق المستخدمة لرسم الخط المستقيم.

الفضاء المتجانس:هو فضاء ثلاثي الأبعاد يمكن أن تمثل فيه أي نقطة ثنائية الأبعاد بأي نقطة على الإشعاع ما عدا نقطة تقاطع محاور الإحداثيات (h=0).

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة الرابعة
عمليات مشاهدة المناظر الثنائية الأبعاد
Two-Dimensional Viewing Operations

محتويات الوحدة

الموضوع	الصفحة
المقدمة	93
تمهيد	93
أهداف الوحدة	**93**
1. القطع (القص) الثنائي الأبعاد Two-Dimensional Clipping	**94**
2. قص النقطة Point Clipping	**95**
3. قص الخط Line Clipping	**96**
4. خوارزمية كوهين– سذرلاند Cohen-Sutherland	**97**
لمحة مسبقة عن الوحدة الدراسية التالية	**107**
إجابات التدريبات	**107**
مسرد المصطلحات	**108**
المراجع	**110**

المقدمة

تمهيد

مرحباً بك في الوحدة الرابعة وهي بعنوان عمليات مشاهدة المناظر الثنائية الأبعاد وفي داخل هذه الوحدة تجد عزيزي الدارس القص الثنائي الأبعاد وقص النقطـــة وقص الخط خوارزمية كوهين–سذرلاند كما أننا قد أوردنا فيها أمثلة وأسئلة تقويم ذاتي وتدريبات لتعميق الفهم والتطبيق. نرجو أن نستفيد منها كما نرجو أن تشاركنا في نقـــدها وتقويمها.

والآن إلى الأهداف التى يجب أن تحققها الوحدة.

أهداف الوحدة



عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة يجب أن تكون قادراً على أن:

(1) **توضح** عمليات مشاهدة المناظر الثنائية الأبعاد.

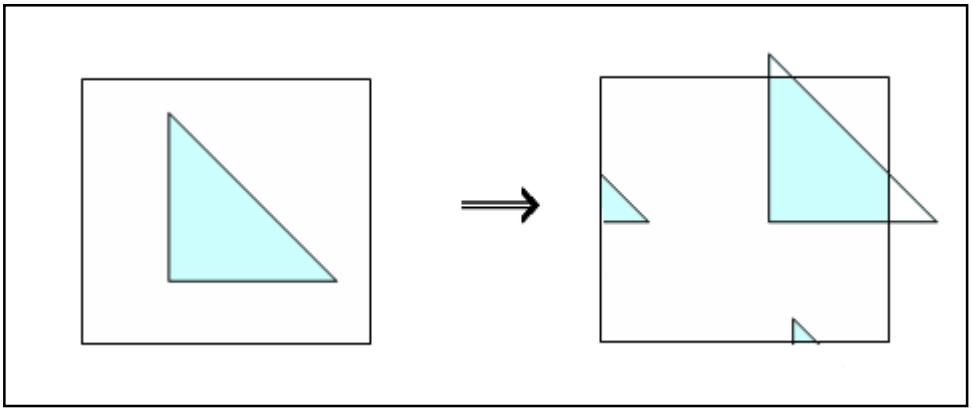
(2) **تحقق** قص النقطة.

(3) **تستخدم** الخوارزمية التي تقص الخط

(4) **تشرح** خوارزمية كوهين–سذرلاند و تبين ومراحلها.

1. القطع (القص) الثنائي الأبعاد Two-Dimensional Clipping

عملية القطع تهتم بقص جزء من المنظر من النافذة، مثل قص صورة من مجلة. ويمكن اعتبارها تحويلاً هندسياً مع قص أو قطع كل الخطوط والمنحنيات التي تعبر حدود النافذة. **يقسم القص** إي المنظر إلى جزئيين **جزء مشاهد** وآخر **غير مشاهد. إن لم يتم القطع** (أو القص) قد **يظهر** جزء من المنظر الذي يقع **خارج حدود** (حافات) **النافذة** في الجهة المقابلة (العكسية) منها، كما هو موضح في الشكل 4.1. رغم أن حدوث هذا **يقل** مع استخدام **تقنيات العتاد الحديثة** فمن الجيد **التخلص** من جزء المنظر الذي يقع خارج حدود النافذة (تأثير التغليف wraparound).



الشكل 4.1: تأثير التغليف Wraparound Effect

تعتمد **خوارزميات القص بالكامل** على **شكل النافذة.** في هذا المقرر ستفترض نافذة **مستطيلة جنبتاها موازية لمحاور الإحداثيات الرئيسة.**

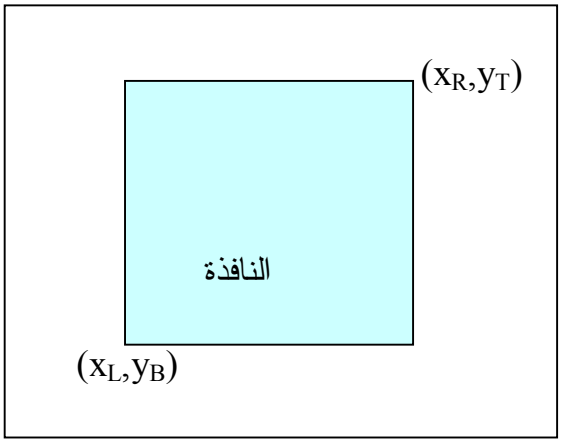
2. قص النقطة Point Clipping

يتحقق قص النقطة بسهولة **بمضاهاة إحداثيات النقطة** مع حدود النافـذة، فـإذا كانت النافذة محددة بالأركان الأربعة يسار (L) ويمين ® وأعلى (T) وأسفل (B) كمـا هو موضح فـي الشـكل 4.2، **تكـون النقطـة مشـاهدة إن اسـتوفيت المتباينـات التالية inequalities:**

$$x_L \le x \le x_R \qquad 4.1a$$

$$y_B \le y \le y_T \qquad 4.1b$$

إن لم تستوفى أي من هذه المتباينات فإنه لا تعرض النقطة. ورغم سهولة هـذه الطريقة إلا أنه ليس عملياً تقسيم كل الشكل إلى نقاط، وعليه لابد من **خوارزميـة أكثـر كفاءة** تعتمد على **الخطوط والمضلعات.**



الشكل 4.2: حدود النافذة

3. قص الخط Line Clipping

الخوارزمية التي تستخدم لقص الخط تنقسم إلى جزئيين

- **فحص** كل مقاطع الخطوط وفصل المقاطع التي تتقاطع مع حدود النافذة.
- **قطع** (قص) المقاطع المتحصل عليها في الخطوة الأولى بحساب تقاطعها مع حدود النافذة.

الخطوة الأولى يمكن تحقيقها بتقسيم الخطوط كالآتي:

- **خطوط مشاهدة:**

في هذه الحالة تكون نقاط نهاية المقطع بالكامل داخل حدود النافذة. مقطع الخط AB في الشكل 4.3 ضمن هذه المجموعة، ويتم عرض هذه الخطوط دون أي اختبارات أخرى.

- **خطوط غير مشاهدة:**

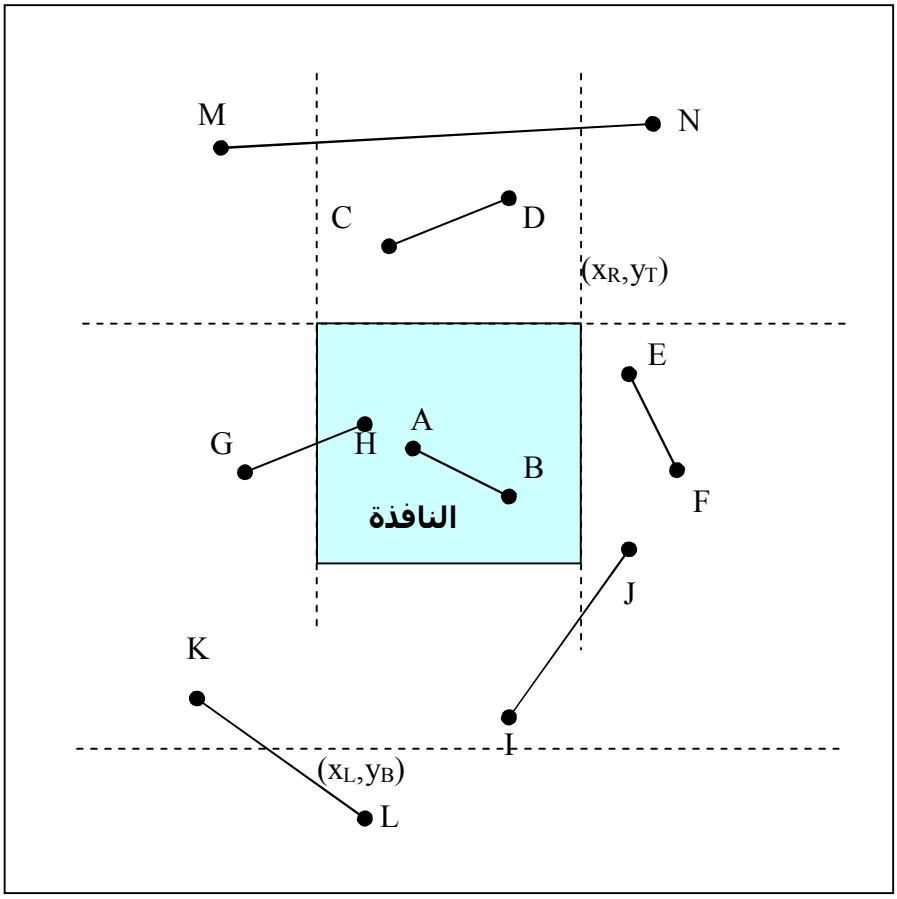
إذا كان نقطتا النهاية الاثنتان خارج حدود النافذة فإنه يكون الخط من النقطة $P_1(x_1,y_1)$ إلى النقطة $P_2(x_2,y_2$ غير مشاهد إن لم تستوفى أي من المتباينات التالية:

$$\begin{aligned} &x_1 \text{ and } x_2 < x_L \\ &x_1 \text{ and } x_2 > x_R \\ &y_1 \text{ and } y_2 < y_B \\ &y_1 \text{ and } y_2 > y_T \end{aligned} \qquad 4.2$$

مقاطع الخطوط CD و EF تستوفى كل المتباينات في 4.2.

- **غير محدد :**

لا يتضمن مقطع الخط في أي من الحالات أعلاه، ويجب أن تفحص للقص، والمقاطع GH و IJ و KL في الشكل أمثلة لهذا النوع من الخطوط.



الشكل 4.3: مواقع مقاطع الخطوط المختلفة بالنسبة للنافذة

4.خوارزمية كوهين–سذرلاند Cohen-Sutherland

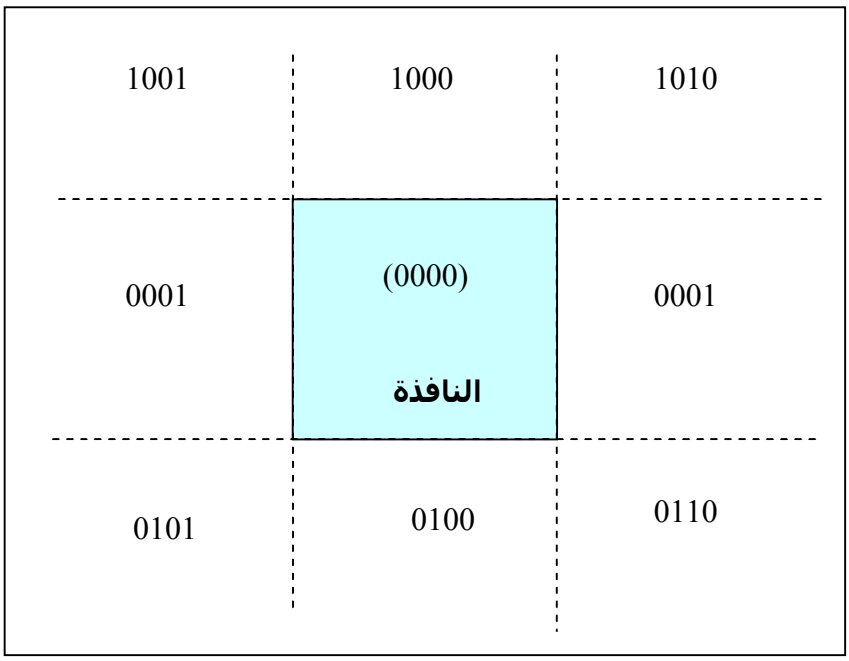
خوارزمية كوهين–سذرلاند تمثل إجراءات بسيطة **وذات كفاءة عالية لتصـــنيف مقاطع الخطوط بالنسبة لحدود النافذة** و تطبق الخوارزمية على **مرحلتين:**

- **المرحلة الأولى:**

يحدد كل نقطة نهاية لمقطع الخط الذي يراد فحصه شفرة code من 4 ثنائيـــات bits وذلك **اعتمادا** على **المناطق التسع** التي تحيط وتشتمل على النافذة، كما هـــو موضح في الشكل 4.4.

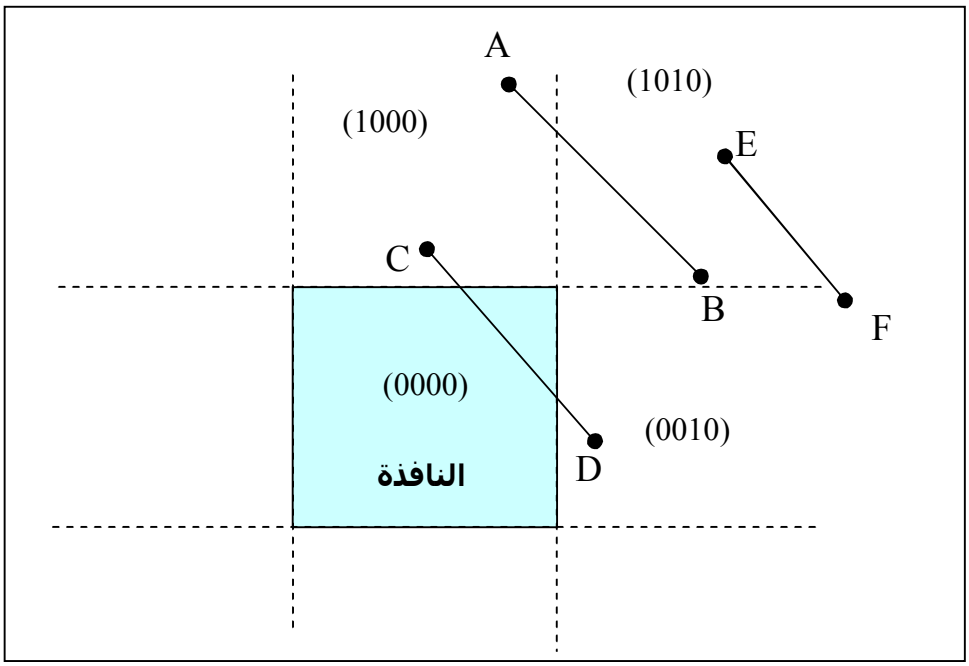
كل ثنائية في الشفرة تأخذ **القيمة 1** (صحيح) أو **القيمة 0** (خطاء) إبتداءً من أقصى اليسار على ضوء ما يلي:

- الثنائية ألأولى = 1 : نقطة نهاية المقطع أعلى النافذة
- الثنائية الثانية = 1 : نقطة نهاية المقطع أسفل النافذة
- الثنائية الثالثة = 1 : نقطة نهاية المقطع يمين النافذة
- الثنائية الرابعة = 1 : نقطة نهاية المقطع يسار النافذة



الشكل 4.4: شفرات مناطق النقطة النهائية

من البديهي إذا كانت ثنائيات الشفرة **تساوي (0000)** أن **تكون نقطة النهاية داخل النافذة أو عند حافتها**. وكل **ثنائيات** الشفرات **تحدد بمقارنة** إحداثيات نقطة نهاية مقطع الخط **مع** إحداثيات حدود (حافات) النافذة و الشكل 4.5 يعطى أمثلة لشفرة النقطة النهائية.



الشكل 4.5: أمثلة لشفرات النقاط النهائية

المرحلة الثانية:

يتم فحص النقاط النهائيةلكل مقاطع الخطوط مع بعضها البعض بناء على ما يلي:

- إذا كانت نفس الثنائية قد تم ضبطها على 1 (صحيح) في كل من شفرات النقاط النهائية، فإنه يكون المقطع غير مشاهد. فمثلا المقطع EF في الشكل 4.4 لديه شفرات نقاط نهائية تساوي (1010) و (0010) يقع على يمين النافذة.
- إذا ساوت الشفرة (0000) في كل نقاط نهاية المقطع، فإنه يكون المقطع بالكامل داخل النافذة.
- إن كانت قيمة الثنائيات تساوى 1 في مواضع مختلفة من الشفرتين يكون المقطع غير محدد؛ مثل الخطوط AB و CD في الشكل 4.4 لديهم شفرات نقاط نهائية تساوى (1000) و (0010)، فمثل هذه الخطوط قد لا تعبر النافذة، فمثلا الخط AB يكون غير مشاهد أما الخط CD فمشاهد جزئيا ويجب قصه.

أسئلة تقويم ذاتي



1. ماذا نعني بتأثير التغليف wraparound؟
2. تعتمد خوارزمية القص على

أ/ شكل النافذة. ب/ إحداثيات المحاور. ج/ نقطة تقاطع الإحداثيات (اختر الإجابة)

3. لفحص كل مقاطع الخطوط فإننا نقسمها إلى ثلاثة أقسام ما هي هذه الأقسام؟
4. اكتب المتباينات التي تجعل النقطة مشاهدة؟

❖ **الإضافة المنطقية بطريقة الثنائيةbitwise logical AND :**

لفحص الحالة السابقة يستخدم منطق خاص يعرف بمنطق "الإضافة المنطقية بطريقة الثنائية" bitwise logical AND، حيث يفترض إن ما يلي صحيح عند فحص شفرات النقاط النهائية ثنائية بثنائية:

نقطة النهاية 1 Endpoint 1		نقطة النهاية 2 Endpoint 2		الإضافة المنطقية بطريقة الثنائية Bitwise Logical AND
0	and	0	← خطاء ←	0
0	and	1	← خطاء ←	0
1	and	0	← خطاء ←	0
1	and	1	← خطاء ←	1

بناء على هذا المنطق تحدد مشاهدة الخط على النحو التالي:

مشاهد ⟵ تساوي كلاً من شفرات النقطتين النهائيتين(0000)

غير مشاهد ⟵ تختلف الشفرات ولا تساوي الإضافة المنطقية بطريقة الثنائية (0000)

غير محدد ⟵ تساوي الإضافة المنطقية بطريقة الثنائية (0000)، لكن لا تساوي كل شفرة نقطة نهائية على حده (0000)

باستخدام عملية الإضافة يتم فحص مقاطع الخطوط المعتبرة، الجدول 4.1 يعطى أمثلة للحسابات المطلوبة في هذه العملية.

جدول 4.1: حسابات المنطق

الخط	شفرات النقاط النهائية		الإضافة المنطقية	تعليق
1	1000	1001	1000	غير مشاهد
2	0101	0100	0100	غير مشاهد
3	0000	0000	0000	مشاهد
4	1001	0110	0000	فحص
5	0001	1000	0000	فحص

لفحص الخط نحتاج إلى **تحديد تقاطعه** مع حافة النافذة و **أبسط** طريقة لإيجاد **التقاطع** هي **حل** معادلات الخط والحافة **بالنسبة للنافذة المستطيلة** والتي حافتاها موازيتان لمحاور الإحداثيات و**لا نحتاج لفحص الحافات الأربع** في نفس الوقت.

وبتحديد ثنائية الشفرة التي لا تساوي 0 في ناتج ألإضافة المنطقية بطريقة الثنائية، **تحدد حافة النافذة** التي يتقاطع معها الخط كالاتى:

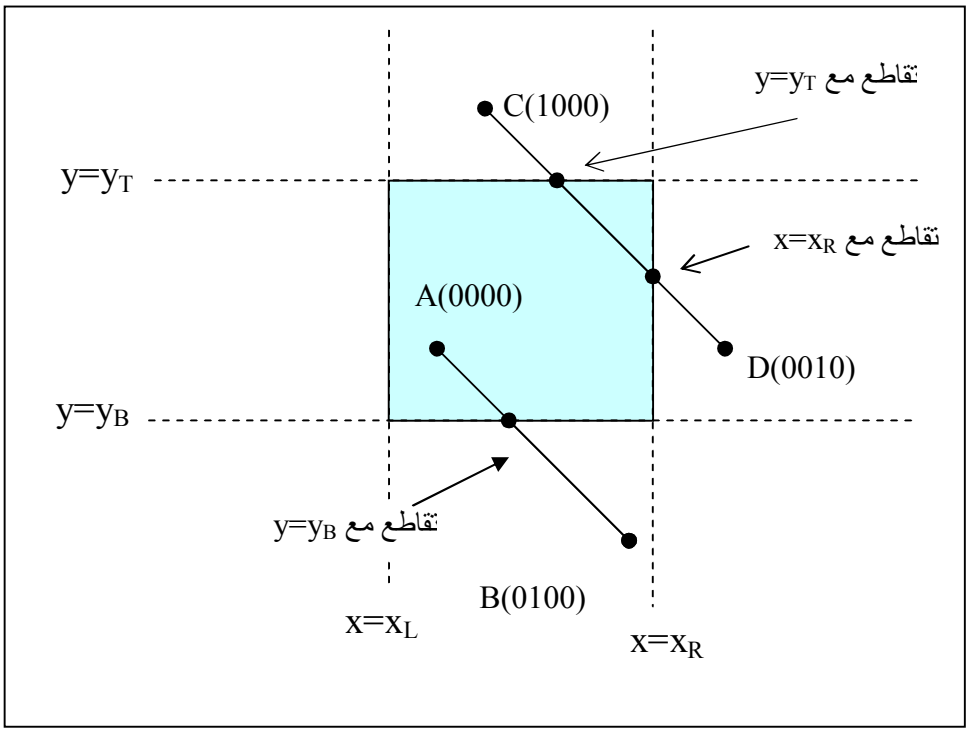
- الثنائية ألأولى = 1 : تقاطع مع $y=y_T$
- الثنائية الثانية = 1 : تقاطع مع $y=y_B$
- الثنائية الثالثة = 1 : تقاطع مع $x=x_R$
- الثنائية الرابعة = 1 : تقاطع مع $x=x_L$

الشكل 4.5 يوضح هذه العملية لخطين، يوجد التقاطع بضبط المعادلات البارامترية لكل من مقطع الخط وحافة النافذة. معادلات الخط الذي يصل النقاط $P_1(x_1,y_1)$ و $P_2(x_2,y_2)$ هي:

$$x = x_1 + t(x_2 - x_1) \qquad 4.3a$$

$$y = y_1 + t(y_2 - y_1) \qquad 4.3b$$

عند حافة النافذة تصبح المعادلات x=constant و y=constant. وتحدد نقطة التقاطع بإيجاد قيمة t من احد المعادلات 4.3 والتعويض في الأخرى.



الشكل 4.5: حساب التقاطعات

مثال:

تعطي حافات نافذة مستطيلة بالتالي:$x_L=2$

و $y_B=2$ و $x_R=8$ و $y_T=8$. افحص المشاهدة للمقاطع التالية باستخدام كوهين-ثيزرلاند وإن كان من الضروري قص مقاطع الخطوط عند حافة النافذة المناسبة.

الحل:

الخطوة الأولى: فحص الشفرات

- **الخط AB**
  - A(3,10) → (1000)
  - B(6,12) → (1000)
  - الإضافة المنطقية = **(1000)**
  - الخط **<u>غير مشاهد</u>**

- **الخط CD**
  - C(4,1) → (0100)
  - D(10,6) → (0010)
  - الإضافة المنطقية = **(0000)**
  - الخط **<u>غير محدد</u>**

الخطوة الثانية: قص الخط CD

- نقطة النهاية C لديها الشفرة (0100) حيث إن الثنائية الثانية لا تساوي صفراً، فيجب إيجاد التقاطع مع الحافة $y=y_B=2$. المعادلة البارمترية للخط CD

$$x = 4 + t(10-4) = 4 + 6t \quad 1$$
$$y = 1 + t(6-1) \quad = 1 + 5t \quad 2$$

عوض y=2 في المعادلة 2 نتحصل على قيمة t التالية:

$$t=1/5 = 0.2$$

وأيضاً

$$x = 4 + (1/5)(6) = 5.2$$

نقطة التقاطع هي $I_1(5.2,2)$

- نقطة النهاية D لديها الشفرة (0010) حيث إن الثنائية الثالثة لا تساوي صفراً، فيجب إيجاد التقاطع مع الحافة $x=x_R=8$. المعادلة البارمترية للخط CD

عوض x=8 في المعادلة 1 نتحصل على قيمة t التالية:

$$8 = 4+6t \rightarrow t=4/6 = 0.667$$

وأيضاً

$$y = 1 + (5)(2/3) = 4.33$$

نقطة التقاطع هي $I_2(8,4.33)$ **حيث إن كلاً من** I_1 **و** I_2 **على حافات النافـذة، تكـون شفرات النهاية تساوى (0000) لكل منها وعليه يكون مقطع الخط مشاهد بـين نقـاط التقاطع.**

أسئلة تقويم ذاتي



1. عرف الإضافة المنطقية بطريقة الثنائية
2. ما هو ابسط الطرق لإيجاد تقاطع الخط مع حافة الناقذة
3. ماذا نقصد بالنافذة المستطيلة؟
4. بناء على منطق الإضافة المنطقية بطريقة الثنائية متى يكون الخط:

   أ/ مشاهداً؟ ب/ غير مشاهد؟ ج/ غير محدد؟
5. كيف تحدد حافة النافذة التي يتقاطع معها الخط؟

تدريب (1)



1. لنفترض نافذة محددة بالإحداثيات التالية: (20,20-) و (60,60) في نظام الإحداثيات العالمي، اعتبر مقطع خط يعطى نقاطاً نهايته بالأتي: (30,0-) و (80,40).
   - أ- أوجد الإضافة المنطقية لشفرات النقاط النهائية
   - ب- أوجد تقاطع الخط مع حافة النافذة المناسبة
   - ج- افترض أن منفذ المشاهدة يحدد بالإحداثيات (10,30) و (200,130)، اوجد إحداثيات منفذ المشاهدة لنقاط التقاطع.
2. في السؤال رقم 1، هل يحدث حجم النافذة مقارنة مع حجم منفذ المشاهدة تشويشاً في منظر الكائن؟ اشرح إجابتك.
3. نافذة محددة بإحداثيات الركن الأسفل الأيسر (2,2) وإحداثيات الركن الأعلى الأيمن (6,8)، يراد قص مقطع الخط من A(3,1) إلى B(7,5) على هذه النافذة.
   - أ- أوجد شفرات نقاط النهاية لمقطع الخط والإضافة المنطقية لهذه الشفرات
   - ب- احسب نقاط تقاطع الخط مع حافات النافذة إن وجدت.

خلاصة

إن استخدام تقنيات العتاد الحديثة في القص قلل قد من مشكلة ظهور جزء المنظر الذي يقع خارج حدود (حافات) النافذة في الجهة المقابلة منها. ومع سهولة استخدام خوارزمية قص النقطة إلا أنه ليس عملياً تقسيم كل الشكل إلى نقاط، وعليه لابد من خوارزمية أكثر كفاءة تعتمد على الخطوط والمضلعات، ومن هنا كان استخدام خوارزمية قص الخط وخوارزمية كوهين–سذرلاند، وهي خوارزمية بسيطة وذات كفاءة عالية، تستخدم الإضافة المنطقية بطريقة الثنائية لفحص الخطوط حيث يفترض أن التالي صحيح عند فحص شفرات النقاط النهائية ثنائية بثنائية، كما نحتاج عند فحص الخط إلى تحديد تقاطعه مع حافة النافذة، وختام هذه الخلاصة عزيزي الدارس يمكن إنتاج نسخ مختلفة من المنظر باستخدام التحويلات.

لمحة مسبقة عن الوحدة التالية

الوحدة التي تلي هذه الوحدة هي عن التحويلات الهندسية الثلاثية الأبعاد وسوف نتعرف فيها على نظام الإحداثيات، والكمائن الثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)، و التحويلات الهندسية الثلاثية الإبعاد، والإزاحة، والدوران، نتمنى أن تكون وحدة مفيدة.

إجابات التدريبات

التدريب(1)

(أ) الإضافة المنطقية = 0000 و عليه فإن الخط يحتاج إلى فحص.

ب)

يمكن تمثيل النقاط على الخط بالمعادلات البارامترية التالية:

$$X = -30 + 110t$$
$$y = 40t$$

لإيجاد نقاط التقاطع نحل المعادلات البارامترية مع:

1\ $x = -20$

$-20 = -30 + 110t$

$t = 1/11$

$y = 40/11 = 3.64$

و لكن هذه النقطة تقع خارج النافذة و عليه مرفوضة.

2\ $y = 20$

$20 = 40t$

$t = 0.5$

$x = -30 + 110(0.5) = 25$

نقطة التقاطع الأولى هي (25,20) .

3\ $x = 60$

$60 = -30 + 110t$

$t = 9/11$

$y = 40*9/11 = 32.727$

نقطة التقاطع الثانية هي (60,32.727) .

ج)

منفذ المشاهدة لنقاط التقاطع.

$S_X = (200 - 10) / (60 - (-20)) = 19/8$

$S_Y = (130 - 30) / (60 - 20) = 5/2$

$$[x_v \quad y_v \quad 1] = [x_w \quad y_w \quad 1] \begin{bmatrix} 19/8 & 0 & 0 \\ 0 & 5/2 & 0 \\ 230/4 & -20 & 1 \end{bmatrix}$$

و عليه، نقطة التقاطع الأولى في منفذ المشاهدة هي (116.875,30) .

نقطة التقاطع الثانية في منفذ المشاهدة هي (200,61.818) .

.2

لايحدث تشويش لأنه يتم نقل الكائن بنفس نسبة الأبعاد و لكنه قد يصغر أو يكبر قليلاً ممايؤدي إلى ضياع بعض التفاصيل و لكن في كافة الأحوال يبقى الشكل العام ثابتاً.

3.(أ) شفرة النقطة (3,1) هي : 0100
شفرة النقطة (7,5) هي : 0010

(ب)

يمكن تمثيل النقاط على الخط بالمعادلات البارامترية التالية:

$X = 3 + 4t$

$y = 1 + 4t$

لإيجاد نقاط التقاطع نحل المعادلات البارامترية مع:

1\ $y = 2$

$2 = 1 + 4t$

$t = 1/4$

$x = 3 + 1 = 4$

نقطة التقاطع الأولى هي (4,2) .

2\ $x = 6$

$6 = 3 + 4t$

$t = 3/4$

$y = 1 + 3 = 4$

نقطة التقاطع الثانية هي (6,4) .

مسرد المصطلحات

القص: هو عملية تحويل هندسي بقطع كل الخطوط والمنحنيات التي تعبر حدود النافذة.

خوارزمية كوهين–سذرلاند: تمثل إجراءات بسيطة وذات كفاءة عالية لتصنيف مقاطع الخطوط بالنسبة لحدود النافذة.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

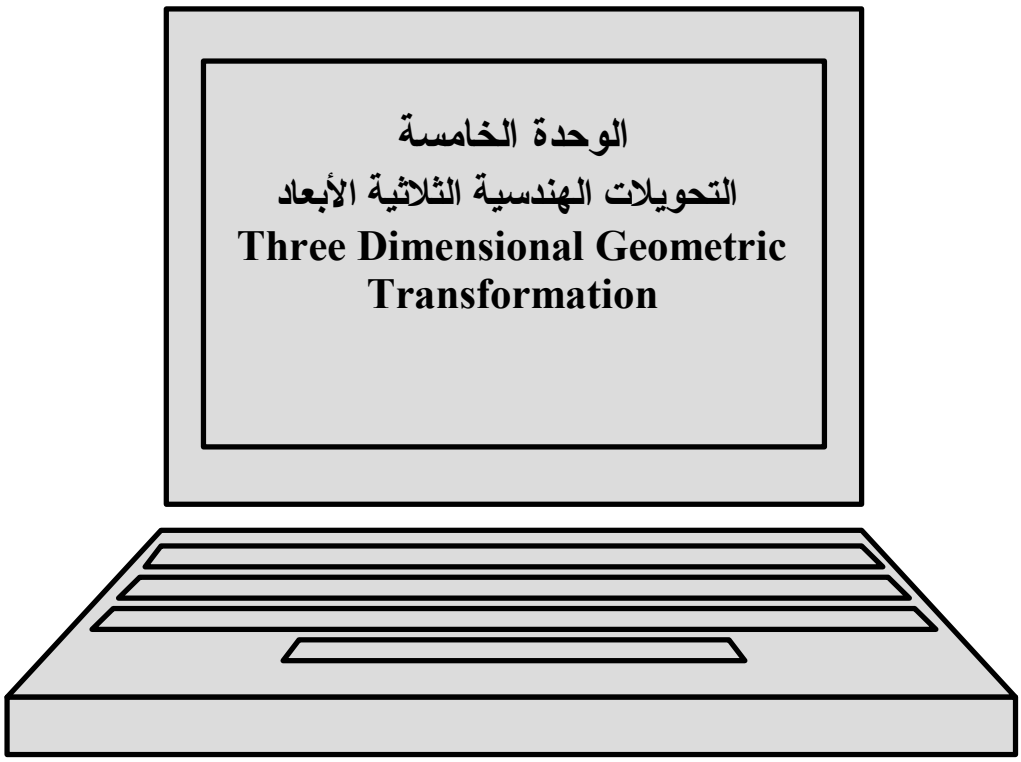
2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

6. Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة الخامسة
التحويلات الهندسية الثلاثية الأبعاد
Three Dimensional Geometric
Transformation

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**115**
تمهيد	115
أهداف الوحدة	115
1. نظام الإحداثيات	**116**
1.1 تمثيل النقاط	116
2.1 محاور الإحداثيات	117
2. الكمائن الثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)	**117**
1.2 مكونات البيانات التي تصف الوجه	118
2.2 تحويل نقاط المشهد	118
3.2 الإحداث الرابع	119
3.التحويلات الهندسية الثلاثية الإبعاد	**121**
4. ضبط الحجم scaling	**123**
5.الإزاحة	**127**
6. الدوران	**130**
1.6 مصفوفة الدوران	130
2.6 اشتقاق مصفوفة الدوران	132
3.6 مقلوبات المصفوفات	133
الخلاصة	138
لمحة مسبقة عن الوحدة الدراسية التالية	138
إجابات التدريبات	139
مسرد المصطلحات	143
المراجع	144

المقدمة

تمهيد

عزيزي الدارس!

مرحباً بك في الوحدة الخامسة وموضوعها التحويلات الهندسية الثلاثية الأبعاد، حيث نبدأ بنظام الإحداثيات، ثم الكمائن ثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)، ثم بعد ذلك نلج إلى التحويلات الهندسية الثلاثية الأبعاد، وننتقل بعد ذلك لضبط الحجم، الإزاحة، والدوران، وفي داخل هذه الوحدة عزيزي الدارس ستجد أمثلة لتعزيز الشرح وأسئلة للتقويم الذاتي وباقة من التدريبات لزيادة الفهم وتثبيت المعرفة.نرجو أن تستفيد منها وأن تشاركنا في نقدها وتقويمها.

أهداف الوحدة



بعد دراستك هذه الوحدة ينبغي أن تكون قادراً **عزيزي الدارس** على أن:

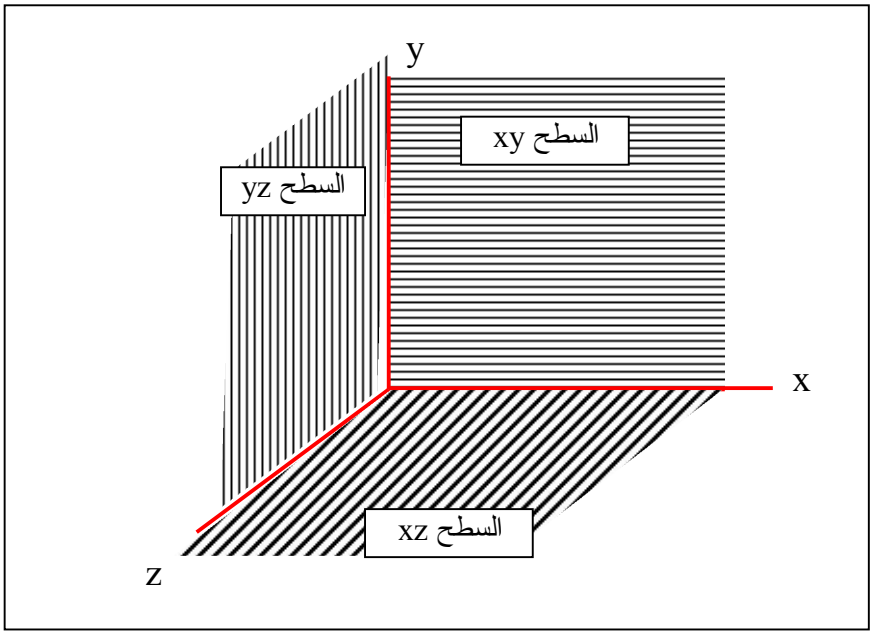
- **تستخدم** نظام الإحداثيات.
- **تلخص** الكمائن الثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)
- **تمثل** التحويلات الهندسية ثلاثية الأبعاد.
- **تضبط** حجم المنظر وإزاحته.
- **تتحقق** من شكل الكائن.

1. نظام الإحداثيات

الكائن الثلاثي الأبعاد **ينتج** عادة في نظام إحداثيات عالمي ثلاثي الأبعاد، ثم **ينقل** إلى العارض الذي يستخدم عادة نظام إحداثيات ثنائي الأبعاد. **إن تغير** الكائن الثلاثي الأبعاد وانتاجه ومشاهدته **يحتاج** إلى استخدام **الهندسة الثلاثية الأبعاد وتحويلات الإحداثيات**.

1.1 تمثيل النقاط

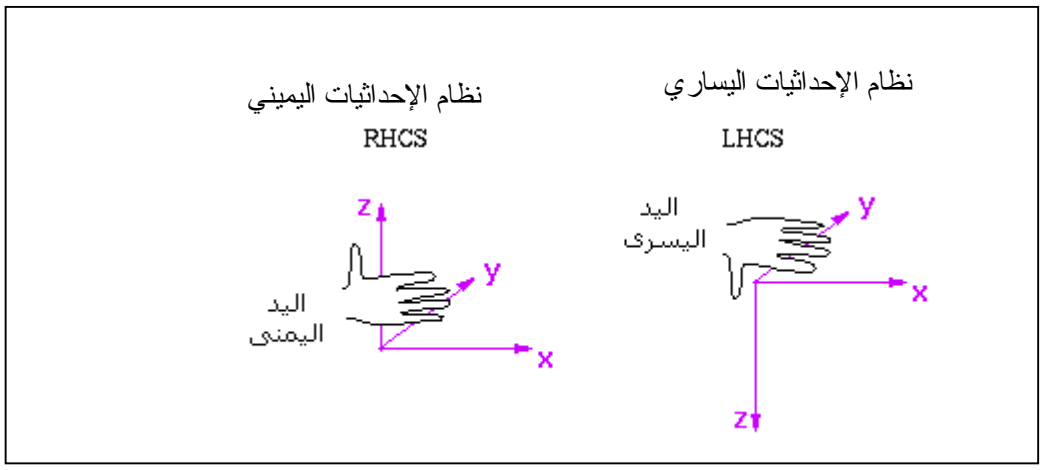
النقاط في الفضاء الثلاثي الأبعاد تمثل **بنظام إحداثيات ثلاثي الأبعاد** والذي يمكن اعتباره امتدادا **للنظام الثنائي الأبعاد** الذي تمت مناقشته في الوحدة السابقة، **بإضافة محور z متعامد مع السطح xy** مما **ينتج** سطحين أساسيين آخرين هما xz و yz كما هو موضح في الشكل 5.1.



الشكل 5.1: تمثيل ثلاث أسطح أساسية متعامدة

2.1 محاور الإحداثيات

توجه محاور الإحداثيات في النظام الثلاثي الأبعاد قد يكون **يمينياً** right-handed أو **يسارياً** left-handed، كما هو موضح بالشكل 5.2.



الشكل 5.2. توجه نظام الإحداثيات الثلاثي

2.الكائنات الثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)

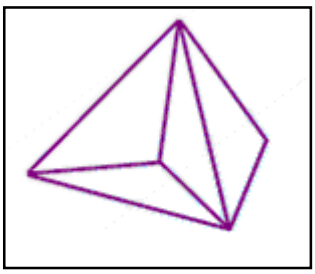
3-Dimensional Objects Bounded by Planar Surfaces (Facets)

تعرف الأوجه المستوية **بترتيب** مجموعة رؤوس ثلاثية الأبعاد على سطح واحد **تكون مضلعاً مقفولاً closed polygon** (ترسم خطوط مستقيمة من كل رأس إلى الرأس التالي مع ربط الرأس الأخير مع الأول).

في الأشكال الثلاثية الأبعاد **تتكون كل الأوجه من مضلعات مستوية planar polygons** (الشكل 5.3).

1.2 مكونات البيانات التي تصف الوجه

- **بيانات عددية numerical data** وتعطي قائمة نقاط الأبعاد الثلاثية وعددها N×3 إن كان عدد النقاط N.
- **بيانات طوبولوجيا topological data** وتحدد النقاط التي تربط مع بعضها لتكوين حافات الوجه.



الشكل 5.3: شكل ثلاثي الأبعاد

2.2 تحويل نقاط المشهد

إن استخدام الصيغ القانونية لكل من المسقط المنظور والمتعامد **يسهّل** من عملية الحساب، **ويتطلب** ذلك تحويل المشهد المحدد في إي نظام إحداثيات ثلاثي **بحيث** يكون اتجاه المنظر على **طول محور z** (في حالة المسقط المنظور) و**نقطة المنظر** في **تقاطع** الإحداثيات.

وبصورة عامة نحتاج لتحويل إحداثيات إي **نقطة** في المشهد **بحيث** تكون **نقطة منظر ما** مختارة $\mathbf{C} = [C_x, C_y, C_z]$ في تقاطع محاور الإحداثيات.

في كثير من الأحيان **نحتاج لتحويل نقاط المشهد لأسباب أخرى مثل**

- ❖ إنتاج تأثيرات خاصة في الصورة،
- ❖ دوران الكائن.

نتحصل على التحويلات من هذا النوع **بضرب** إي **نقطة** في المشهد **بمصفوفة تحويل**، بطريقة مماثلة لما تم مناقشته في الوحدة السابقة، وهناك **مشكلة** هنا، هي انه **لا**

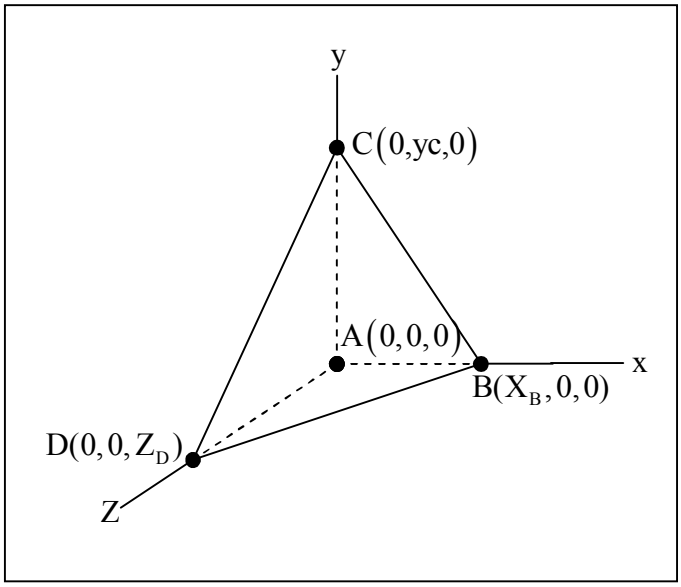
يمكننا الحصول على تحويل عام باستخدام إحداثيات ديكارتي Cartesian coordinates، ولهذا **السبب** نستخدم عادة نظاماً يعرف **بالإحداثيات المتجانسة *homogeneous coordinates***، كما ذكر فى الوحدة السابقة.

3.2 الإحداث الرابع

تمثل النقاط في الفضاء الثلاثي الأبعاد باستخدام ثلاثة إحداثيات، الشكل 5.4 يوضح **تمثيل الهرم المثلثى tetrahedron** ، **وفى** نظام الإحداثيات المتجانس **يكون للنقاط الثلاثية الأبعاد إحداث رأسي رابع** كالاتى:

$$\mathbf{P} = [p_x, p_y, p_z, s]$$

الإحداث الرابع هو عامل تغير الحجم (تقيس) scale factor. إن **تحويل** الإحداثيات **المتجانسة** إلى الإحداثيات **الديكارتية** يتم **بقسمها** في إحداثيات **رأسية** أخرى، فالإحداثيات الديكارتية **المكافئة** ل $[p_x, p_y, p_z, s]$ **هي** $[p_x/s, p_y/s, p_z/s]$. **وفي** أغلب الأحيان تكون **قيمة** العامل s **تساوى 1**. إن **أهمية** استخدام الإحداثيات المتجانسة كما ذكر في الوحدة السابقة، تتمثل في أنها **تسمح** لنا باستخدام **إزاحة** أي نقطة في المشهد **باستخدام** ضرب المصفوفات.



الشكل 5.4: تمثيل الهرم المثلثي tetrahedron

في نظام الإحداثيات المتجانسة **يمثل** الهرم المثلثي **باستخدام** مصـــفوفة ترتيبـــه 4×4 كالآتي:

$$[\mathrm{P}] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ x_B & 0 & 0 & 1 \\ 0 & y_C & 0 & 1 \\ 0 & 0 & z_D & 1 \end{bmatrix} \qquad 5.1$$

أسئلة تقويم ذاتي

1. مم ينتج الكائن الثلاثي الأبعاد؟
2. عرف الأوجه المستوية
3. ما هو الإحداث الرابع؟
4. كيف يتم تمثيل النقاط في الفضاء الثلاثي الأبعاد؟
5. ما هي أوجه الحاجة لتحويل إحداثيات إي نقطة في مشهد ما؟
6. لماذا نستخدم في التحويلات نظام الإحداثيات المتجانسة بدلا عن نظام الإحداثيات الديكارتي؟

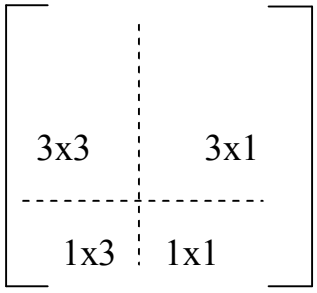
3. التحويلات الهندسية الثلاثية الأبعاد

تنفذ التحويلات في الفضاء الثلاثي الأبعاد باستخدام نفس الأسلوب الذي استخدم تنفيذ التحويلات في المجال الثنائي الأبعاد مع إضافة إحداث–z، وفي نظام الإحداثيات المتجانس **تمثل** التحويلات في الفضاء الثلاثي الأبعاد

❖ **باستخدام مصفوفة من النوع:**

$$\begin{bmatrix} A & B & C & 0 \\ D & E & F & 1 \\ G & H & I & 0 \\ J & K & L & S \end{bmatrix} \qquad 5.2$$

يمكن تقسيم هذه المصفوفة كالآتي:



❖ **المصفوفات الفرعية:**

المصفوفة الفرعية التي ترتيبها 3×3 في اليسار الأعلى تسمح

- ♦ بضبط الحجم scaling
- ♦ الانعكاس reflection
- ♦ القص shearing
- ♦ الدوران rotation.

❖ **المصفوفة الفرعية بالترتيب 3×1 في اليسار الأدنى تنتج**

- ♦ الإزاحة

❖ **المصفوفة الفرعية بالترتيب 1×1 في اليمين الأسفل تنتج**

- ♦ ضبط حجم شامل منتظم "uniform global scalin".

❖ **المصفوفة بالترتيب 3×1 في أعلى اليمين يمثل وجودها**

- ♦ جزء من تمثيل الإحداثيات المتجانسة.

4.ضبط الحجم Scaling

النقطة P(x,y,z,1) **يعاد ضبطها** إلى النقطة P'(x',y',z',1) في **نظام الإحداثيات المتجانسة** بالتحويل التالي:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} A & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.3$$

يسمى هذا عادة **إعادة الضبط بمرجعية نقطه تقاطع محـــاور الإحـــداثيات. وإذا كانت** العوامل A و E و I **مختلفة** فإنه يكون منظر الكائن **مشوشا**، وإن **تســـاوت** هـــذه العوامل **فسوف يتغير الحجم**، و**هذا النوع** من ضبط الحجم الكلي يمكن أيضا أن **ينـــتج من التحويل** التالي:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix}$$

$$= [x \quad y \quad z \quad s] \qquad 5.4$$

لإيجاد الإحداثيات الديكارتية تعدل المعادلة 5.4 على النحو التالي:

$$[x \quad y \quad z \quad s] = \left[\frac{x}{s} \quad \frac{y}{s} \quad \frac{z}{s} \quad 1\right]$$

لقيمة s أكبر من واحد **ينقص** حجم المنظر. **ولتكبير** المنظر **العامل 1/s** يجـــب أن **يستخدم في مصفوفة ضبط الحجم**، وتصبح الإحداثيات **النهائية** كالآتي:

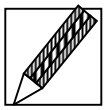
$$[sx \quad sy \quad sz \quad 1]$$

أسئلة تقويم ذاتي



1. في ماذا تستخدم المصفوفة الفرعية التي ترتيبها 3×3 في اليسار الأعلى؟
2. عند إعادة الضبط بمرجعية نقطه تقاطع محاور الإحداثيات متى يكون منظر الكائن مشوّشا؟
3. لماذا يستخدم العامل 1/s في مصفوفة ضبط الحجم؟

تدريب (1)



أملأ الفراغات في الجدول التالي:

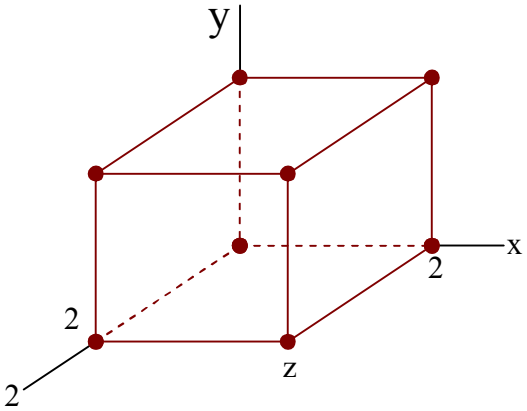
ترتيب المصفوفة الفرعية	الموقع	الخاصية
3*3		
		الإزاحة
		ضبط حجم شامل
		منتظم
	أعلى اليمين	

مثال:

تخيل مكعباً طول كل من أضلاعه 2 سم كما هوة موضح في الشكل أدناه وتمثل الإحداثيات المتجانسة للمكعب بهيئة المصفوفة التالية:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 0 & 2 & 2 & 1 \end{bmatrix}$$

استخدم مصفوفة ضبط الحجم لتصغير حجم المكعب لمكعب وحدة.



الحل:

مصفوفة ضبط الحجم المطلوبة لتحويل مكعب 2 سم لمكعب وحـــدة هي:

$$[T_{SC}]=\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{a}$$

أو

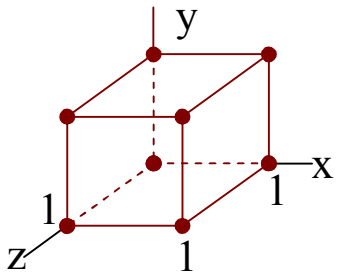
$$[T_{SC}]=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \qquad \text{b}$$

يمكن استخدام أي من المصفوفتين وباستخدام المصفوفة b نتحصل على:

$$[P']_{CUBE}^{UNIT} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 0 & 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 \\ 2 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 0 & 2 & 2 & 2 \end{bmatrix}$$

في الهيئة المعيارية

$$[P']_{CUBE}^{UNIT} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$



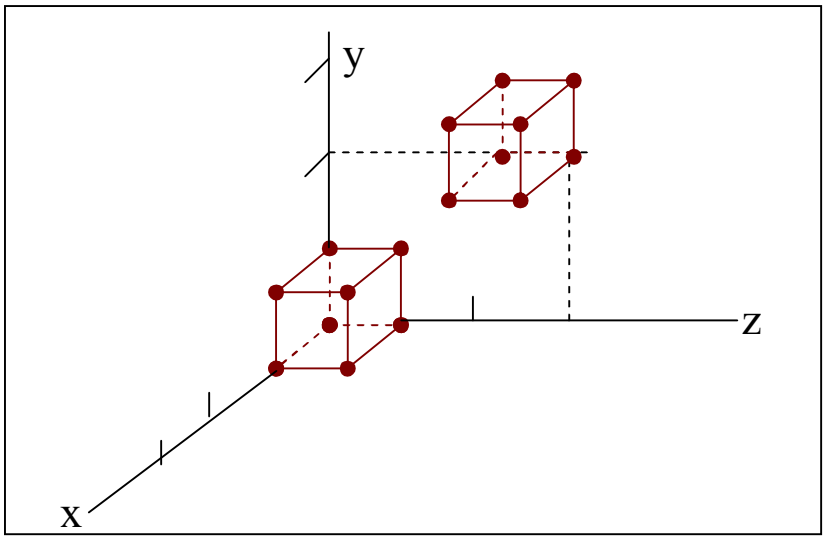
5.الإزاحة

مصفوفة التحويل التالية تحرك (تزيح) النقطة (x,y,z) إلى النقطة (x',y',z') عبر (J,K,L):

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ J & K & L & 1 \end{bmatrix} \qquad 5.5$$

تمثل قيم كل من J و K و L **الإزاحة النسبية** للنقطة في اتجاهات x و y و على التوالي.

الشكل 5.5 يوضح إزاحة مكعب وحدة لديه أحد الرؤوس عند تقاطع محاور الإحداثيات لموقع جديد في الفضاء يحدد بمصفوفة الإزاحة.

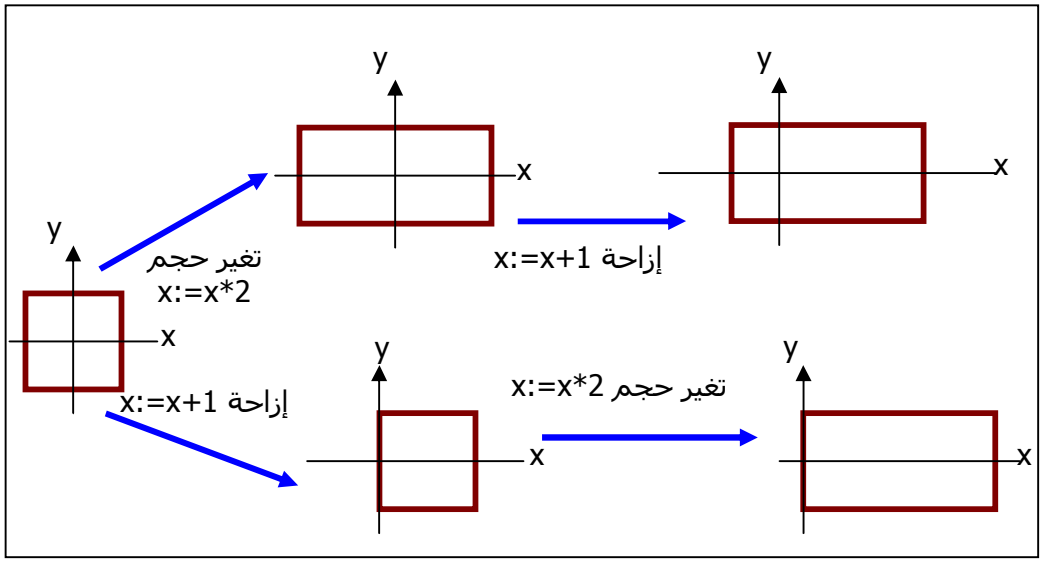


الشكل 5.5. إزاحة مكعب وحدة

أي رأس من رؤوس هذا المكعب يزاح بنفس القدر، عليه في هيئة المصفوفة تصبح إحداثيات المكعب المزاح

$$[P']_{CUBE}^{UNIT} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 3 & 3 & 1 & 1 \\ 3 & 3 & 0 & 1 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

ويلاحظ هنا أن تعرض الكائن إلى **أكثر من تحويل مثل ضبط الحجم والإزاحة**، فمن **الضروري** القيام بهذه التحويلات **بالترتيب الصحيح** ولا يمكن **عكسها**. والشكل 5.6 يوضح هذه الحالة باستخدام صورة بسيطة.



الشكل 5.6. أهمية ترتيب التحويلات

مثال:

تخيل كائناً عند نقطة تقاطع محاور الإحداثيات،
ثم قم بتكبير حجم الكائن إلى ضعف حجمه الأصلي ثم حرّكــه إلى
النقطة [5,5,20].

الحل:

ترتيب التحويل: تغيير الحجم المتبع بإزاحة، يستخدم المصفوفات التالية

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 5 & 5 & 20 & 1 \end{bmatrix}$$

نضرب أولا مصفوفة الإزاحة لإزاحة النقاط

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 5 & 5 & 20 & 1 \end{bmatrix}$$

6. الدوران

الدوران في الأبعاد الثلاثة **مهم** لإعطاء صورة **شاملة** لمعرفة شكل الكائن و**التحقق** منه بالنظر من **زوايا مختلفة**. إن **الدوران** في الأبعاد **الثلاثة** أكثر **تعقيدا** مقارنة بالدوران في المجال الثنائي الأبعاد **نسبة** لأنه **يجب تحديد محور للدوران بدلا عن نقطة**.

يتم التعامل مع الدوران بصورة مختلفة، حيث إنه من الضروري **تحديد محور الدوران**. ومن الضروري الاهتمام **باتجاه الدوران** أيضاً والمعادلات أعلاه تتبع عرف **نظام محاور اليد اليسرى**، مما يعنى أن محور y الموجب يكون رأسيا، ومحور x الموجب على اليمين و محور z الموجب إلى داخل الصفحة، وفي هذه الحالة يكون الدوران مع اتجاه دوران عقارب الساعة عندما ينظر إليه من ناحية الموجبة للمحاور والعكس صحيح بأن الدوران يكون عكس دوران عقارب الساعة إن نظر إليه من الناحية السالبة للمحاور.

1.6 مصفوفة الدوران

بتمديد المصفوفة التي استخدمت للدوران في المجال الثنائي الأبعاد، يمكن كتابة مصفوفة الدورات بزاوية θ حول محور−z عكس عقارب الساعة كالآتي:

$$[T_R]_z^{\theta} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.6$$

تنتج هذه المصفوفة الانتقال التالي

$$\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned} \qquad 5.7$$

إن كان الدوران مع عقارب الساعة تصبح مصفوفة الدوران حول المحور−z

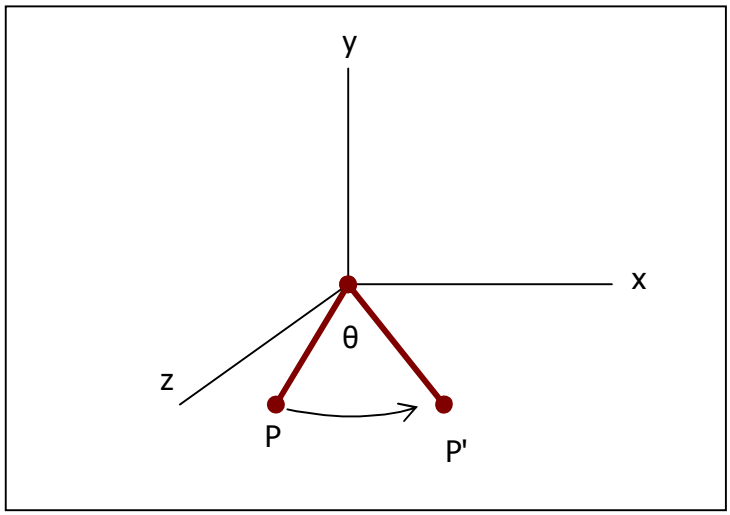
$$[T_R]_z^\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.8$$

بطريقة مماثلة يمكن إيجاد الدوران بزاوية θ حول محور−y عكس عقـــارب الساعة، باعتبار الشكل 5.7

$$[T_R]_y^\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.9$$

تنتج هذه المصفوفة الانتقال التالي

$$\begin{aligned} x' &= x\cos\theta + z\sin\theta \\ y' &= y \\ z' &= -x\sin\theta + z\cos\theta \end{aligned} \qquad 5.10$$



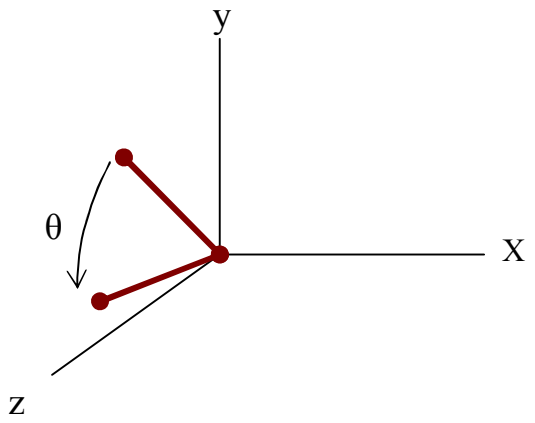
الشكل 5.7. الدوران حول محور−y

إن كان الدوران مع عقارب الساعة فتصبح مصفوفة الدوران حول المحور−y

$$[T_R]_y^{\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.11$$

2.6 اشتقاق مصفوفة الدوران

يمكن اشتقاق مصفوفة الدوران حول المحور−x عكس عقارب الساعة من الشكل 5.8



P P'

الشكل 5.8. الدوران حول محور−x

$$[T_R]_x^{\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.12$$

تنتج هذه المصفوفة الانتقال التالي

$$\begin{aligned} x' &= x \\ y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \end{aligned} \qquad 5.13$$

إن كان الدوران مع عقارب الساعة تصبح مصفوفة الدوران حول المحور−x

$$[T_R]_x^\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 5.14$$

3.6 مقلوبات المصفوفات

مقلوبات inverse هذه المصفوفات يمكن أن يتم من غير استخدام **طريقة حذف غاوسينGaussean elimination** وذلك **باعتبار** معنى أي **تحويل**. لمصـــفوفة **تغيـــر الحجم نستبدل القيم** s_x و s_y و s_z **بالقيم** $1/s_x$ و $1/s_y$ و $1/s_z$ على التوالي.

ولمصفوفة الإزاحة نستبدل القيم t_x و t_y و t_z **بـــالقيم** $-t_x$ و $-t_y$ و $-t_z$ علـــى التوالي. **لمصفوفة الدوران يلاحظ** التالي:

$$\cos(-\theta) = \cos(\theta) \text{ and } \sin(-\theta) = -\sin(\theta)$$

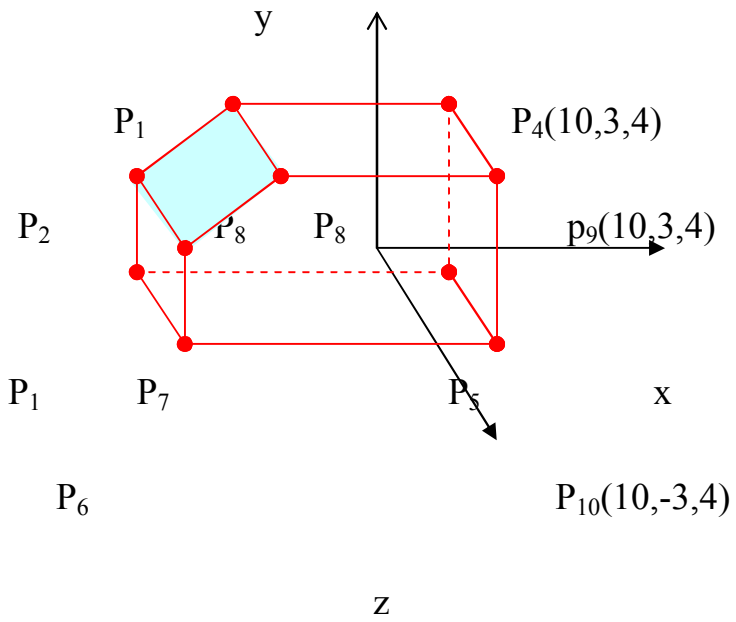
وعليه نتحصل على مقلوب المصفوفة بتغير إشارة حدود جيـــب الزاويـــة sin terms.

مثال:

خذ القالب المشطوف الموضح في الشكل 5.8. النقاط P_1 و P_2 تحدد شـــكل (هندســـة) القالب ويمكن كتابتها في هيئة المصفوفة التالية:

$$[P] = \begin{bmatrix} -10 & -3 & -4 & 1 \\ -10 & 1 & -4 & 1 \\ -8.5 & 3 & -4 & 1 \\ 10 & 3 & -4 & 1 \\ 10 & -3 & -4 & 1 \\ -10 & -3 & 4 & 1 \\ -10 & 1 & 4 & 1 \\ -8.5 & 3 & 4 & 1 \\ 10 & 3 & 4 & 1 \\ 10 & -3 & 4 & 1 \end{bmatrix}$$

دور (لف) القالب بحيث يكون السطح الأسفل المحدد بالنقاط P_1 و P_5 و P_{10} و P_6 موازياً للسطح xy.



الحل:

لجعل السطح المحدد بالنقاط P_1 و P_5 و P_{10}و P_6 موازياً للسطح xy، **يتطلب الـــدوران 90^0 عكس عقارب الساعة حول المحور-x** المصفوفة المناسبة تعطى بالأتي:

$$[T_R]_x^{\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

باستخدام $\theta = 90^0$ تصبح المصفوفة

$$[T_R]_x^{90^o} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

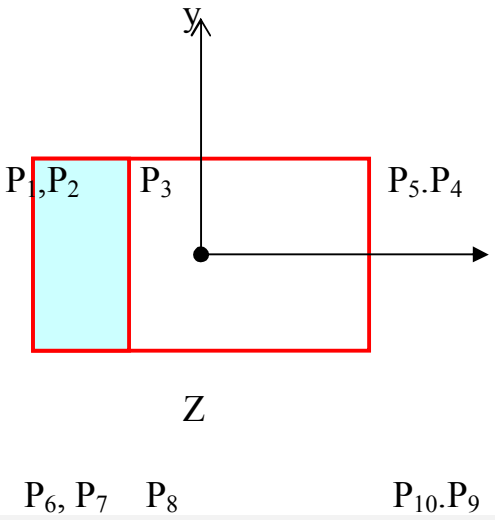
مصوفة النقاط المحولة تعطي بالآتي:

الشكل أدناه يوضح منظر الكائن بعد الدوران 90^0 عكس عقارب الساعة حول محور –× كما

$$[P'] = [P][T_R]_x^{90^\circ}$$

$$= \begin{bmatrix} -10 & -3 & -4 & 1 \\ -10 & 1 & -4 & 1 \\ -8.5 & 3 & -4 & 1 \\ 10 & 3 & -4 & 1 \\ 10 & -3 & -4 & 1 \\ -10 & -3 & 4 & 1 \\ -10 & 1 & 4 & 1 \\ -8.5 & 3 & 4 & 1 \\ 10 & 3 & 4 & 1 \\ 10 & -3 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -10 & 4 & -3 & 1 \\ -10 & 4 & 1 & 1 \\ -8.5 & 4 & 3 & 1 \\ 10 & 3 & -4 & 1 \\ 10 & 4 & -3 & 1 \\ -10 & -4 & -3 & 1 \\ -10 & -4 & 1 & 1 \\ -8.5 & -4 & 3 & 1 \\ 10 & -4 & 3 & 1 \\ 10 & -4 & -3 & 1 \end{bmatrix}$$

يشاهد على محور –z.



يتم **التعامل مع الدوران بصورة مختلفة**، حيث إنه **من الضروري** تحديد **محور** الدوران و **من الضروري** أيضاً الاهتمام **باتجاه** الدوران. والمعادلات أعلاه تتبع **عرف** نظام **محاور اليد اليسرى**، مما يعنى أن **محور y الموجب يكون رأسيا، ومحور x الموجب على اليمين** و **محور z الموجب إلى داخل الصفحة**، و في هذه الحالة يكون **الدوران** مع **اتجاه دوران عقارب الساعة** عندما ينظر إليه من الناحية **الموجبة**

للمحاور **والعكس صحيح** بأن الدوران يكون **عكس دوران عقارب الســاعة** إن نظــر إليه من **الناحية السالبة للمحاور.**

أسئلة تقويم ذاتي

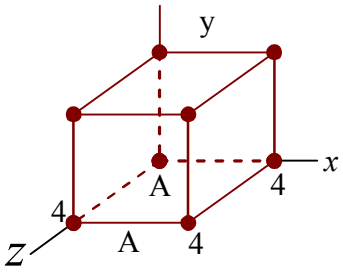


1. الدوران في الأبعاد الثلاثة أكثر تعقيدا مقارنة بالدوران في المجال الثنائي الأبعاد، وضح.
2. اكتب مصفوفة الدوران بزاوية θ حول محور $-z$ عكس عقارب الساعة
3. كيف يمكن إيجاد مقلوبات مصفوفات الدوران من غير استخدام طريقة حذف غاوسين Gaussean؟

تدريب (1)

1. لف الخط المستقيم الذي يصل النقاط A(1,1,1) و B(4,3,2) 30^0 حول محور–x و 60^0 وحول محور–y في الحالتين الدوران عكس عقارب الساعة. أعد العملية في الترتيب العكسي أي الدوران حول محور–y أولا ثم الدوران حول محور–x، قارن الإحداثيات النهائية في الحالتين.
2. خذ المكعب الموضح في الشكل أدناه مع طول الضلع يساوى 4 سم.
   أ. انقص حجم المكعب إلى النصف (طول الضلع يصبح 2 سم)
   ب. أزح المكعب المتحصل عليه لتكون النقطة A عند (1,1,1)
   ج. أدر المكعب الموضح بزاوية 45^0 عكس اتجاه عقارب الساعة حول محور–z،

   ارسم منظر المكعب المتحصل عليه في كل حالة.



الخلاصة

أخي الدارس خلصنا إلى أنه يمكن تحويل النظام الثنائي الأبعاد إلى نظام ثلاثي الأبعاد بإضافة محور z متعامد مع السطح xy مما ينتج سطحيين أساسيين آخريين هما xz و yz . ويمكن توجيه محاور الإحداثيات إلى اليمين أو اليسار ومن ثم يتم ترتيب مجموعة الرؤوس ثلاثية الأبعاد على سطح واحد لتكون مضلعاً مقفولاً يعرف بالأوجه المستوية.

إن تحويل نقاط المشهد يمكّن من مشاهدة المنظر من زوايا مختلفة مما يدعم إنتاج تأثيرات خاصة بالصورة كما يدعم دوران الكائن في الاتجاهات المختلفة لتكوين نقطة منظر مختارة.

يمكن تمثيل التحويلات الهندسية الثلاثية الإبعاد في نظام الإحداثيات المتجانس باستخدام المصفوفات التي يمكن تقسيمها إلى أربع مصفوفات فرعية تدعم كل منها تحويلات معينة، مثل الإزاحة وضبط الحجم. أما عملية الدوران فلا بد فيها من تحديد محور الدوران واتجاه الدوران.

لمحة مسبقة عن الوحدة التالية

سوف نشرح في هذه الوحدة التالية التي تحمل عنوان الإسقاط والقص، إسقاطات، نماذج الإطار السلكي ثم الإسقاطات المتوازية، والإسقاط المتعامد المتعدد المشاهد، والإسقاطات المنظورة، وتحويلات المشاهدة، وأحجام المشهد، ومعادلات السطح وأخيراً نختم شرحنا بعملية القص، ونأمل أن تكسبك المزيد من المعرفة والمهارات في تخطيطات الحاسوب.

إجابات التدريبات

تدريب(1)

ترتيب المصفوفة الفرعية	الموقع	الخاصية
3*3	**اليسار الأعلى**	ضبط الحجم ، الانعكاس، القص، الدوران
1×3	**اليسار الأدنى**	الإزاحة
1×1	**اليمين الأسفل**	ضبط حجم شامل منتظم
1×3	أعلى اليمين	جزء من تمثيل الإحداثيات المتجانسة.

تدريب(2)

1. لف الخط المستقيم الذي يصل النقاط A(1,1,1) و B(4,3,2) 30^0 حول محور-x و 60^0 حول محور-y في الحالتين الدوران عكس عقارب الساعة. اعد العملية في الترتيب العكسي اى الدوران حول محور-y أولا ثم الدوران حول محور-x، قارن الإحداثيات النهائية في الحالتين.

المصفوفة الخاصة بالدوران 30 درجة حول محور x هي:

$$[T_R]_x^\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

المصفوفة الخاصة بالدوران 60 درجة حول محور y هي:

$$[T_R]_y^\varphi = \begin{bmatrix} 0.5 & 0 & -0.866 & 0 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

المصفوفة التي تمثل الخط المستقيم هى:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

في حالة لف المستقيم أولاً حول محور x ثم حول محور y تكون المصفوفة الناتجة هي:

$$\begin{bmatrix} 1.683 & 0.366 & -0.183 & 1 \\ 4.799 & 1.598 & -1.848 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & -0.866 & 0 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

في حالة لف المستقيم أولاً حول محور y ثم حول محور x تكون المصفوفة الناتجة هي:

$$\begin{bmatrix} 1.366 & 1.049 & 0.183 & 1 \\ 3.732 & 3.830 & -0.634 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & -0.866 & 0 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

من النتيجة أعلاه يمكن الإستنتاج أنه في التحويلات الثلاثية الأبعاد يؤثر ترتيب التحويلات على الشكل النهائي للعنصر المراد إجراء التحويلات عليه و لذا يجب أخذ الترتيب في الإعتبار.

2. اعتبر المكعب الموضح في الشكل أدناه مع طول الضلع يساوى 4 سم.
المصفوفة التي تمثل رؤوس المكعب هى:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 0 & 4 & 0 & 1 \\ 0 & 0 & 4 & 1 \\ 4 & 4 & 0 & 1 \\ 4 & 0 & 4 & 1 \\ 0 & 4 & 4 & 1 \\ 4 & 4 & 4 & 1 \end{bmatrix}$$

أ. انقص حجم المكعب إلى النصف (طول الضلع يصبح 2 سم)

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 0 & 4 & 0 & 1 \\ 0 & 0 & 4 & 1 \\ 4 & 4 & 0 & 1 \\ 4 & 0 & 4 & 1 \\ 0 & 4 & 4 & 1 \\ 4 & 4 & 4 & 1 \end{bmatrix}$$

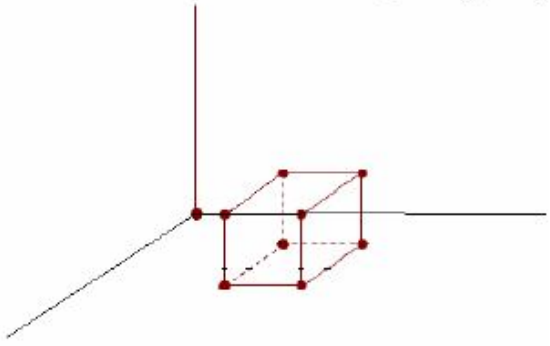
و الشكل كالتالي:



ب. أزح المكعب المتحصل عليه لنكون النقطة A عند (1,1,1)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 3 & 3 & 1 & 1 \\ 3 & 1 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 3 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix}$$

و الشكل كالتالي:



مسرد المصطلحات

الكائنات الثلاثية الأبعاد المحدودة بأسطح مستوية (الأوجه)

3-Dimensional Objects Bounded by Planar Surfaces (Facets)

تعرف الأوجه المستوية **بترتيب** مجموعة رؤوس ثلاثية الأبعاد على سطح واحد **تكـــون مضلعاً مقفولاً closed polygon**

بيانات عددية numerical data وتعطي قائمة نقاط الأبعاد الثلاثية وعددها N×3 إن كان عدد النقاط N.

بيانات طوبولوجيا topological data وتحدد النقاط التي تربط مع بعضـــها لتكـــوين حافات الوجه.

تحويل نقاط المشهد: تحويل إحداثيات إي نقطة في المشهد بحيث تكون نقطة منظر ما. الإحداث الرابع هو عامل تغير الحجم (تقيس) scale factor.

التحويلات الهندسية الثلاثية الإبعاد:تنفذ التحويلات في الفضاء الثلاثي الأبعاد باستخدام نفس الأسلوب الذي استخدم تنفيذ التحويلات في المجال الثنائي الأبعاد مع إضافة إحداث–z.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة السادسة
الإسقاط والقص
Projection and Clipping

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**147**
تمهيد	147
أهداف الوحدة	147
1. إسقاطات نماذج الإطار السلكي Projections of Wire-Frame Models	**148**
2. تحويلات المشاهدة Viewing Transformations	**158**
3. أحجام المشهد View Volumes	**159**
4. معادلات السطح Plane Equations	**162**
5. القص Clipping	**163**
الخلاصة	**167**
لمحة مسبقة عن الوحدة الدراسية التالية	**167**
إجابات التدريبات	**168**
مسرد المصطلحات	**170**

المقدمة

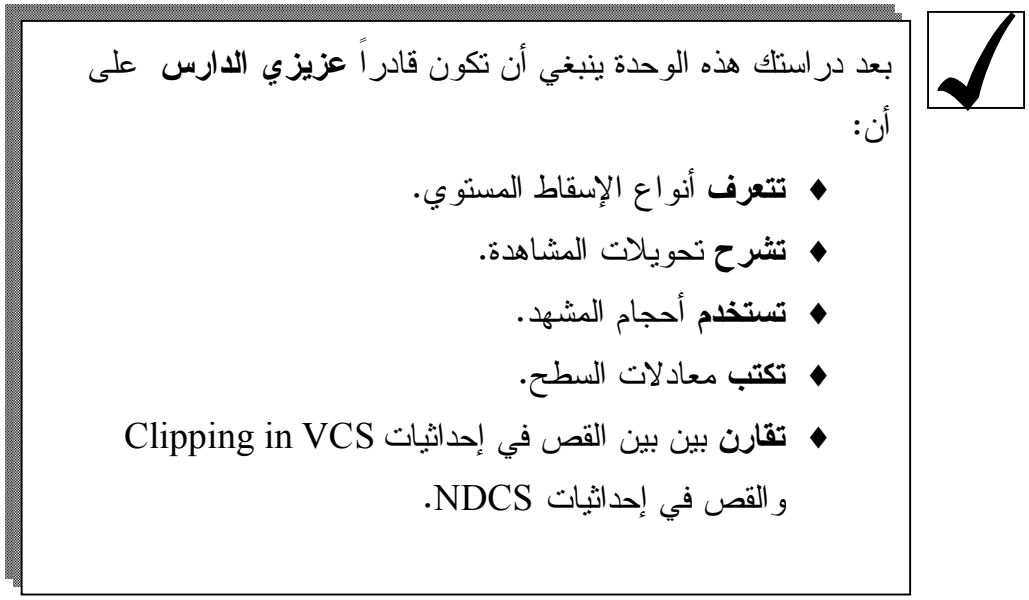
تمهيد

مرحباً بك في هذه الوحدة السادسة والتي عن الإسقاط والقص وسنتناول فيها شرح إسقاطات نماذج الإطار السلكي

ثم نتناول الإسقاطات المتوازية، والإسقاط المتعامد المتعدد المشاهد ،الإسقاطات المنظورة، وتحويلات المشاهدة، وأحجام المشهد، ومعادلات السطح وأخيراً نختم شرحنا بعملية القص.

عزيزي الدارس في داخل هذه الوحدة ستجد أسئلة تقويم وتدريبات وستجد إجابتها في نهاية الوحدة حتى تتمكن من تثبيت معرفتك بالإسقاط والقص.

أهداف الوحدة



بعد دراستك هذه الوحدة ينبغي أن تكون قادراً **عزيزي الدارس** على أن:

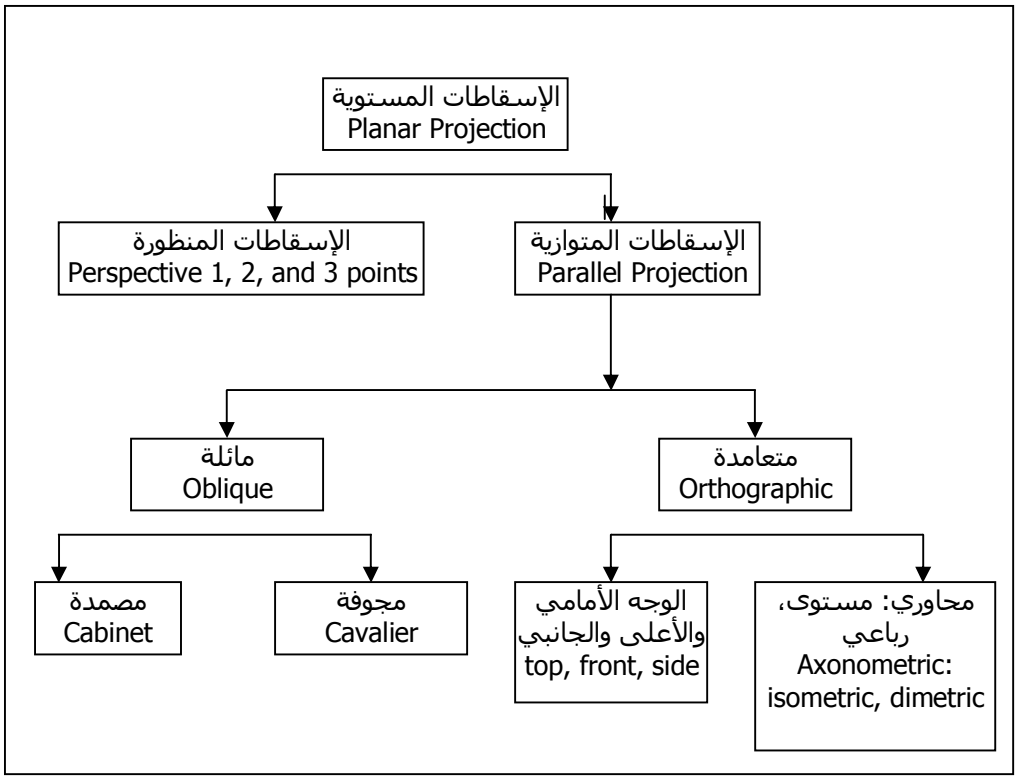
- **تتعرف** أنواع الإسقاط المستوي.
- **تشرح** تحويلات المشاهدة.
- **تستخدم** أحجام المشهد.
- **تكتب** معادلات السطح.
- **تقارن** بين بين القص في إحداثيات Clipping in VCS والقص في إحداثيات NDCS.

1. إسقاطات نماذج الإطار السلكي

Projections of Wire-Frame Models

- **يعرف الوضع** (التخطيط) Mapping: $f: R^n \rightarrow R^m$.
- **الإسقاط** projection حالة خاصة من التخطيط حيث تكون m اقل من n.
- و**يستخدم الإسقاط السطحي** planar Projection لوضع الأشكال الثلاثية الأبعاد على سطح.

توجد أنواع مختلفة للإسقاط المستوية، كما هو موضح بالشكل 6.1



الشكل 6.1: أنواع الإسقاطات المستوية

الإسقاط projection

جهاز العرض في نظم التخطيط الحاسوبية **ثنائي الأبعاد** ويتطلب منا هذا **تعريف تحويل** من الفضاء الثلاثي الأبعاد إلى سطح جهاز العرض الثنائي الأبعاد، و**هذا التحويل يعرف بالإسقاط projection**، ويتم **مشاهدة** الكائن الثلاثي الأبعاد بالتحويل من الفضاء الثلاثي الأبعاد إلى السطح الثنائي الأبعاد، ثم **تستخدم بدائيات** primitives الرسم البسيطة **لرسم الكائن.**

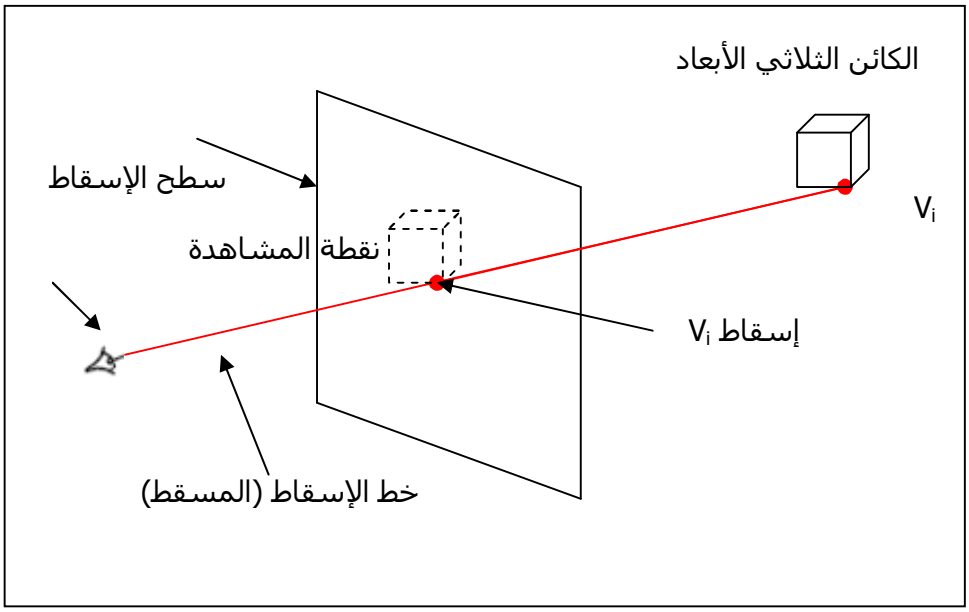
لإسقاط كائن ثلاثي الأبعاد على سطح ثنائي الأبعاد يتطلب:

- **اختيار** سطح الإسقاط projection surface أولا.
- **تحديد** المساقط projectors أو الخطوط التي تمر عبر كل رأس للكائن.

الإسقاط الهندسي المستوى planar geometric projections:

توضع رؤوس الإسقاط projection vertices عند **تقاطع** المسقط مع سطح الإسقاط. **وأبسط وأكثر طرق** الإسقاط استخداما **لمشاهدة** للأشكال الثلاثية الأبعاد هو **استخدام أسطح إسقاط مستوية وخطوط مستقيمة للمساقط**، **ويسمى** هذا **بالإسقاط الهندسي المستوى planar geometric projections**، الشكل 6.2. **ينشا اتجاه المشاهدة** من المشاهد إلى الكائن بواسطة خطوط ساقطة (مساقط) تمر عبر السطح (سطح الإسقاط) الذي سيظهر عليه الساقط.

عموما يمكن الإسقاط على أسطح مختلفة مثل **السطح الكروي والقمع** الخ، كما يمكن **استخدام المساقط المنحنية لإنتاج التأثير على العدسة**، وسنتعرض في هذا المقرر للإسقاط المستوى الخطي فقط والذي يمثل أبسط نسق للنظر للكائن حيث ترسم كل الحافات المسقطة.



الشكل 6.2: الإسقاط المستوي

أنواع الإسقاطات الهندسية المستوية:

يوجد نوعان من الإسقاطات الهندسية المستوية:

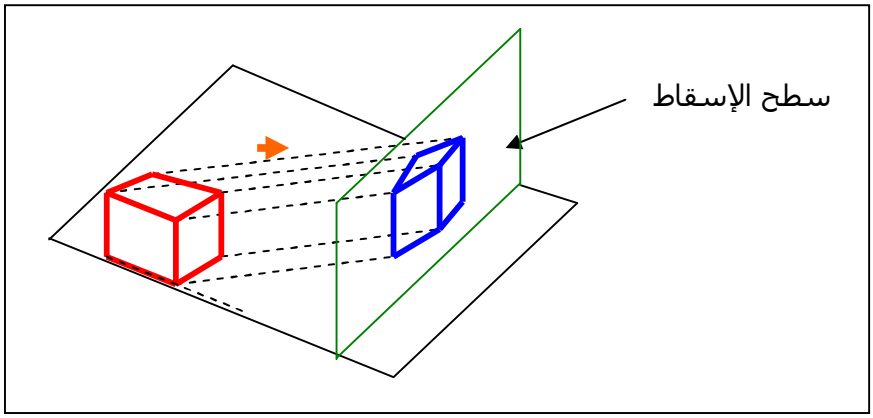
- الإسقاطات المتوازية parallel projection.
- الإسقاط المنظوري perspective projection.

❖ **الإسقاطات المتوازية Parallel Projection:**

تستخدم هنا مساقط **متوازية** و يكون **مركز الإســقاط** centre of projection **فـــي اللاتناهي** infinity وعليه تصبح المساقط موازية لبعضها كما ذكر أعلاه.

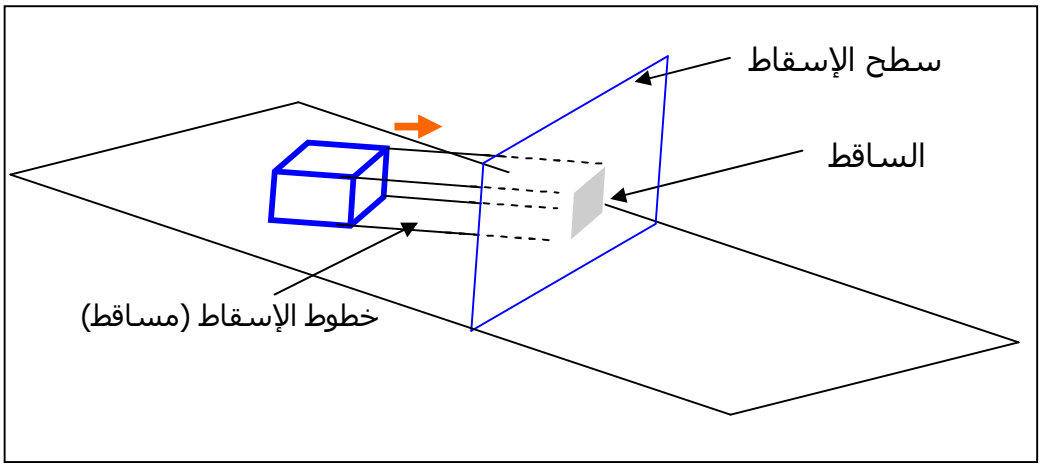
يوجد نوعان من المساقط يمكن استخدمهما في الإسقاطات المستوية هما

- **المساقط المائلة oblique بالنسبة لسطح الإسقاط**،وهنا يجعل أحد أسطح الكائن موازياً لسطح الإسقاط، كما هو موضح بالشكل 6.3.



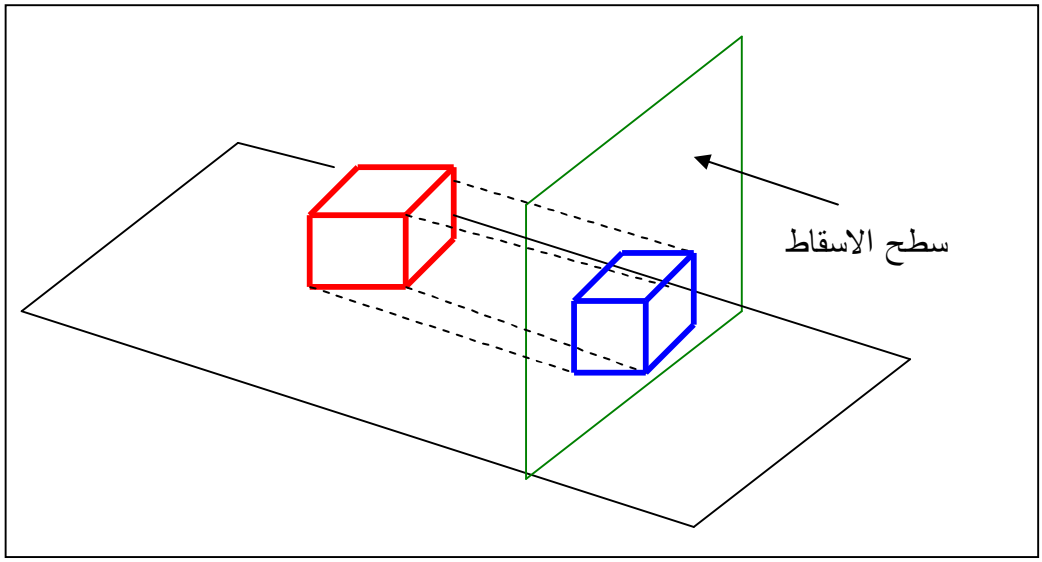
الشكل 6. 3: مثال للمساقط المائلة

- **المساقط متعامدة orthographic**، حيث تكون المساقط متعامدة مع سطح الإسقاط والذي قد يكون:
  - **متعامداً متعدد المشاهد multivew orthographic**، حيث تبقى أسطح الكائن موازية لسطوح الإسقاط الرئيسة، كما هو موضح في الشكل 6.4،



الشكل 6.4: مثال لسطح الإسقاط

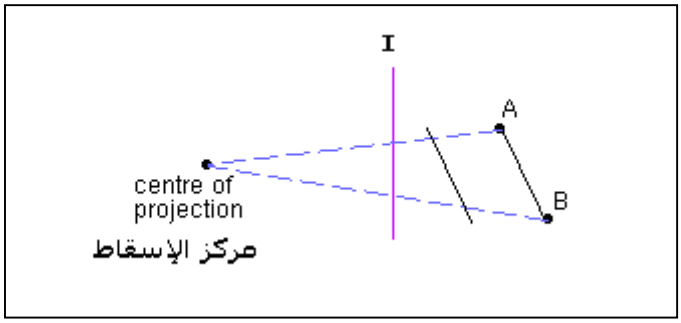
- **إسقاط محاوري axonometric**، ويكون فيه سطح الكائن مائلا بالنسبة لسطح الإسقاط، كما هو موضح الشكل 6.5.



الشكل 6.5: الإسقاط المحاورى

❖ **الإسقاطات المنظورة perspective projections:**

في الإسقاط المنظورى يقع **مركز الإسقاط** على **مسافة محددة** من سطح الإسقاط و**تمر** كل خطوط الإسقاط (المساقط) **عبر هذه النقطة** والتي تدعى **نقطة المنظر viewpoint**، كما هو موضح بالشكل 6.6. والإسقاط المنظورى يرسم مقصر الخطوط foreshortening وهو غير خطى non-linear



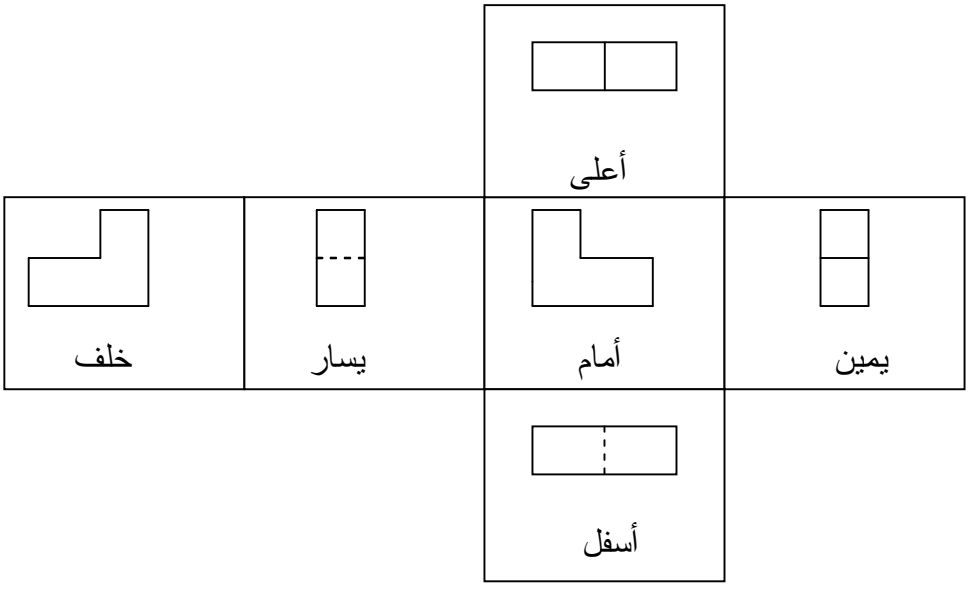
الشكل 6.6: الإسقاط المنظور

- **معايرة أسطح الإسقاط:**

لتقليل الخلط الذي قد يحدث عند التعامل مع مسائل الإسقاط، يجب علينا **معايرة** أسطح الإسقاط بجعلها دائما موازية للسطح z=0 (السطح الذي يحتوى على المحور x والمحور y). إن هذه المعيارية لا تحد من **العمومية، فإذا كان** سطح الإسقاط غير موازياً للسطح z=0 يمكننا استخدام **تحويلات الإحداثيات** في الفضاء الثلاثي الأبعاد لجعله موازياً للسطح z=0. و**سيشاهد** في هذا المقرر منظر الكائن **في النصف الموجب** من الفضاء ($z > 0$) **فقط**، وعليه فإن الإسقاطات التي تبدأ في الرؤوس ستمتد في اتجاه z السالب.

❖ **الإسقاط المتعامد المتعدد المشاهد:**

يعتبر الإسقاط المتعامد **أكثر** أنواع الإسقاطات استخداماً في التخطيط بالحاسوب وخاصة في **مجال العلوم الهندسية** وذلك لإعطاء مشاهد متعددة للكائن، كما هو موضح في الشكل 6.7.



الشكل 6.6: تمثيل الإسقاط المتعامد المتعدد المشاهد

في الشكل 6.8، اعتبر النقطة P(x,y,z) والتي أسقطت على السطح−xy ، نقطة الإسقاط P هي النقطة P'(x',y',0)، التحويل الذي ينتج هذا الإسقاط يمكن التعبير عنه في هيئة المصفوفة على النحو التالي:

$$[x' \quad y' \quad 0 \quad 1]=[x \quad y \quad z \quad 1]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad 6.1$$

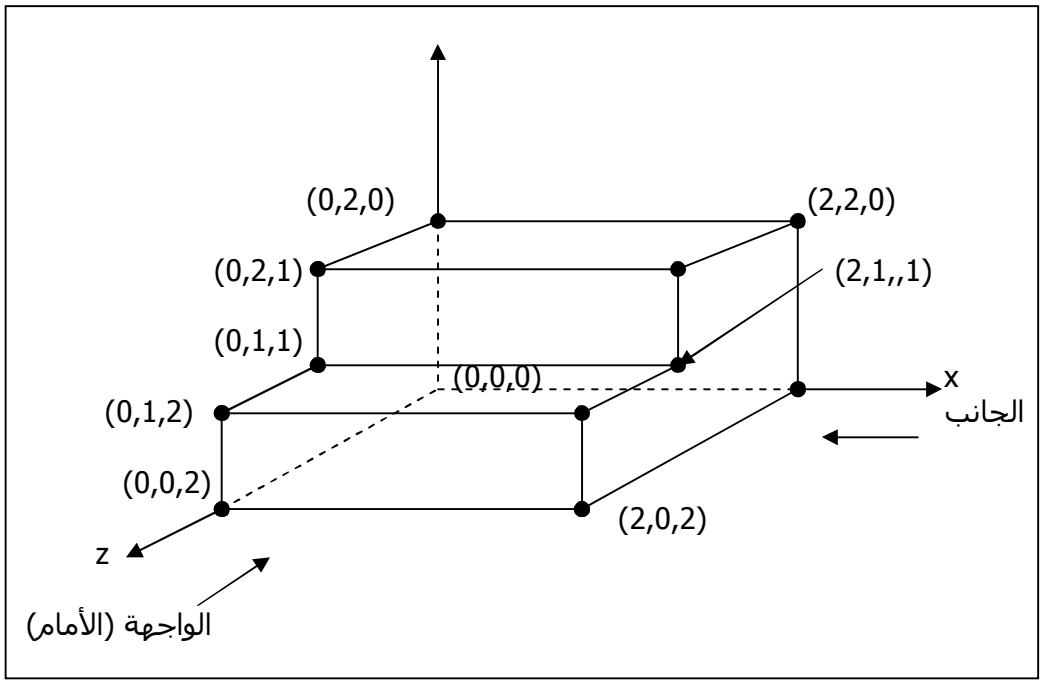
نفس الأسلوب يمكن أن يستخدم للإسقاط على السطح xz والسطح yz، وتعطـــى المصفوفات المقابلة بالتالي:

$$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad 6.2$$

$$\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad 6.3$$

مثال:

خذ الكائن الموضح في الشكل 6.7. أوجد الإسقاط المتعامد للوجه front والجنب side في الاتجاه الموضح بالسهم.



الشكل 6.7

<u>الحل</u>

مشهد الواجهة المحدد هو الإسقاط على السطح xy، وباستخدام المعادلة 6.1، تصبح مصفوفة النقاط كالآتي:

$$[P']_{xy} = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

- نتحصل على مشهد الجانب

باستخدام الإسقاط المتعامد على السطح yz.

باستخدام المعادلة 6.2 نتحصل على

$$[P']_{yz} = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

تستخدم دعوات الدوال التالية للإسقاطات المتعامدة في مكتبة التخطيط *OpenGL*

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
```

ثم يتبع بواحد من

```
glOrtho(left, right, bottom, top, near, far)
gluOrtho2D(left,right,bottom,top)
```

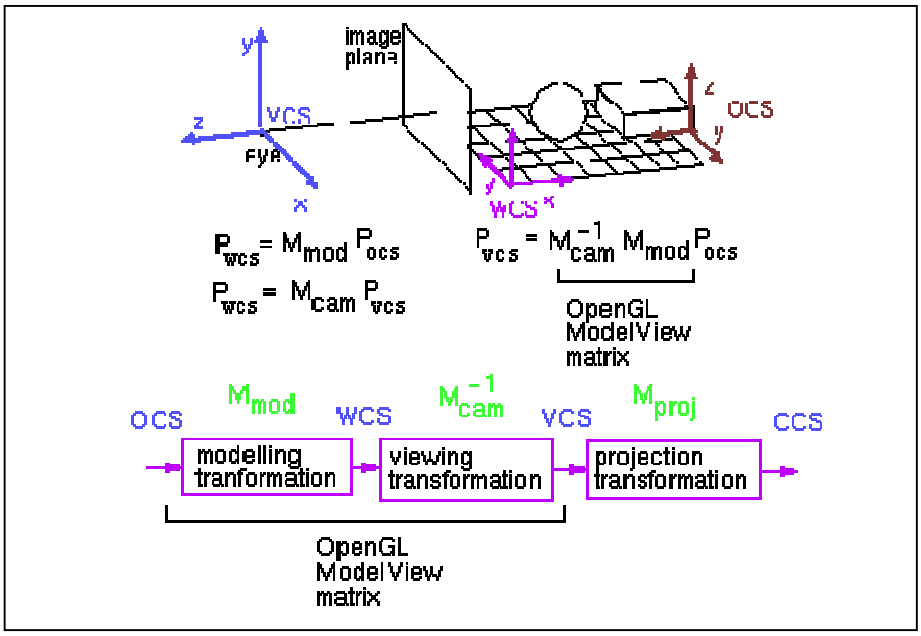
أعلاه يكون near > 0 و far > 0 لوضع أسطح القطع على z=-near و z=-far. إن نداء الدالة `gluOrtho2D()` يشابه نداء الدالة `glOrtho()` مع أخذ near=0 و far=1

أسئلة تقويم ذاتي

1. اذكر أنواع الإسقاطات المستوية
2. كيف تتم عملية مشاهدة الكائن ثلاثي الأبعاد في جهاز عرض ثنائي الأبعاد؟
3. عرف نقطة المنظر viewpoint في الإسقاط المنظوري
4. وضح كيفية معايرة أسطح الإسقاط.
5. ما الفرق بين الإسقاط المتعامد متعدد المشاهد والإسقاط المحاوري؟
6. ما أكثر أنواع الإسقاطات استخداماً في التخطيط بالحاسوب؟

2.تحويلات المشاهدة Viewing Transformations

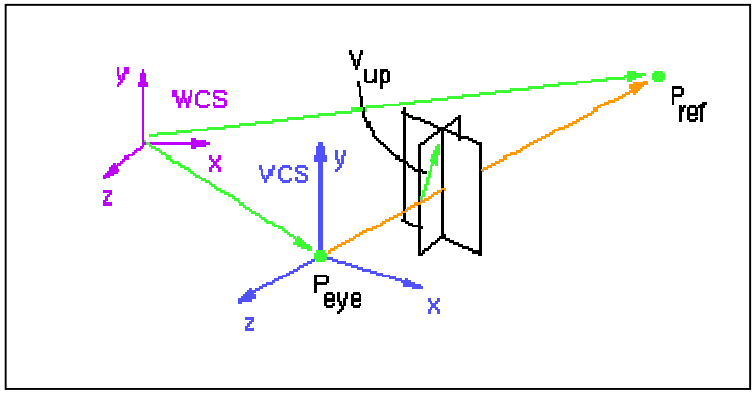
نمذجة التحويلات تعبر عن موقع الكائن في العالم، ويمكننا اشتقاق تحويل المشاهدة بتحديد موقع الكاميرا في العالم أولاً و يعكس هذا التحويل للتعبير عن موقع العالم بالنسبة للكاميرا ، الشكل 6.8.



الشكل 6.8: تحويلات المشاهدة

لتحديد موقع واتجاه الكاميرا قد نستخدم ما يلي كما هو موضح في الشكل 6.9:

- نقطة العين
- نقطة مرجعية
- متجه أعلى



الشكل 6.8: تحديد موقع الكاميرا

مكتبة OpenGL الإضافية توفر الدالة *gluLookAt()* لإنتاج تحويل المشاهدة.

gluLookAt(ex,ey,ez,rx,ry,rz,ux,uy,uz)

حيث *(ex,ey,ez)* تعطى **نقطة العين** the eye point و تعطى *(rx,ry,rz)* **النقطة المرجعية أو نقطة المشاهدة** the reference or 'lookat' point و *(ux,uy,uz)* يعطى **المتجه الأعلى** the up vector.

تستدعى الدالة الطبقات المتعددة اللاحقة postmultiplies للمصفوفة الحالية، وعليه أسهل طريقة لاستخدامها هي:

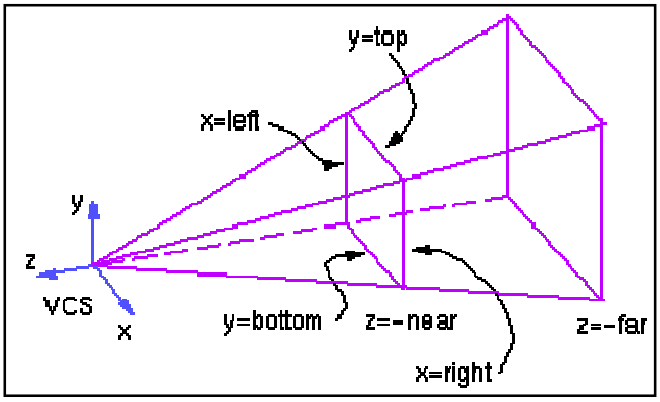
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(ex,ey,ez,rx,ry,rz,ux,uy,uz);
/* setup modelling transformations here */
```

3.أحجام المشهد View Volumes

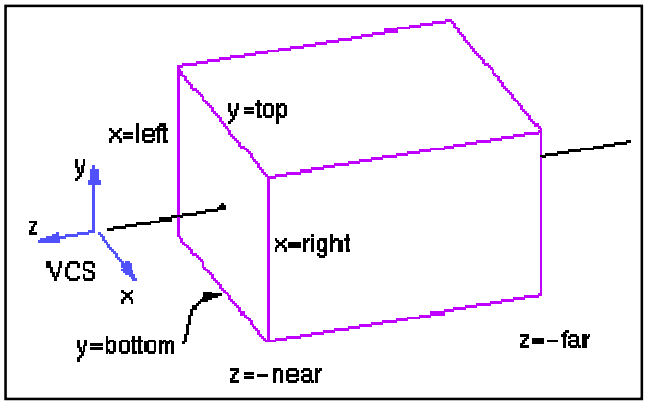
تستخدم إحجام المنظر للقيام بالعمليات التالية:

- **قص** الكائن (أجزاء المشهد) التي لا تسقط في نافذة المشاهدة.
- **تحديد** نطاق z لحسابات الرؤية.

يعطى حجم المشهد المنظورى كما هو موضح في الشكل 6.9. أما حجم المنظر المتعامد فيعطى كما هو موضح في الشكل 6.10.



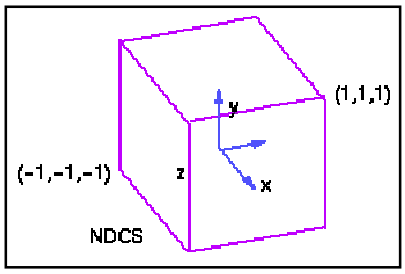
الشكل 6.9: حجم منظر الإسقاط المنظورى



الشكل 6.10. حجم منظر الإسقاط المتعامد

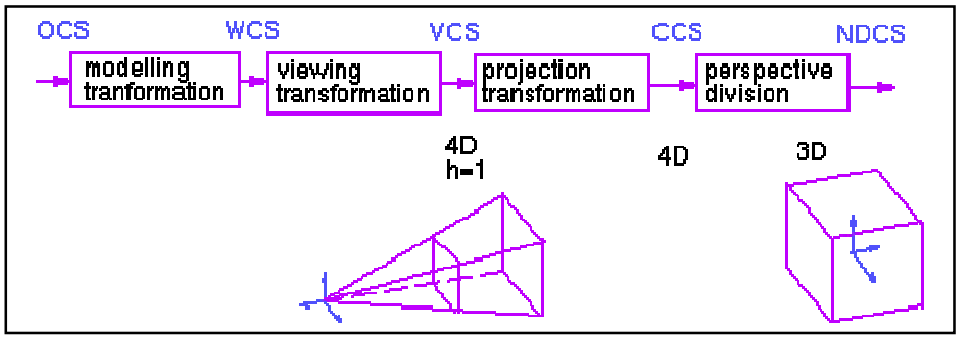
نظام إحداثيات الجهاز المعياري ***normalized device coordinate system*** **or** ***NDCS*:**

يمكن القيام بعمليات القص مباشرة باعتماد أحجام المنظر المحددة، ولكن من الأبسط والأسهل استخدام **أحجام المنظر المخروطية** *canonical* view volume. **يعرف نظام الإحداثيات** باسم نظام إحداثيات الجهاز المعياري *normalized device coordinate system* or *NDCS*، الشكل 6.11.



الشكل 6.11. نظام الإحداثيات NDCS

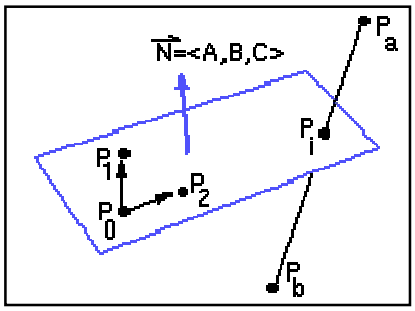
التحويل من نظام إحداثيات النظر VCS إلى نظام إحداثيات الجهاز المعياري NDCS يمكن النظر إليه كجزء من تحويلات الإسقاط، كما هو موضح فى الشكل 6.12.



الشكل 6.12. التحويل بين نظم الإحداثيات

4.معادلات السطح Plane Equations

الشكل 6.13، يوضح العوامل المختلفة للسطح والتي يمكن اســتخدامها لإيجــاد معادلات السطح أدناه والتي تقسم إلى ثلاثة أنواع.



الشكل 6.13. عوامل السطح

- **معادلات السطح الضمنية Implicit plane equation**

$$Ax + By + Cz + D = F(x,y,z)$$

$$F(x,y,z)=0$$

لكل النقاط على السطح الذي يعطى باستخدام: $F(P)=N.P+D$

- **معادلات السطح الوسطية (البارامترية) Parametric plane equation**

السطح Plane(s,t)

$$Plane(s,t) = P0 + s(P1-P0) + t(P2-P0),$$

وذلك باعتبار كل من P0 و P1 و P2 غير مرتبطين خطيا.

بالنسبة للسطح $Plane(s,t)$

$$Plane(s,t) = P0 + s\ V1 + t\ V2,$$

حيث V1 و V2 يمثلان رأسين أساسيَيْن.

- **معادلة السطح غير المباشرة Explicit plane equation**

$$z = -(A/C)x - (B/C)y - D/C\ ,$$

هذه المعادلة صالحة لقيم C الموجبة.

يعطى تقاطع الخط والسطح بالمعادلة:

$$L(t) = Pa + t(Pb - Pa)$$

بالتعويض في معادلة السطح والحل للعامل t نتحصل على:

$$\vec{N} \bullet (P_a + t(P_b - P_a)) + D = 0$$

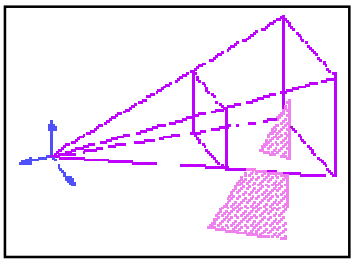
$$t = \frac{-D - \vec{N} \bullet P_a}{\vec{N} \bullet P_b - \vec{N} \bullet P_a} = \frac{-F(P_a)}{F(P_b) - F(P_a)}$$

أسئلة تقويم ذاتي



1. ماذا نستخدم لتحديد موقع الكاميرا؟
2. ما وظيفة الدالة *gluLookAt()* ؟
3. اكتب المعادلة التي تعطي تقاطع الخط والسطح.
4. للقيام بعمليات القص مباشرة باعتماد أحجام المنظر المحددة، لماذا يفضل استخدام أحجام المنظر المخروطية؟
5. ما هي أقسام معاملات السطح؟

5.القص Clipping



الشكل 6.14: القص في الفضاء الثلاثي الأبعاد

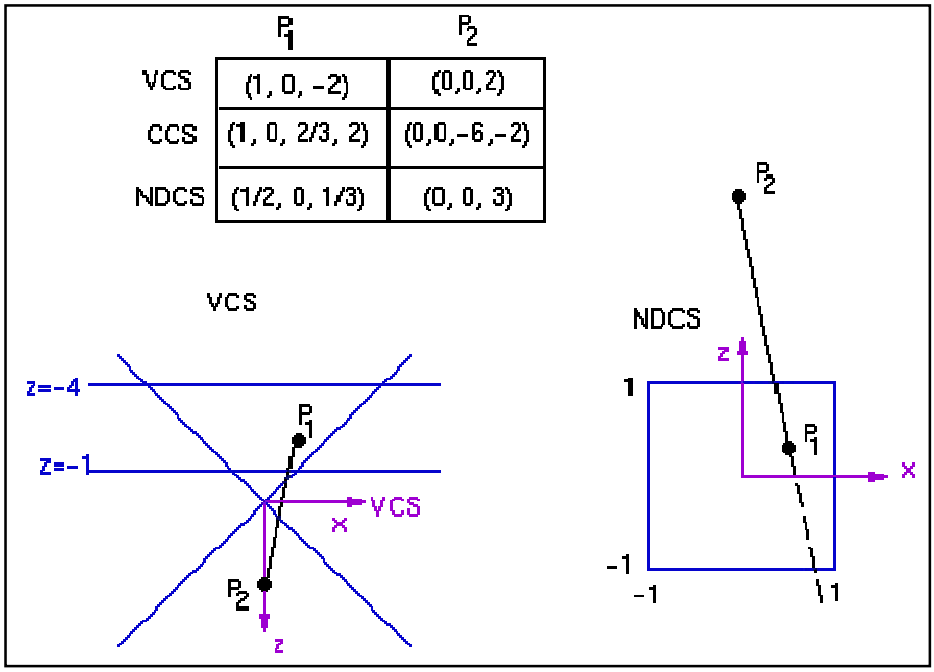
إن خوارزمية **كوهين–سيزارلاند** يمكن تمديدها للتعامل مع المناظر الثلاثية الأبعاد، و **يمكننا** القيام بالقص في **نظم الأحداثيات VCS أو CCS أو NDCS**. وإن كان هناك **عيوب** في **استخدام NDCS**، كما سنذكر لاحقا في هذه الوحدة.

القص في إحداثيات Clipping in VCS:

كل من خوارزمية قص الخط وخوارزمية قص المضلع تستخدم اختبارات الدخل والخرج في نصف الفضاء، وبالنسبة لأحجام المناظر المتعامدة التي تعرضنا لها سابقا، فإن خوارزمية كوهين–سيزارلاند للقص تعمل بنفس الطريقة في حالة الأبعاد الثلاثة مثل في حالة الأبعاد الثنائية، حيث توجد شفرات الرؤوس وتفحص للقبول أو الرفض، فإذا لم يوجد قبول أو رفض بديهي (مباشر)، فإنه يقص الخط مقابل أحد أسطح مشاهدة الحجم الستة، ويختبر مرة ثانية، وتستمر العملية.

- **القص في إحداثيات NDCS:**

تمثل إحداثيات NDCS نظاماً جيد لعمليات القص حيث إن معادلات الأسطح تحدد ببساطة وتبقى دون تغير، بالإضافة إلى ذلك، فإن الخطوط في VCS تماثل خطوطاً في NDCS، وعليه يمكن حساب التقاطعات، مع اعتبار أن الحقيقة هي إن فضاء NDCS لديه اعوجاج غريب نسبة للقسمة قبل الرسم المنظورى. و إن المشكلة التي قد تنتج للقص في NDCS هي فقدان إشارة المعلومات كما هو موضح في المثال الموضح فى الشكل 6.15.



الشكل 6.15: القص في إحداثيات NDCS

- **القص في إحداثيات Clipping in CCS:**

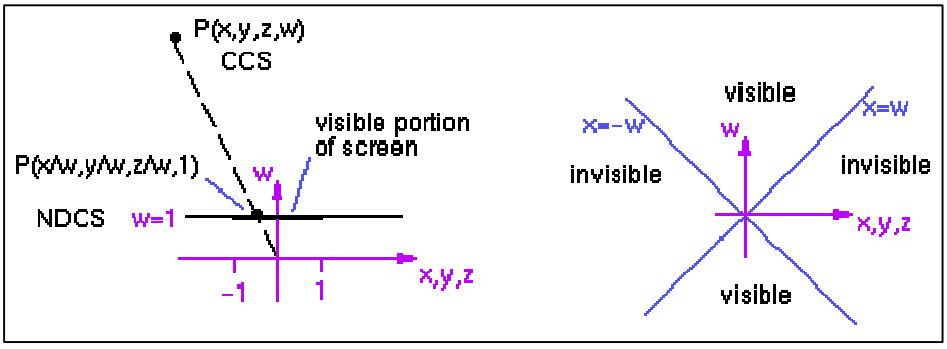
لكي نحدد مدى القص في CSS علينا أولا النظر إلى مدى القص في NDCS

```
-1 <= x/w <= 1
```

ويعني هذا أنه في CSS لدينا

```
-w <= x <= w
```

يمكن المشاهدة كما هو موضح في الشكل 6.16.



الشكل 6.11: القص في CCS

أسئلة تقويم ذاتي

1. كيف تتم عملية القص؟
2. ماهي ميزات وعيوب القص في إحداثيات NDCS؟

- **القص في إحداثيات NDCS**

تدريب (1)

قارن بين القص في إحداثيات Clipping in VCS والقص في إحداثيات NDCS.

الخلاصة

عزيزي الدارس ،تستخدم الإسقاطات لمعرفة التخطيط ووضع الأشكال الثلاثية الأبعاد على سطح ثنائي الأبعاد باستخدام الأنواع المختلفة من الإسقاطات ويتطلب هذا اختيار سطح الإسقاط وتحديد المساقط التي تمر عبر رأس الكائن. فمثلا في الإسقاط الهندسي المستوي توضع رؤوس الإسقاط عند تقاطع المسقط مع سطح الإسقاط .

في الإسقاطات المتوازية يكون مركز الإسقاط في اللاتناهى وعليه تصبح المساقط موازية لبعضها، أما في الإسقاط المنظورى فإن مركز الإسقاط يقع على مسافة محددة من سطح الإسقاط التي تمر كل خطوط الإسقاط عبرها وتسمى نقطة المنظر، وتتم معايرة الآسقاطات لتقليل الخلط الذي قد يحدث عند التعامل مع مسائل ألإسقاط.

إن خوارزمية كوهين–سيزارلاند يمكن تمديدها للتعامل مع المناظر الثلاثية الأبعاد، كما يمكن القيام بالقص في نظم الإحداثيات VCS أو CCS أو NDCS.

لمحة مسبقة عن الوحدة التالية:

الوحدة التالية بعنوان الرؤية ونماذج الإضاءة وسوف نستعرض فيها تنقية الوجه الخلفي و تحديد الرؤية، ونماذج الاستضاءة، والتظليل ، ونماذج الإضاءة في مكتبة OpenGL، و تتبع الإشعاع.

نتمنى لك أخي الدارس أن تستفيد منها

إجابات التدريبات

- تدريب(1)

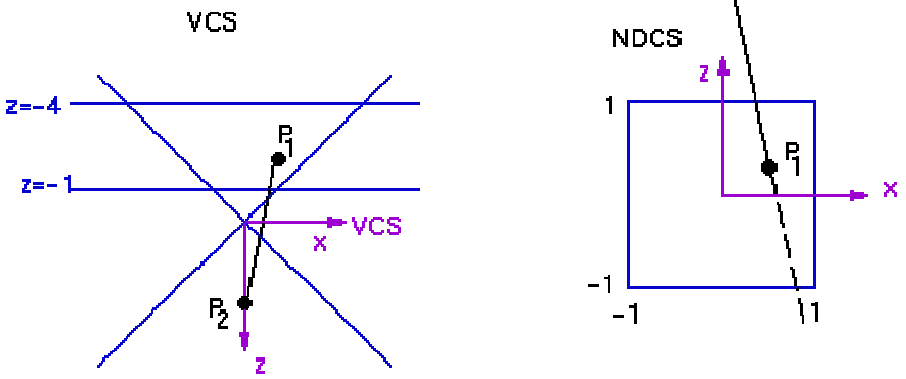
القص في إحداثيات Clipping in VCS:

كل من خوارزمية قص الخط وخوارزمية قص المضلع تستخدم اختبارات الدخل والخرج في نصف الفضاء. بالنسبة لأحجام المناظر المتعامدة التي تعرضنا لهل سابقا إن خوارزمية كوهين–سيزارلاند للقص تعمل بنفس الطريقة في حالة الأبعاد الثلاثة مثل في حالة الأبعاد الثنائية. حيث توجد شفرات الرؤوس وتفحص للقبول أو الرفض. إن لم يوجد قبول أو رفض بديهي (مباشر)، يقص الخط مقابل احد أسطح مشاهدة الحجم الست، ويختبر مرة ثانية، وتستمر العملية.

- **القص في إحداثيات NDCS:**

تمثل إحداثيات NDCS نظام جيد لعمليات القص حيث أن معادلات ألأسطح تحدد ببساطة وتبقى دون تغير. بالإضافة إلى ذلك، الخطوط في VCS تماثل خطوط في NDCS، وعليه يمكن حساب التقاطعات، مع اعتبار الحقيقة إن فضاء NDCS لديه اعوجاج غريب نسبة للقسمة قبل الرسم المنظورى. إن المشكلة التي قد تنتج للقص في NDCS هي فقدان إشارة المعلومات كما هو موضح في المثال الموضح فى الشكل 6.15.

	P_1	P_2
VCS	(1, 0, −2)	(0,0,2)
CCS	(1, 0, 2/3, 2)	(0,0,−6,−2)
NDCS	(1/2, 0, 1/3)	(0, 0, 3)



الشكل 6.15: القص في إحداثيات NDCS

مسرد المصطلحات

الإسقاط projection: هو تحويل من الفضاء الثلاثي الأبعاد إلى سطح جهاز العــرض الثنائي الأبعاد.

الإسقاط الهندسي المستوى planar geometric projections:عبارة عن استخدام أسطح إسقاط مستوية وخطوط مستقيمة للمساقط.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

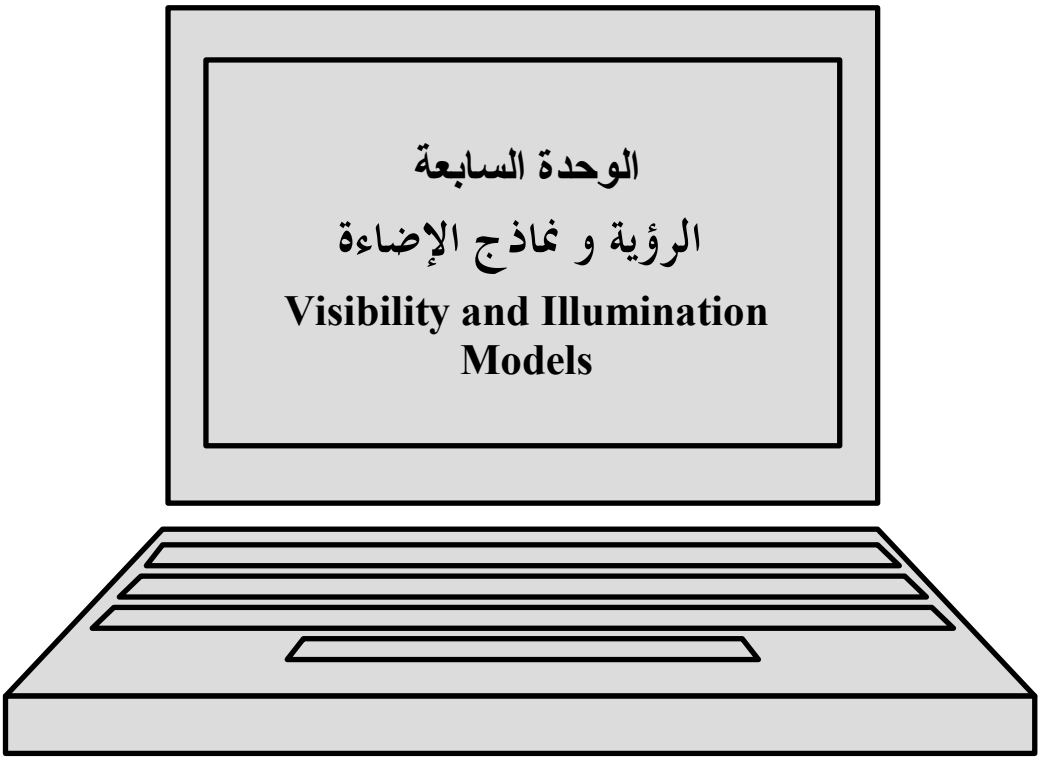
2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة السابعة
الرؤية و نماذج الإضاءة
Visibility and Illumination Models

محتويات الوحدة

الموضوع	الصفحة
المقدمة	175
تمهيد	175
أهداف الوحدة	175
1. تنقية الوجه الخلفي Back-face Culling	176
2.تحديد الرؤية Visibility	178
3. نماذج الاستضاءة illumination Models	179
4.التظليل Shading:	185
5. نماذج الإضاءة في مكتبة OpenGL	186
6. تتبع الإشعاع Ray Tracing	187
الخلاصة	189
لمحة مسبقة عن الوحدة الدراسية التالية	189
إجابات التدريبات	190
مسرد المصطلحات	193
المراجع	194

المقدمة

تمهيد

مرحبا في هذه الوحدة السابعة التي نستعرض فيها الرؤية و نماذج الإضاءة، متناولين بالشرح كيفية تنقية الوجه الخلفي، وتحديد الرؤية، ونماذج الاستضاءة، والتظليل Shading، ونماذج الإضاءة في مكتبة OpenGL، وتتبع الإشعاع كما سنورد خوارزميات لتطبيق المعرفة وأسئلة تقويم ذاتي وتدريبات لكي تساعدك في المراجعة والاستذكار.

أهداف الوحدة

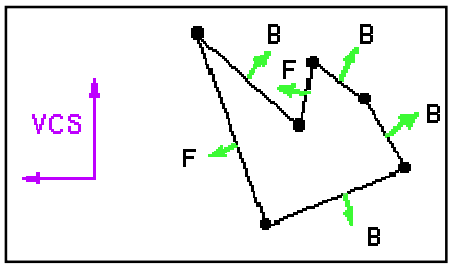


عزيزي الدارس عند انتهائك من دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- توضح كيفية تنقية الوجه الخلفي
- تتعرف تحديد الرؤية ونماذج الاستضاءة
- تتعرف التظليل
- تضبط عوامل الضوء باستخدام مكتبه Open GL

1. تنقية الوجه الخلفي Back-face Culling

تحذف تنقية الوجه الخلفي مباشرة كل **المضلعات** التي لا تواجه المشاهد، كما هو موضح بالشكل 7.1. ويمكن **استخدام** تنقية الوجه الخلفي في **إحداثيات VCS أو NDCS**.

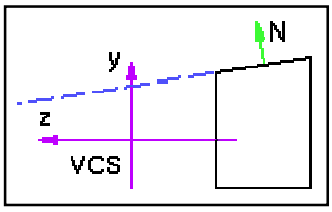


الشكل7.1. حذف الأوجه

لنستعرض عزيزي الدارس ما يلي:

أولاً :القص في VCS:

في إحداثيات VCS قد تكون **المحاولة الأولى** لإجراء تنقية الوجه الخلفي هي مباشرة **استخدام المكون z في السطح المتعامد**، كما يعبر عنه في VCS كما هو موضح بالشكل 7.2 وهذا الأسلوب **يفشل** في كثير من الأحيان.



الشكل7.2. المحاولة المباشرة في نظام VCS

❖ **معادلة السطح للمضلع:**

الإستراتيجية الأفضل لاستخدام تقنية الوجه الخلفي هي **إنشاء معادلة السطح** للمضلع واختبار ما إذا كان نظر المشاهد يسقط على سطح يحتوى على المضلع أم لا،

باختبار سطح المشاهدة Plane(Peye)، و**إن كان** Plane(Peye)<0 يعنـــى إن **نظـــر المشاهد أسفل السطح** الذي يحتوى على المضلع **وعندها تستخدم التنقية**.

❖ **التنقية في VCS:**

يمكن تلخيص التنقية في VCS فيما يلي:

- **احسب** متعامد السطح N=(A,B,C) ويتم معايرته.
- **جد** D في معادلة السطح plane equation باستبدال اى رأس مضلع في معادلة السطح.
- **احسب** Plane(Peye) لتحديد مستوى الرؤية أعلى أو أسفل، ويماثل هذا مراجعة إشارة D.

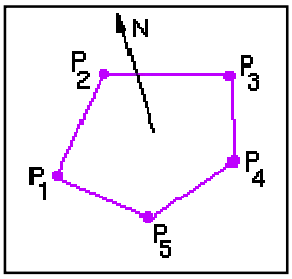
❖ **إيجاد الرؤية في إحداثيات NDCS:**

لنعتبر الآن **تنقية الأوجه** في إحداثيات **NDCS**. في هـــذا النظـــام **المكـــون z لمتعامد** السطح يعكس **الرؤية الحقيقية**. فإن كان هذا المكون **موجباً**، فإن المتعامد يشير **بعيدا عن العين** وعليه يجب تنقية المضلع. الشكل 7.3.

❖ **متعامد السطح:**

من الواضح أننا و للقيام بتنقية الوجه نحتاج إلى **متعامد السطح**.

و **يمكن استخدام** إحدى الطريقتين أدناه **لإيجاد** متعامد السطح.



الشكل7.3. إيجاد الرؤية في نظام NDCS

- **الطريقة الأولى:**

تستخدم **حاصل الضرب الموجه cross-product لحافات المضلعين** و **الترتيب** الذي يستخدم لتخزين الرؤوس يجب أن يكون **متناسقاً** consistence ومثال لهذا، إن حفظت رؤوس المضلع بالترتيب CCW عند النظر إليها من أعلى وجه المقدمة فإننا يمكننا رؤية:

$$N = (P2 - P1) \times (P3 - P2)$$

- **الطريقة الثانية:**

طريقة أخرى أكثر صعوبة وهي استخدام **الأسطح المسقطة في yx و xz و yz. لتحديد المساحات** التي يمكن استخدامها **لحساب المتعامد**، لنأخذ أولاً حالة **ثنائية** الأبعاد.

2. تحديد الرؤية Visibility

نحتاج **لخوارزميات الرؤية لتحديد مكونات المشهد** التي تحجبها مكونات أخرى، ويمكن **تصنيف** هذه الخوارزميات في مجموعتين:

- **خوارزميات فضاء المنظر image-space algorithms:**
  - **تستخدم** لبدائيات العرض مثل عنصر الصورة و خطوط المسح.
  - **تحسن** الرؤية حتى مستوى دقة العارض.
  - **أمثلة:** الخازنة الويبطة–z و وادكين Watkin's و تتبع الشعاع ray-tracing.
- **خوارزميات فضاء الكائن object-space algorithm:**
  - **فواصل** المجال الثنائي.
  - **اختلافات** في خوارزميات الدهان (التظليل) painter's algorithm.
  - أسوا حالة: إنتاج $O(N^2)$ بدائية من N بدائية أصلية.

أسئلة تقويم ذاتي

(1)وضح العملية التي تتم لتنقية الوجه الخلفي

(2) لخص تنقية الوجه الخلفية في إحداثيات VCS

(3) ماهي الإستراتيجية الأفضل لاستخدام تنقية الوجه الخلفي؟

(4) بين الطرق المستخدمة لإيجاد متعامد السطح لتنقية الوجه الخلفي؟

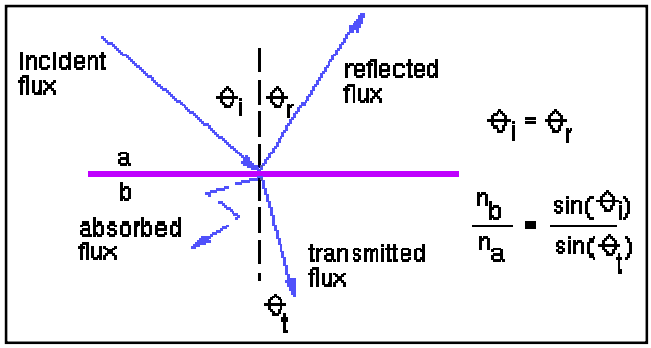
تدريب (1)

1. المحاولة الأولى لإجراء تنقية الوجه الخلفي في إحداثيات VCS تفشل في كثير من الأحيان، صف المعالجة المطلوبة
2. إذا كان Plane(Peye)<0 فأين يقع نظر المشاهد؟
3. في ماذا تستخدم خوارزميات فضاء المنظر image-space algorithms؟

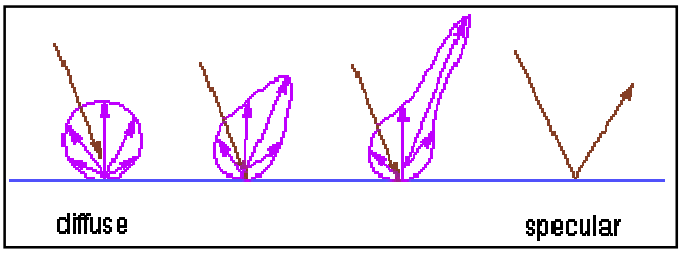
3. نماذج الاستضاءة illumination Models

❖ **الاستضاءة المحلية local illumination:**

يحدد ضوء منفرد يتقاطع مع سطح واحد.



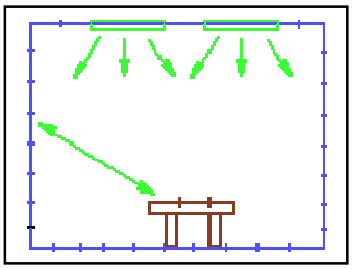
- **الصفات العامة لنماذج الاستضاءة المحلية Local Illumination Models هي:**



  - **السرعة** في الحساب.
  - **تجريبي** heuristic وغير كامل.
  - **جيدة** للضوء في اتجاه نقطة المشاهدة.

❖ **الاستضاءة الشاملة global illumination:**

تعطى نماذج لتبادل الأضواء بين جميع السطح و نهتم هنـــا باستضـــاءة موقـــع معين، وقد اتت أهمية هذا من مجالات انتقال الحرارة والتطبيقات الهندسية والعلمية.



- **مكونات الاستضاءة:**

أكثر **نماذج الإضاءة** المستخدمة لديها **ثلاثة مكونات:**

الاستضاءة = المحيط + الانتشار + البريق

I = ambient + diffuse + specular

♦ **المحيط:**

حد المحيط يسمح ببعض **التحكم** الشامل في **شدة إضاءة المشهد، نموذجيا**

$$I = Ia * ka$$

حيث **Ia ثابت استضاءة المحيط** يحدد مرة واحدة لكل المشـــهد، و **ka معامـــل انعكاس المحيط** ambient reflection coefficient وعادة **يقيد** بان يكون **فـــي المدى [0,1]**.

- **انتشار الانعكاس:**

عادة **يتصف** انتشار الانعكاس بالتالي:

- **انعكاس** لامبرتين *Lambertian* reflection.
- **شدة إضاءة** السطح الظاهرة لا تعتمد على زاوية المشاهدة.
- **توزيع الضوء المبعثر** scattered light لا يعتمد على اتجاه وصول الضوء.

ويعطى حد الانتشار بالمعادلة التالية:

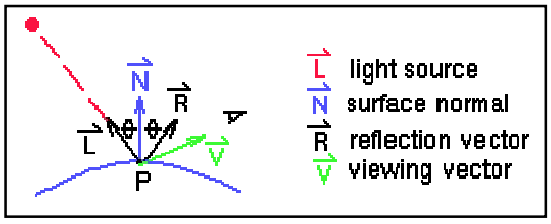
$$Id = Ii\ k_diff \cos(\theta),$$

حيث

- **Ii** : شدة مصدر الضوء i.
- **k_diff**: معامل انعكاس السطح.
- **θ الزاوية** بين اتجاه الضوء ومتعامد السطح وتكون قيمة في المدى [-π/2, π/2].

- **البريق:**

العامل الأخير الذي يحدد الاستضاءة المحلية هو البريق، والشكل أدناه يوضح الوضع:

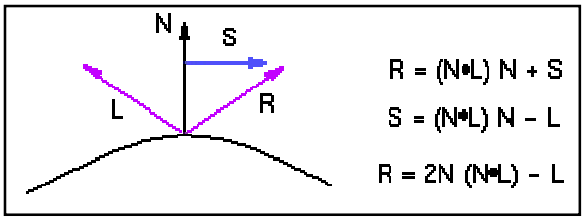


- **حساب البريق:**

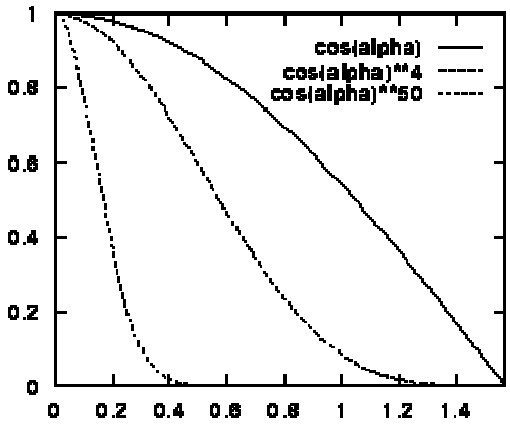
يستخدم نموذج فونق عادة لحساب البريق كالآتي:

$$I_s = I_i\, k_{spec}\, \cos^n(\alpha)$$
$$= I_i\, K_{spec}\, (R.V)^n$$

حيث k_spec معامل بريق السطح، **α** الزاوية بين متجه الانعكاس والمشاهدة **،عامل السطح n** يستخدم لتحديد درجة خشونة السطح للسطح الأملس (المرآة) n=∞ للأسطح الخشنة جدا n=1. و تحسب R كالآتي:



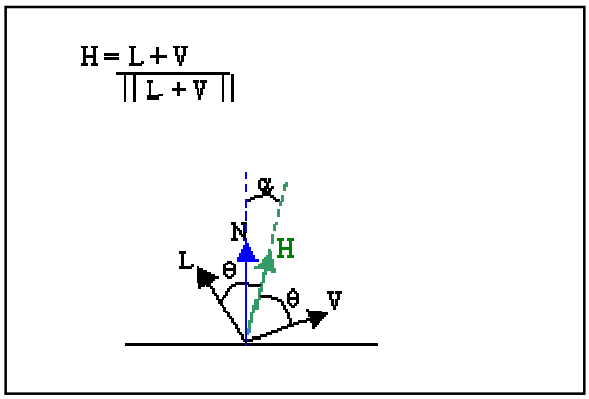
الدالة $\cos^n(\alpha)$ تبدو كالآتي:



- **نموذج بليين Blinn:**

يمكن أيضا حساب البريق باستخدام نموذج بليين Blinn الذي توافق أكثـــر مـــع نتائج التجارب، ويستخدم متجه منتصف المسافة H كالآتي:

$$I_s = I_i\, k_{spec} \cos^n(alpha)$$
$$= I_i\, K_{spec}\, (N.H)^n$$



- **ميزات نموذج بليين Blinn:**

ميزات هذا النموذج تشتمل على:

- ✓ الأساس النظري.
- ✓ حاصل الضرب N.H لا يمكن أن يكون سالباً إذا كان $N.L > 0$ و $N.V > 0$.
- ✓ بالنسبة لاتجاه الضوء والمسقط المتعامد قيمة H تكون ثابتة.

- **النموذج الكلي للاستضاءة المحلية:**

نتحصل على النموذج الكلي للاستضاءة المحلية بضم النماذج المختلفة المذكورة أعلاه **وبافتراض نموذج فونق للاستضاءة نتحصل على:**

$$\mathbf{I = I_a\, ka + I_i\, k_{diff}\, (N.L) + I_i\, k_{spec}\, (R.V)^n}$$

حيث أن كلاً من العوامل k_a و k_{diff} و k_{spec} تصف الأسطح المختلفة وتكون قيمتها فـــي المدى من 0 إلى 1.

○ **التعامل مع الألوان:**

للتعامل مع الألوان تستخدم ثلاثة معادلات مشابهة للمعادلات أعلاه وهى:

$$I_r = I_{a\ r}\, k_{a\ r} + I_{i\ r}\, k_{diff\ r}\, (N.L) + I_{i\ r}\, k_{spec\ r}\, (R.V)^n$$
$$I_g = I_{a\ g}\, k_{a\ g} + I_{i\ g}\, k_{diff\ g}\, (N.L) + I_{i\ g}\, k_{spec\ g}\, (R.V)^n$$
$$I_b = I_{a\ b}\, k_{a\ b} + I_{i\ b}\, k_{diff\ b}\, (N.L) + I_{i\ b}\, k_{spec\ b}\, (R.V)^n$$

عند استعمال هذا النموذج يلاحظ ما يلي:

- عندما تكون قيمة $I1 < 0$ تقص I1 إلى واحد.
- عندما تكون $N.L < 0$ تقص I1 إلى صفر أو يقلب المتعامد.
- للتعامل مع مصادر ضوء متعددة، اجمع شدة كل منها للحصول على الشدة الكلية.

أسئلة تقويم ذاتي

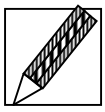


(1) ما هي الصفات العامة لنماذج الاستضاءة المحلية؟

(2) اذكر مكونات أكثر نماذج الإضاءة استخداماً

(3) ماهي ميزات نموذج بليين؟

تدريب (2)



1. كيف يتم حساب البريق باستخدام نموذج بليين Blinn؟
2. في الاستضاءة الشاملة نهتم بإضاءة موقع معين لماذا؟
3. في معادلة الانتشار التالية:

$$\mathbf{Id = Ii\ k_diff \cos(\theta)}$$

وضح المراد من الآتي: i) Ii : ii) k_diff :iii) cos(θ)

4.التظليل Shading

تستخدم في التخطيط بالحاسوب

♦**خوارزميات التظليل Shading Algorithms** التالية:

- **التظليل المستوي *flat shading***

يستخدم نموذج الاستضاءة مرة واحدة للمضلع، ويستخدم هذا لوناً لكل المضلع.

- **تظليل قورود *Gouraud shading***

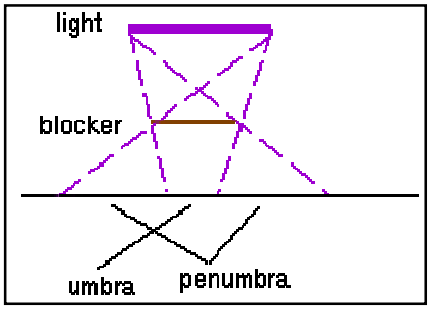
يحسب نموذج الاستضاءة عند الرؤوس، ثم يستكمل هذه الألوان عبر المضلع.

- **تظليل فوتق *Phong shading***

يستكمل المتعامدات خلال عملية تحويل المسح، ثم يطبق نموذج الاستضاءة عند كل عنصر صورة.

♦**طرق تظليل المشهد:** يمكن تظليل المشهد بأحد الطرق التالية:

- الأداء الإسقاطي projective rendering.
- يرسم الكائن لمرة أخرى باستخدام تحويلات تظليل إضافية.
- تسويد الكائن الموجود.
- يستخدم خازنة الطباعة الوسيطة stencil buffer ليمنع التظليل المزدوج.
- double shadows'.
- تتبع الإشعاع ray tracing.
- جزء من الطريقة.
- يبث أشعاع التظليل.
- مساحة مصادر الضوء area light sources.



♦ **مشاكل خوارزميات التظليل:** تعانى خوارزميات التظليل من المشاكل التالية

- الاعتماد على الاتجاه.
- الصورة غير الظليلة silhouettes.
- المشهد المشوش.
- توليد متعامدات الرؤوس.

5. نماذج الإضاءة في مكتبة OpenGL

يُستدعى ما يلي **لضبط عوامل الضوء**

```
 glLightfv(GL_LIGHT0, GL_AMBIENT,
amb_light_rgba );
 glLightfv(GL_LIGHT0, GL_DIFFUSE,
dif_light_rgba );
 glLightfv(GL_LIGHT0, GL_SPECULAR,
spec_light_rgba );
 glLightfv(GL_LIGHT0, GL_POSITION, position);
 glEnable(GL_LIGHT0);
```

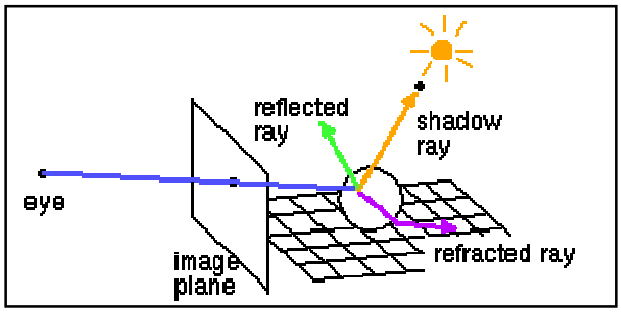
المستدعيات التاليات **تحدد خواص السطح** لاستخدامها فيما بعد في كل الكمائن (الأشياء) التي يتم رسمها.

```
glMaterialfv( GL_FRONT, GL_AMBIENT, ambient_rgba );
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse_rgba );
glMaterialfv( GL_FRONT, GL_SPECULAR, specular_rgba );
glMaterialfv( GL_FRONT, GL_SHININESS, n );
```

6. تتبع الإشعاع Ray Tracing

يستخدم أسلوب تبع الإشعاع للآتي:

- لضبط المشاهدة والمشاهد (الكمائن) الشفافة transparent objects.
- استضاءة شاملة للضوء المرآوى.
- استضاءة جزئية على موقع معين في الهندسة الضوئية geometric optics.



أسئلة تقويم ذاتي

(1) اذكر خوارزميات التظليل موضحاً استخدماتها

(2) عدد الطرق المختلفة لتظليل المشهد

(3) ما المشكلات التي تعاني منها خوارزميات التظليل؟

تدريب(3)



1. اضبط عوامل الضوء باستخدام مكتبة OpenGL؟
2. حدد خواص السطح التي ستستخدم لرسم الكائن في مكتبة OpenGL ؟
3. في ماذا يستخدم أسلوب تتبع الإشعاع؟

الخلاصة

لإعطاء رؤية واضحة للمشهد عزيزي الدارس يتم استخدام تقنية الوجه الخلفي لتقوم بحذف كل المضلعات التي لا تواجه المشاهد وأفضل الإستراتجيات لاستخدام هـذه التقنية هي إنشاء معادلة السطح للمضلع لاختبار إن كان نظر المشاهد يسقط على سـطح يحتوي على المضلع أم لا.

ولحساب متعامد السطح نستخدم طريقتين؛ طريقة حاصـل الضـرب الموجـه cross-product لحافات المضلعين أو طريقة الأسطح المسقطة فـي yx و xz و yz. ولتحديد المساحات التي يمكن استخدامها لحساب المتعامد، أما تحديد الرؤية فتستخدم لـه خوارزميات عديدة كخوارزمية فضاء المنظر مثلا.

تتأثر الاستضاءة بمحيط السطح وانتشار الضوء وبريقه فخشونة السـطح تـأثر على انتشار الضوء، وتحسب درجة خشونة السطح الأملس (المـرآة) n=∞ وللأسـطح الخشنة جدا n=1، أما البريق فيمكن حسابه بواسطة نموذج فونق أو بواسـطة نمـوذج بلبين.

التعامل مع الألوان يكون بمعادلات معينة ، أما التظليل فتستخدم له خوارزميات خاصة لكن جميعها يعاني من مشاكل عدة منها على سبيل المثال الاعتماد على الاتجاه والصورة غير الظليه .

لمحة مسبقة عن الوحدة التالية

الوحدة التالية بعنوان المنحنيات والأسطح وسوف تتعرف فيها على المنحنيات الوسيطة (البارامترية) ومنحنيـات هيرميـت و منحنيـات بيزيـر، والاسـتمرارية الهندسـية و البارامترية، ثم الأسطح البارامترية.

إجابات التدريبات

تدريب(1)

1. **المعالجة** هي **إنشاء معادلة السطح** للمضلع والاختبار إن كان نظر المشاهد يسقط على سطح يحتوى على المضلع أو لا.
2. يعنى إن **نظر المشاهد أسفل السطح** الذي يحتوى على المضلع .
3. **تستخدم** على بدائيات العرض مثل عنصر الصورة و خطوط المسح.

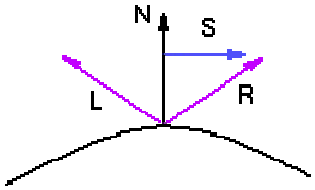
تدريب(2)

.1

يستخدم نموذج فونق عادة لحساب البريق كالآتي:

$$I_s = I_i\, k_{spec}\, \cos^n(\alpha)$$
$$= I_i\, K_{spec}\, (R.V)^n$$

حيث k_spec معامل بريق السطح، **α** الزاوية بين متجه الانعكاس والمشاهدة ،**عامل السطح n** يستخدم لتحديد درجة خشونة السطح للسطح الأملس (المرآة) $n=\infty$ للأسطح الخشنة جدا $n=1$. و تحسب R كالآتي:

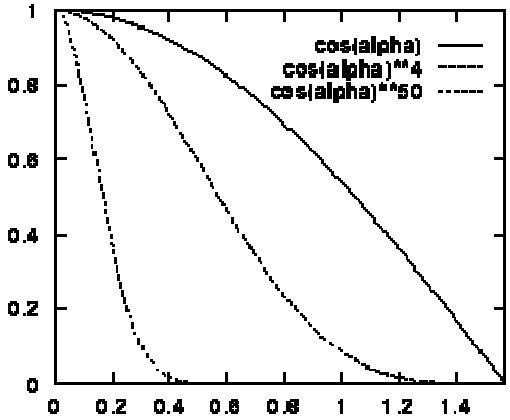


$R = (N \bullet L)\, N + S$

$S = (N \bullet L)\, N - L$

$R = 2N\, (N \bullet L) - L$

الدالة $\cos^n(\alpha)$ تبدو كالآتي:



2. واتت أهمية هذا من مجالات انتقال الحرارة والتطبيقات الهندسية والعلمية.

3. **Ii** : شدة مصدر الضوء i.

k_di: معامل انعكاس السطح.

θ الزاوية بين اتجاه الضوء ومتعامد السطح وتكون قيمة في المدى $[-\pi/2, \pi/2]$.

تدريب(3)

1. **يستدعى** التالي **لضبط عوامل الضوء**

```
glLightfv(GL_LIGHT0, GL_AMBIENT, amb_light_rgba );
glLightfv(GL_LIGHT0, GL_DIFFUSE, dif_light_rgba );
glLightfv(GL_LIGHT0, GL_SPECULAR, spec_light_rgba );
glLightfv(GL_LIGHT0, GL_POSITION, position);
glEnable(GL_LIGHT0);
```

2. **المستدعيات** التاليات **يحددون خواص السطح** لاستخدامهم فيما بعد في كل الكمائن (الأشياء) التي يتم رسمها.

```
 glMaterialfv( GL_FRONT, GL_AMBIENT, ambient_rgba
);
 glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse_rgba );
 glMaterialfv( GL_FRONT, GL_SPECULAR, specular_rgba
);
 glMaterialfv( GL_FRONT, GL_SHININESS, n );
```

3. تعانى خوارزميات التظليل من المشاكل التالية:

- الاعتماد على الاتجاه.
- الصورة الغير ظليه silhouettes.
- المشهد المشوش.
- توليد متعامدات الرؤوس.

مسرد المصطلحات

تنقية الوجه الخلفي: تنقية تستخدم لحذف كل المضلعات التي لا تواجه المشاهد.

الاستضاءة: نعني بها تقاطع الضوء مع السطح سواء كان مفرداً مع سطح واحد أو أضواء مع جميع السطح.

التظليل المستوي: يستخدم نموذج الاستضاءة مرة واحدة للمضلع، ويستخدم هذا لوناً لكل المضلع.

تظليل قورود: يحسب نموذج الاستضاءة عند الرؤوس، ثم يستكمل هذه الألوان عبر المضلع.

تظليل فوتق: يستكمل المتعامدات خلال عملية تحويل المسح، ثم يطبق نموذج الاستضاءة عند كل عنصر صورة.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison- Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة الثامنة
المنحنيات و الأسطح
Curves and Surfaces

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**197**
تمهيد	**197**
أهداف الوحدة	**197**
1. المنحنيات الوسيطة (البارامترية)	**198**
2. منحنيات هيرميت	**204**
3. منحنيات بيزير	**206**
4. منحنيات بيزير من الدرجة N N BEZIER CURVES OF DEGREE	**208**
5. **منحنيات هيرميت وبيزير المتقطعة PIECEWISE HERMITE AND BEZIER CURVES**	**208**
6. **الاستمرارية الهندسة والبارامترية GEOMETRIC AND PARAMETRIC CONTINUITY**	**209**
7. **الأسطح البارامترية Parametric Surfaces**	**210**
الخلاصة	**214**
لمحة مسبقة عن الوحدة التالية	**214**
إجابات التدريبات	**215**
مسرد المصطلحات	**216**
المراجع	**217**

المقدمة

تمهيد

أخي الدارس مرحباً بك إلى الوحدة الثامنة من هذا المقرر والتي هي بعنوان المنحنيات والأسطح وسوف نشرح فيها بالتفاصيل المنحنيات الوسيطة (البارامترية)، و منحنيات هيرميت، و منحنيات بيزير، والاستمرارية الهندسية و البارامترية ، ثم الأسطح البارامترية. وسوف تجد عزيزي الدارس في هذه الوحدة أمثلة للشرح والتطبيق وكذلك سوف تجد تدريبات لها أجوبة في نهاية الوحدة ، وأسئلة تقويم ذاتي، نرجو منك أن توليها اهتمامك حتى تستفيد منها ، كما نرجو أن تشاركنا في نقد هذه الوحدة وتقويمها.

أهداف الوحدة



عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة يجب أن تكون قادراً على أن:

- تميز بين المنحنيات الوسيطة (البارامترية) و منحنيات هيرميت و منحنيات بيزير.
- تناقش الاستمرارية الهندسية و البارامترية.
- تصف الأسطح البارامترية.

1. المنحنيات الوسيطة (البارامترية) Parametric Curves

يمكن تمثيل المنحنيات والأسطح **بطرق** مباشرة أو غير مباشرة أو باستخدام تمثيل **وسيطة** (بارامترية).

إن التمثيل **البارامتري** لا يكون في العادة **أحادى**، ولتوضيح هذا خذ التمثيل البارامتري للخط:

$$L(P_0,P_1) = P_0 + u(P_1-P_0),\ u = [0...1]$$
$$L(P_0,P_1) = v(P_1-P_0)/2 + (P_1+P_0)/2,\ v = [-1...1]$$

يمكن تغير العوامل لنأخذ أي قيمة بين قيدين معينين، لتحويل العوامل من $u = [a...b]$ إلى $w = [0...1]$ يمكننا استخدام $w = (u-a)/(b-a)$ والتي تعطي $u = w(b-a) + a$ وعليه نتحصل على

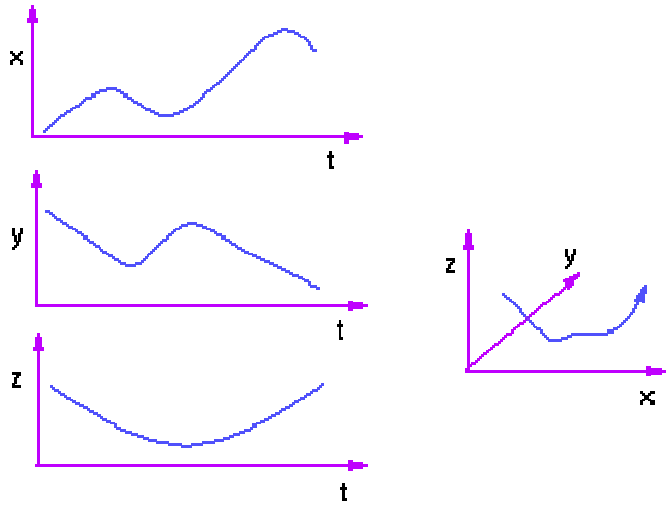
$$P(u),\ u = [a...b] = P(w(b-a)+a),\ w = [0...1]$$

❖ المنحنيات المكعبة cubic curves:

تستخدم بكثرة في التخطيط، نسبة لعدم مرونة منحنيات الترتيب الأدنى ومنحنيات الترتيب الأعلى حيث إنها تضيف تعقيدات غير مطلوبة، وتحدد عوامل المنحنى التكعيبي بما يلي:

$$x(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$
$$y(t) = b_3t^3 + b_2t^2 + b_1t + b_0$$
$$z(t) = c_3t^3 + c_2t^2 + c_1t + c_0$$

في العادة يمكن اعتبار $t = [0...1]$.



يمكن كتابة المعادلات الوسيطة (البارامترية) أعلاه بصورة مدمجة على النحو التالي:

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$x(t) = T \cdot A$$

بطريقة مماثلة يمكننا كتابة :

y(t) = T B
z(t) = T C

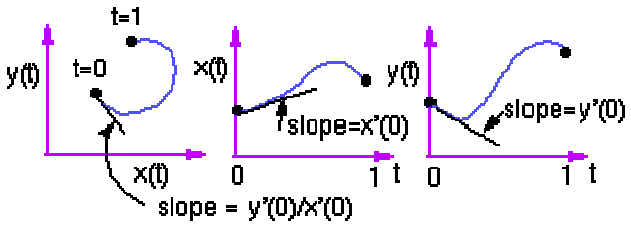
كل بعد يتم التعامل معه بصورة مستقلة عن بقية الأبعاد، وعليه يمكننا التعامل مع منحيات لديها أي عدد من الأبعاد.

مشتقات derivatives المنحنى:

مشتقات derivatives المنحنى بالنسبة إلى t يعبر عنه كالآتي:

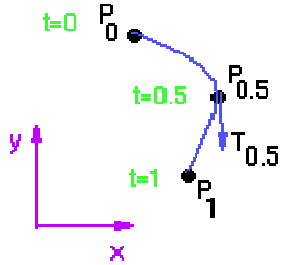
x'(t) = [3t2 2t 1 0] A

من السهل **اعتبار العامل t** يمثل الزمن بالترتيب لمشاهدة بعض مواصفات المنحنى، **للمشتقات تفسيرات أولية** في **الفضاء الديكارتي** للمنحنى:

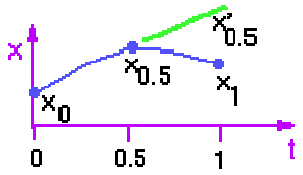


مثال

لنفترض أننا نريد تحديد منحنى
تكعيبي بحيث يمكن للمستخدم تحديد موقع
نقاط النهاية endpoints والنقطة الوسيطة
midpoint بالإضافة إلى المماس عند النقطة الوسيطة.
الشكل أدناه يوضح بعض نماذج المنحنيات.



في البدء ننشئ المنحنى التكعيبي x(t) و لتحقيق هذا يوجد أربع شروط constraints لابد من تحقيقها وأربع مجاهل.



$$x(0) = [\,0\ 0\ 0\ 1\,]\ A$$
$$x(0.5) = [\,0.5^3\ \ 0.5^2\ \ 0.5^1\ \ 0.5^0\,]\ A$$
$$x'(0.5) = [\,3(0.5^2)\ \ 2(0.5)\ \ 1\ \ 0\,]\ A$$
$$x(1) = [\,1\ 1\ 1\ 1\,]\ A$$

or $G_x = B\,A$ where

$$G_x = \begin{bmatrix} x_0 \\ x_{0.5} \\ x'_{0.5} \\ x_1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.125 & 0.25 & 0.5 & 1 \\ 0.75 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

يمكننا الحل لإيجاد A باستخدام $A = B^{-1}x$، وعليه تكون المعادلة

النهائية × على النحو التالي:

$$x(t) = T\ B\text{-}1\ Gx$$

❖ المصفوفة الأساس:

المصفوفة B^{-1} تدعى مصفوفة الأساس ويرمز إليها بالحرف M ، وعليه يمكننا كتابة:

$$x(t) = T\ M\ Gx$$

وفى هذه الحالة يكون لدينا

$$M = \begin{bmatrix} -4 & 0 & -4 & -4 \\ 8 & -4 & 6 & -4 \\ -5 & 4 & 2 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

أخيرا يمكننا أيضا كتابة

$$x(t) = [\,f1(t)\ f2(t)\ f3(t)\ f4(t)\,]\ Gx$$

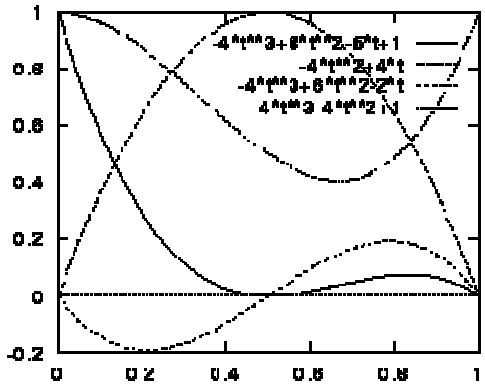
❖ دوال الأساس أو الركائز blending or basis functions :

حيث يتحصل على **الدوال** $f_1(t)...\ f_4(t)$ من حاصـــل **ضـــرب** **M * T** .

تدعى هذه الدوال دوال الأساس أو الركائز blending or basis functions ، **حيث** إنها تعطى **الأوزان** للمكونات المختلفة **للمتجه الهندسي G**، وفى هذا المثـــال هـــذه الأوزان هي:

f1(t) = -4t3 + 8t2 - 5t + 1
f2(t) = -4t2 + 4t
f3(t) = -4t3 + 6t2 - 2t
f4(t) = 4t3 - 4t2 + 1,

حيث الدالة $f_1(t)$ الوزن بالنسبة إلى P_o و $f_2(t)$ الوزن بالنسبة إلى $P_{0.5}$ و $f_3(t)$ الوزن بالنسبة إلى $T_{0.5}$ و $f_4(t)$ الوزن بالنسبة إلى P_1. تبدو دوال الأساس على الهيئـــة التالية:



التعبير عن المنحني في شكل معادلة:

المنحنيات y(t) و z(t) **تنشأ** بصورة **مماثلة** لإنشـــاء المنحنـــى x(t)، **وتبقـــى** مصفوفة الأساس ودوال الأساس على نفس الحال فقط يتغير المتجـــه الهندســـي، حيـــث

يعطى المتجه الهندسى للدالة y(t) مكونات y فى P_0 و $P_{0.5}$ و $T_{0.5}$ و P_1، وعادة **يمكننا كتابة (التعبير)** عن المنحنى كمعادلة متجهة واحدة وهى:

$$P(t) = T\ M\ G$$

تشتمل هذه المعادلة على المعادلات الثلاث التالية:

$$x(t) = T\ M\ Gx$$
$$y(t) = T\ M\ Gy$$
$$z(t) = T\ M\ Gz$$

أسئلة تقويم ذاتي

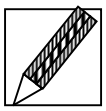


(1)وضح أن التمثيل **البارامتري** لا يكون في العادة **أحادي**

(2)لماذا تستخدم المنحنيات المكعبة بكثرة في التخطيط؟

(3)ماهي أهمية دوال الأساس أو الركائز؟

تدريب(1)



إذا كان المصفوفة B^{-1} تدعى مصفوفة الأساس ويرمز إليها بالحرف M . عليه يمكننا كتابة:

$$x(t) = T\ M\ Gx$$

وإذا كان لدينا في هذه الحالة

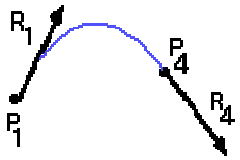
$$M = \begin{bmatrix} -4 & 0 & -4 & -4 \\ 8 & -4 & 6 & -4 \\ -5 & 4 & 2 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

اكتب معادلة الأساس على ضوء ذلك

2. منحنيات هيرميت

مثال

لنعتبر منحنيات هيرميت Hermite
Curve والتي تحدد بنقطتين ومماسين متجهين.



لنشتق معادلة منحنيات هيرميت باستخدام المتجه الهندسي التالي:

$$Gh = [\ P1\ P4\ R1\ R4\ ]T$$

كما ذكر سابقا في هذه الوحدة، فإننا **نعبر عن المنحنى** بما يلي:

$$x(t) = T\ Ah$$
$$= T\ Mh\ Gh$$

❖ شروط تعريف المنحنى:

الشروط التي سوف نستخدمها لتعريف المنحنى هي:

$$x(0) = P1 = [\ 0\ 0\ 0\ 1\ ]\ Ah$$
$$x(1) = P4 = [\ 1\ 1\ 1\ 1\ ]\ Ah$$
$$x'(0) = R1 = [\ 0\ 0\ 1\ 0\ ]\ Ah$$
$$x'(1) = R4 = [\ 3\ 2\ 1\ 0\ ]\ Ah$$

كتابه هذه الشروط في هيئة **مصفوفة** يعطى:

$$Gh = Bh\ Ah$$
$$Ah = (Bh)\text{-}1\ Gh$$
$$x(t) = T\ Ah$$
$$= T\ (Bh)\text{-}1\ Gh$$
$$= T\ Mh\ Gh$$

❖ مصفوفة الأساس لمنحنى هيرميت:

مقلوب المصفوفة B_h يعرف مصفوفة الأساس لمنحنى هيرميت

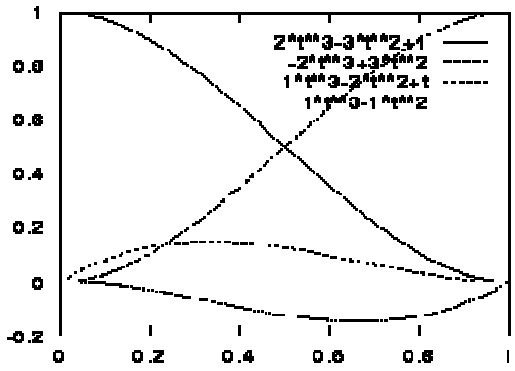
$$M_h = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

❖ دوال الأساس:

كما كان في المثال السابق، **فإنه تمثل** دوال الأساس **الأوزان** للحدود في المتجه الهندسي وتعطى بحاصل ضرب $T\ M_h$، عليه فإن دوال الأساس لمنحنى هيرميت هي:

f1(t) = 2t3 - 3t2 + 1
f2(t) = -2t3 + 3t2
f3(t) = t3 - 2t2 + t
f4(t) = t3 - t2

تبدو دوال أساس منحنى هيرميت على الهيئة التالية:



3. منحنيات بيزير

مثال

منحنيات بيزير Bezier Curves:منحنيات بيزير **تعتبر متغيراً من منحنيات هيرميت،** حيث إنها **تحدد** بأربع نقاط.



المنحنى مرتبطة بشدة بمنحنى هيرميت حيث

P1h = P1
P4h = P4
R1h = 3 (P2 - P1)
R4h = 3 (P4 - P3)

نتحصل على **مصفوفة الأساس** بالتعبير عن المعادلات أعلاه في هيئة المصفوفات:

$$\begin{bmatrix} P1h \\ P4h \\ R1h \\ R4h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix}$$

$$Ph = Mbh\ Pb$$

عليه نكتب **المعادلات لمنحنى بيريز** على النحو التالي:

$$P(t) = T\ Mh\ Mbh\ Pb$$
$$P(t) = T\ Mb\ Pb$$

حيث:

$$M_b = \begin{bmatrix} -1 & 3 & 3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

وتكون **دوال أساس بيريز** كالآتي:

$$P(t) = T\ Mb\ Gb$$
$$= f1(t)\ P1 + f2(t)\ P2 + f3(t)\ P3 + f4(t)\ P4$$

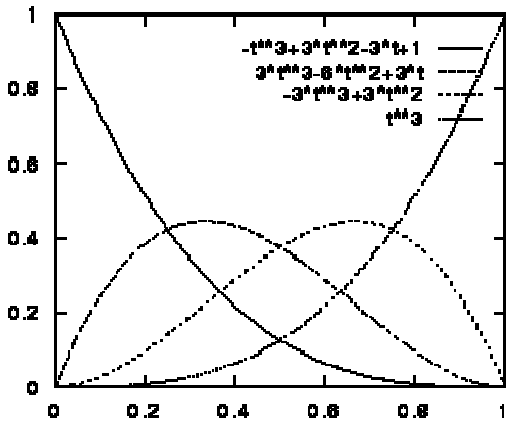
حيث:

$f1(t) = -t3 + 3t2 - 3t + 1$
$f2(t) = 3t3 - 6t2 + 3t$
$f3(t) = -3t3 + 3t2$
$f4(t) = t3$

أو بأسلوب أخر

$f1(t) = (1-t)3$
$f2(t) = 3t\ (1-t)2$
$f3(t) = 3t2\ (1-t)$
$f4(t) = t3$

تبدو دوال أساس منحنى بيريز على الهيئة التالية:



❖ خاصية الغطاء المحدب Convex hull property:

تستخدم خاصية الغطاء المحـــدب Convex hull property للتأكـــد مـــن أن المنحنى لا يتعدى أبدا الغلاف المحدب الذي ينشأ عبر رؤوس التحكم الأربعة، وتستوفى دوال أساس بيزير خاصية الغرف المحدب تحديدا:

$0 <= fi(t) <= 1$ for t in [0,1].
$f1(t) + ... + fn(t) = 1$

4. منحنيات بيزير من الدرجة
Bezier Curves of degree n n

باستخدام **معادلة بيرنستيين المتعددة الحدود** Bernsteinn polynomials يمكننا إنشاء منحنى بيزير بدرجة اعتباطية لمنحيات الدرجات الأعلى من منحنى بيزير التي نفشت حتى الآن، نحتاج إلى المزيد من نقاط التحكم. يعطى منحنى بيزير من **الدرجة n** كالاتى:

$$P(t) = \sum_{i=0}^{n} B_{i,n}(t) P_i$$
$$\text{where } B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

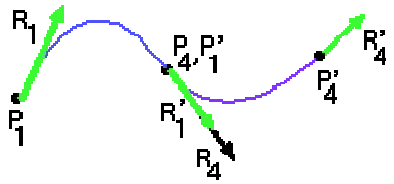
دوال الأساس متكافئة مع الحدود التي تنتج من تمديد

$$[(1-t)+t]^n$$

وذلك باستخدام **نظرية التمديد الثنائي الحد** binomial expansion theorem، ومن الواضح هنا سبب كون مجموع الحدود يساوى واحد.

5. منحنيات هيرميت وبيزير المتقطعة
Piecewise Hermite and Bezier Curves

مقاطع segments منحنى هيرميت **يمكن ترابطها** في هيئة مستمرة من **نقطة نهاية** منحنى هي **نقطه بداية** المنحنى الأخر كما **يجب التأكد** من أن متجهات المماس في هذه النقطة لديها نفس الاتجاه.



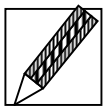
بدلالة الشكل أعلاه، هذا يعنى أن $P_1'=P_4$ و $R_1'=k\ R_4$. بالنسبة منحنى بيزيـــر الشرط هو **تتطابق آخر نقطة** في المنحنى **مع أول نقطة** في المنحنى **الثاني**.

أسئلة تقويم ذاتي



(1) قارن بين منحنيات هيرميت **و**منحنيات بيزير.
(2) تحدث باختصار عن منحنيات هيرميت وبيزير المتقطعة.

تدريب(2)



تحصل على **مصفوفة الأساس لمنحنى هيرميت** بالتعبير عن المعادلات أدناه في هيئة المصفوفات

P1h = P1
P4h = P4
R1h = 3 (P2 - P1)
4h = 3 (P4 - P3

6. الاستمرارية الهندسية والبارامترية Geometric and Parametric Continuity:

❖ **تعطي الاستمرارية الهندسية بالتالي :**

- G_o: المنحنيات المرتبطة.
- G_1: المشتقات الأولى تناسبية عند نقاط الارتباط، مماسات المنحيات لديها نفس الاتجاه ولكن ليس بالضرورة نفس المرتبة magnitude.
- مثلا: $C_1'(1) = (a,b,c)$ and $C_2'(0) = (k*a, k*b, k*c)$.
- G_2: تتناسب المشتقات الأولى والثانية عند نقاط الارتباط.

❖ **تعطى الاستمرارية البارامترية بالتالي:**

- C_0: المنحنيات المرتبطة.

- C_1: المشتقات الأولى متساوية.
- C_2: المشتقات الأولى والثانية متساوية.
- إن اعتبرت t تمثل الزمن فإن يعنى هذا أن التسارع مستمر.
- C_n: المشتقات من الدرجة n متساوية.

ويلاحظ هنا إن الاستمرارية البارامترية من الدرجة n تؤكد الاستمرارية الهندسية من الدرجة n ولكن العكس غير صحيح.

❖ **باعتبار منحنيات هيريت وبيزير نجد أنها:**

- نعطى تحكماً محلياً.
- تعطي استمرارية C_1.
- تستكمل بعض نقاط التحكم.

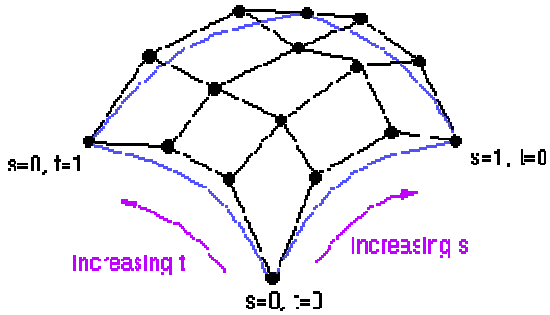
أسئلة تقويم ذاتي

(1) وضح كيف تعطى كل من الاستمرارية الهندسة ولاستمرارية البارامترية

(2) هل الاستمرارية الهندسية من الدرجة n تؤكد الاستمرارية البارامترية من الدرجة n؟

7. الأسطح البارامترية Parametric Surfaces

طريقة بسيطة لتعميم المنحنيات البارامترية إلى السطح البارامترية وهي **السماح لنقاط التحكم بلتغير لمجموعة من المنحنيات البارامترية**، والمثال التالي يوضح **إنشـــاء رقعة بيزير بهذا الأسلوب.**



إن أخذنا مقطعاً من منحنى بيزير يعرف مثلا بالتالي:

```
P(s) = S M G_bez,
   S = [ s^3 s^2 s 1 ],  s in [0,1]
```

يمكننا الاختيار بأن نجعل عناصر G_bez هي منحنيات بيزيز

$$G_bez = \begin{bmatrix} P1(t) \\ P2(t) \\ P3(t) \\ P4(t) \end{bmatrix}$$

حيث:

```
P1(t) = T M G1
P2(t) = T M G2
P3(t) = T M G3
P4(t) = T M G4
```

كل من G1 و G2 و G3 و G4 تمثل المتجهات الهندسية التي تحدد المنحنى التكعيبي البارامتيرى بدالة t لواحد من نقاط التحكم الأربعة للمنحنى الأساسي (الأصلي) P(S). عليه توجد ستة عشر قيمة نحتاجها لتحديد كل من x(s,t) و y(s,t) و z(s,t). ويكون من المفيد لإيجاد أسلوب لكتابة p(S) هيئة مدموجة ومناسبة. ويمكن عمل هذا كالآتي: أولا حول المعادلات من P1(t) ... P4(t) لإنتاج ما يلي:

```
P1(t) = G1^t  M^t  T^t
P2(t) = G2^t  M^t  T^t
P3(t) = G3^t  M^t  T^t
P4(t) = G4^t  M^t  T^t
```

ثم عوض هذه المعادلات في المعادلة الأصلية للدالة P(S)

$$P(s) = S\,M\,G$$

$$= S\,M \begin{bmatrix} P1(t) \\ P2(t) \\ P3(t) \\ P4(t) \end{bmatrix}$$

$$= S\,M \begin{bmatrix} G1\hat{}\,t & M\hat{}\,t & T\hat{}\,t \\ G2\hat{}\,t & M\hat{}\,t & T\hat{}\,t \\ G3\hat{}\,t & M\hat{}\,t & T\hat{}\,t \\ G4\hat{}\,t & M\hat{}\,t & T\hat{}\,t \end{bmatrix}$$

$$= S\,M \begin{bmatrix} g11 & g12 & g13 & g14 \\ g21 & g22 & g23 & g24 \\ g31 & g32 & g33 & g34 \\ g41 & g42 & g43 & g44 \end{bmatrix} M\hat{}\,t\ T\hat{}\,t$$

$$= S\,M\,G_tensor\ M\hat{}t\ T\hat{}t$$

هذا يعرف حاصل ضرب تعميم المتجهة tensor product لتكوين السطح، وبالنسبة للمنحنيات البارامترية تمثل متجهات وعاديه يمكن كتابتها في ثلاث معادلات منفصلة وهى:

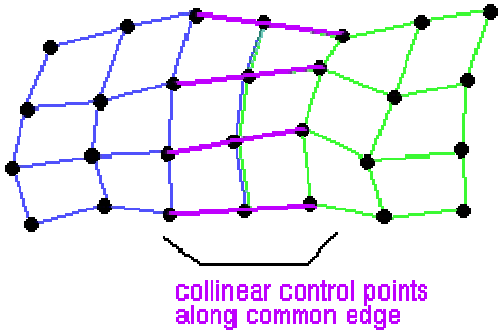
$$x(s,t) = S\,M\,G_x\,M\hat{}t\ T\hat{}t$$
$$y(s,t) = S\,M\,G_y\,M\hat{}t\ T\hat{}t$$
$$z(s,t) = S\,M\,G_z\,M\hat{}t\ T\hat{}t$$
$$s,t \text{ in } [0,1]$$

كل من G_x و G_y و G_z فى المعادلات أعلاه مصفوفة قيم متدرجة scalar حجمها 4×4

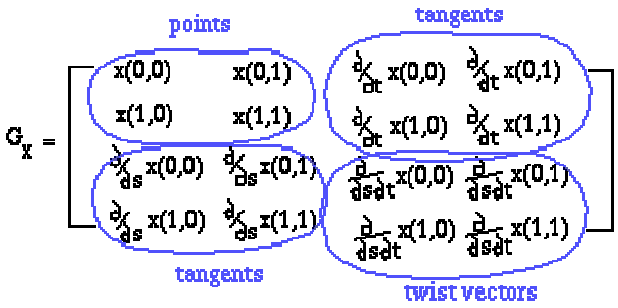
❖ خواص سطح بيزير:

يكون لسطح بيزير الخواص التالية:

- G لديها ست عشر نقطة تحكم.
- تصح خاصية الغرف المحدب.
- تعطى استمرارية G1: تجعل مجموعتين من أربع نقاط تحكم في كل جانب من الحافة شبة الخطية.



الشكل أدناه يوضح سطح هيرميت



أسئلة تقويم ذاتي

(1) كيف يتم تعميم المنحنيات البارامترية إلى السطح البارامترية؟

(2) ما هي خواص سطح بيزير ؟

الخلاصة

عزيزي الدارس، توصلنا من خلال دراستنا لهذه الوحدة إلى أنه **يمكـــن** تمثيـــل المنحنيات والأسطح بطرق مباشرة أو غيـــر مباشـــرة أو باســـتخدام تمثيـــل وســـيطة (بارامترية)، كما أن المنحنيات المكعبة تستخدم بكثرة في التخطيط نسبة لعـــدم مرونـــة منحنيات الترتيب الأدنى ومنحنيات الترتيب الأعلى، و دوال الأساس أو الركائز تعطـــى الأوزان للمكونات المختلفة للمتجه الهندسي، وأن منحنيات بيزير التي تحدد بأربع نقـــاط تعتبر متغيراً من منحنيات هيرميت التي تحدد بنقطتين ومماسين متجهين ، وخلصنا إلـــى إن الاستمرارية البارامترية من الدرجة n تؤكد الاســـتمرارية الهندســـية مـــن الدرجـــة n.ويمكن أيضاً تعميم المنحنيات البارامترية إلى السطح البارامترية بالسماح لنقاط التحكم للتغير لمجموعة من المنحنيات البارامترية.

لمحة مسبقة عن الوحدة التالية

سوف تستعرض في الوحدة التالية بعض الموضوعات المتعلقة بتخطيطات الحاسوب مثل نظرية أخذ العينات وتمثيل الألوان.

إجابات التدريبات

تدريب (1)

معادلة الأساس للمنحنى التكعيبي هي:

$$x(t) = [\ f1(t)\ f2(t)\ f3(t)\ f4(t)\ ]\ Gx$$

تدريب(2)

مصفوفة الأساس لمنحنى هيرميت هي:

$$\begin{bmatrix} P1h \\ P4h \\ R1h \\ R4h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \end{bmatrix}$$

مسرد المصطلحات

منحنيات هيرميت: هى المنحنيات التي تحدد بنقطتين ومماسين متجهين.

منحنيات بيزير: هى منحنيات تعتبر متغيراً من منحنيات هيرميت،حيث إنها **تحدد** بأربع نقاط.

دوال الأساس: تمثل الأوزان للحدود في المتجه الهندسي.

الأسطح البارامترية : طريقة بسيطة لتعميم المنحنيات البارامترية (الوسيطة)إلى ألسطح البارامترية (الوسيطة) بالسماح لنقاط التحكم للتغير لمجموعة من المنحنيات البارامترية

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة التاسعة
مواضيع مختارة
Selected Topics

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**221**
تمهيد	221
أهداف الوحدة	**221**
1. أخذ العينات Sampling	**222**
2.تمثيل الألوان Colour Representation	**225**
الخلاصة	**232**
لمحة مسبقة عن الوحدة الدراسية التالية	**232**
إجابات التدريبات	**233**
مسرد المصطلحات	**236**
المراجع	**237**

المقدمة

تمهيد

مرحباً بك إلى الوحدة التاسعة التي اخترنا لـــك فيهـــا بعـــض الموضـــوعات المتعلقـــة بتخطيطات الحاسوب مثل نظرية أخذ العينات وتمثيل الألوان.

تجد عزيزي الدارس في هذه الوحدة أسئلة تقويم ذاتي للمراجعة وأمثلة للشرح والتطبيـــق وكذلك وضعنا تدريبات مع أجوبتها في نهاية الوحدة، نرجو أن تستفيد منها وأن تشاركنا في نقدها وتقويمها.

أهداف الوحدة



عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة يجب أن تكون قادراً على أن:

تشرح نظرية أخذ العينات.

تناقش تمثيل الألوان.

تمزج الألوان الثلاثة الرئيسة مثل الأحمر R والأخضر G والأزرق B لإنتاج أي طول موجة وفقاً لشدة هذه الألوان.

تنتج ألوان الطباعة بمزج الألوان.

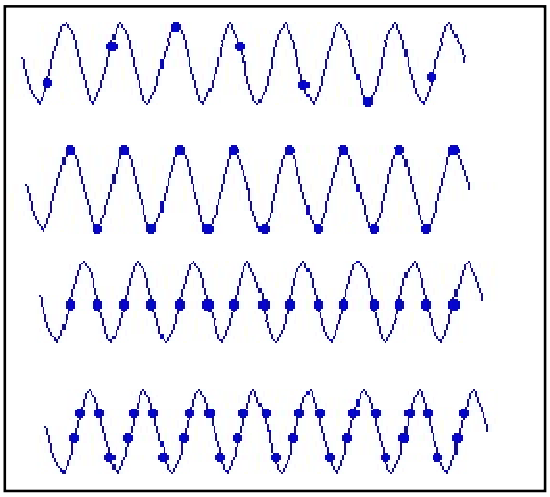
1. أخذ العينات Sampling

نص نظرية أخذ العينات:

يمكن استعادة الإشارة الزمنية المستمرة **بالكامل** من عيناتها **إذا كان** معدل أخذ العينات أكبر بما **يقل عن مرتين** عن أكبر تردد بالإشارة.

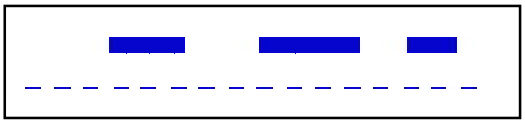
تعرف هذه النظرية بنظرية نيكيوست Nyquist أو شانون Shannon. **ويدعى** أدنى معدل أخذ عينات لاستعادة الإشارة كاملة **بمعدل نيكويست** ***Nyquist rate***.

مثال



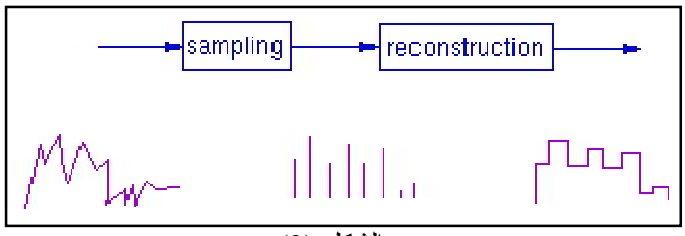
الشكل (1)

مثال



الشكل (2)

الشكل أدناه يوضح عمليات أخذ العينات وإعادة الإنشاء



الشكل (3)

❖ **استراتيجيات تقليل التداخل بين الإشارات:**

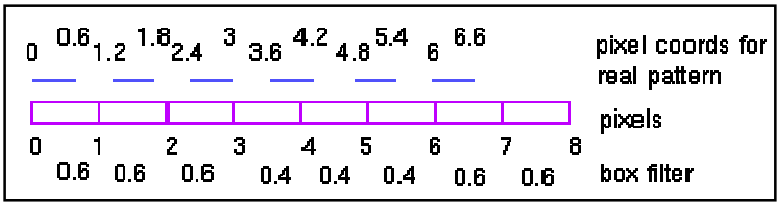
الاستراتيجيات التي تستخدم عادة لتقليل التداخل بين الإشارات هي

- الترشيح المبكر prefilter.
- الترشيح المؤجل postfilter.

• **الترشيح المبكر prefilter:**

يمثل الترشيح المبكر **أخذ عينات من مساحة دون أوزان، ويتصف** بما يلي:

- **استخدام** متوسط شدة مساحة مربع عنصر الصورة.
- **يستكمل** بين الكائن والخلفية عند الحافات.
- **رغم** كفاءته إلا أنَّ لديه بعض المشاكل.
- **ترشيح** صندوق 'box filter'.



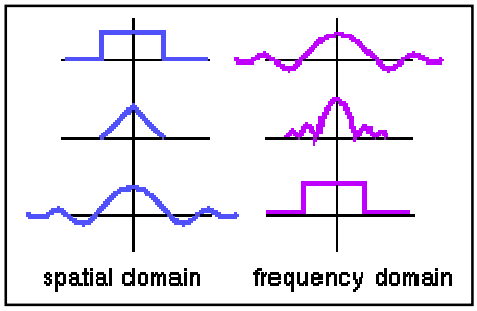
الشكل (4)

تردد أخذ العينات: f

تردد الإشارة: 5f/6

تردد الخرج: f/6

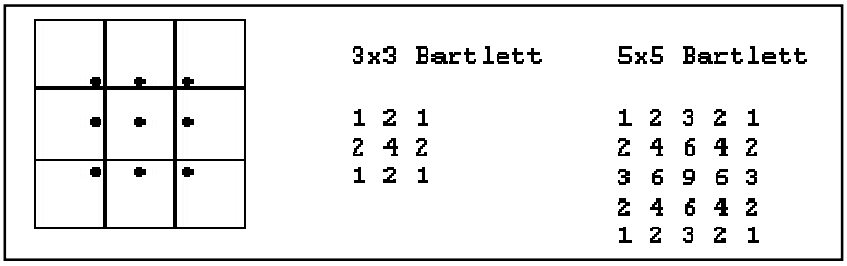
الشكل أدناه يوضح **عملية الترشيح في نطاق المكان spatial والتردد frequency** .



الشكل (5)

- **الترشيح المؤجل:**

يأخذ في الترشيح المؤجل عدداً من العينات لكل عنصر صورة كما هو موضح أدناه.



الشكل (6)

أسئلة التقويم ذاتي

(1) علام تنص نظرية أخذ العينات؟

(2) ما هي الإستراتيجيات التي تستخدم عادة لتقليل التداخل بين الإشارات؟

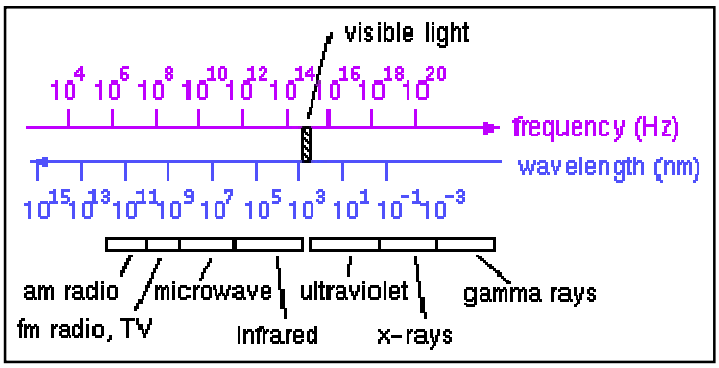
(3) صف الترشيح المبكر

تدريب(1)

وضح بالرسم عملية الترشيح في نطاق المكان والتردد

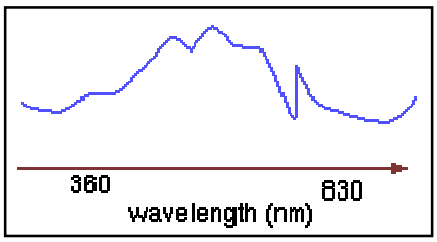
2. تمثيل الألوان Colour Representation

يتكون الضوء المرئي من **طيف(لون)** من **الطاقة الكهرومغناطيسية** لديها أطوال موجات في المدى 400–700 nm طيف الموجات الكهرومغناطيسية موضح في الشكل أدناه:



الشكل (7)

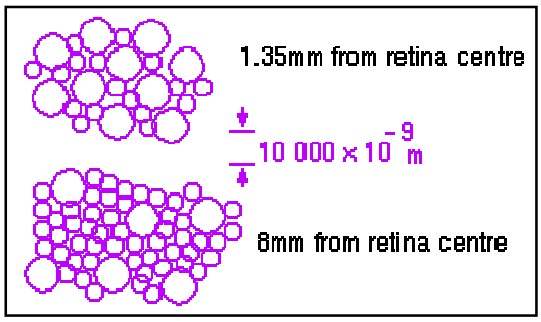
يوضح الشكل أدناه طيف الطاقة الناتجة من انعكاس الضوء من سطح أخضر.



الشكل (8)

❖ **شبكية العين:**

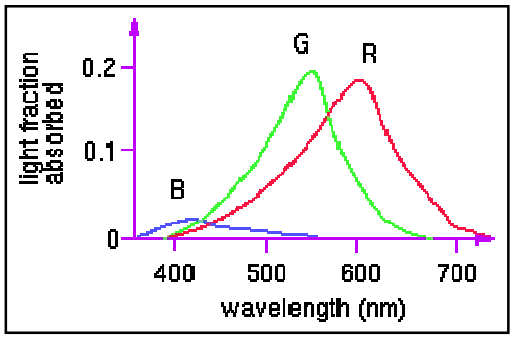
تحتوى شبكية العين على **قضبان *rods*** **وإشكال مخروطية *cones*** كما هو موضح أدناه، **ألأجسام المخروطية هي المسئولة عن إدراك الألوان.**



الشكل (9)

- **أنواع المخروطيات:**

توجد ثلاثة أنواع من المخروطيات، **يشار إليها بالأحرف B و G و R أو مكافئة للأحرف L و M و S** بالتوالي. و**قمة الحساسية** تكون عند حوالي 430nm و 560nm و 610nm لدى المشاهد العادي. **تثار** القضبان والمخروطيات **بامتصاص الضوء** والذي يحدث **تغيرات** في جهد غشاء الخلية.

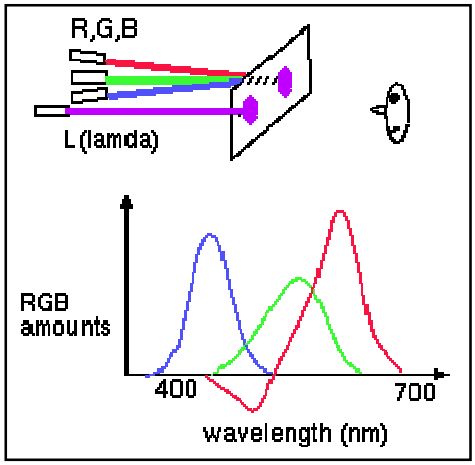


الشكل (10)

- **إدراك الألوان:**

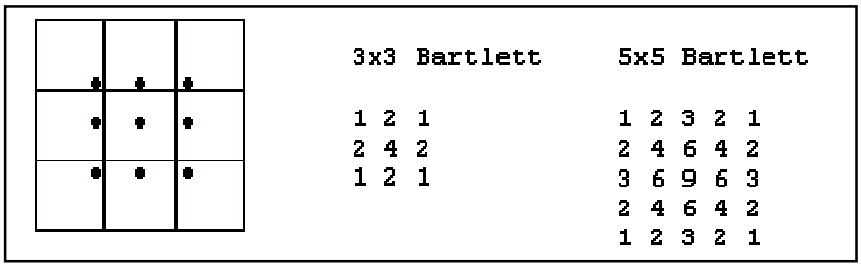
إن إدراكنا للألوان **مرتبط بمحاكاة** ثلاثة أنواع من المخروطيات، **حيث** يتم إدراكنا للفضاء الثلاثي الأبعاد **بمزج ثلاثة ألوان رئيسة** مثل **الأحمر** R (طول

الموجة، λ = 700 nm) و**الأخضر** G (طول الموجة، λ = 546 nm) و**الأزرق** B (طول الموجة، λ = 436 nm)، وبهذه الطريقة يمكننا **إنتاج** إي طول موجة باستخدام مزيج محدد من R G B. و قد **تحدث مشكلة** مثلا لا بد من **إضافة اللون الأحمر** للهدف قبل الحصول على المزيج المطلوب، ويوضح هذا في الرسم البياني بجعل **شدة اللون الأحمر R** (في هذا المثال) تأخذ **قيمة سالبة**.



الشكل (11)

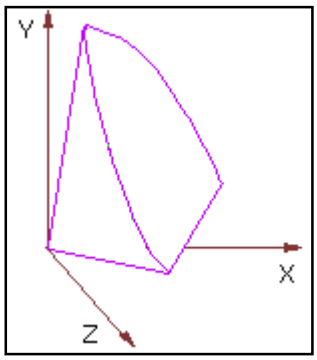
للحصول على التمثيل باستخدام **معامل مزج موجبة** فقط، **حددت** CIE ("Commission Internationale d'Eclairage") ثلاثة **مصادر ضوء افتراضية** جديدة **تعرف بالأحرف x و y و z** . تنتج هذه المصادر **منحنيات ملائمة matching curves موجبة**.



الشكل (12)

- **إيجاد الكميات X و Y و Z المقابلة للون ما:**

لنفترض الآن أنه لدينا طيف (لون) ونـــود إيجـــاد الكميـــات X و Y و Z المقابلة لهذا اللون، فيمكننا إيجاد هذه الكميات بتكامل حاصل ضرب القدرة الطيفيـــة spectral power مع كل من منحنيات الملائمة الثلاثـــة لكـــل أطـــوال الموجـــات wavelengths. أوزان X و Y و Z لفضاء XYZ المحدد فـــي CIE موضـــحة أدناه.



الشكل (13)

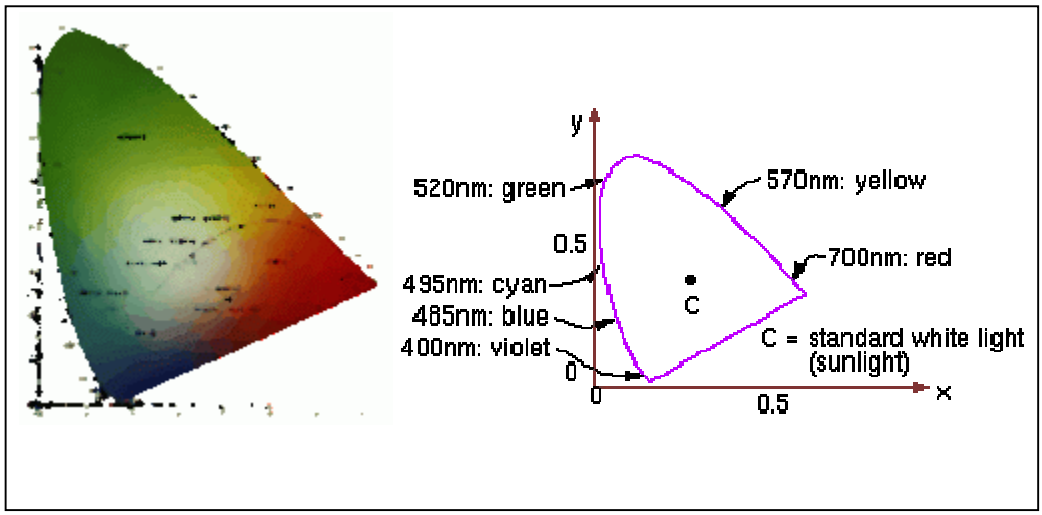
- **فضاء الألوان الثنائي الأبعاد 2D colour space:**

عادة من السهل **التعامل مع فضاء الألوان الثنـــائي الأبعـــاد** 2D colour space و **يتم** هذا في العادة **بإسقاط فضاء الألوان الثلاثـــي الأبعـــاد** 3D colour space **على السطح X+Y+Z=1 لإيجاد مخطـــط لونيـــة** CIE chromaticity diagram **تحدد الإسقاطات** بما يلي:

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} \qquad = 1 - x - y$$

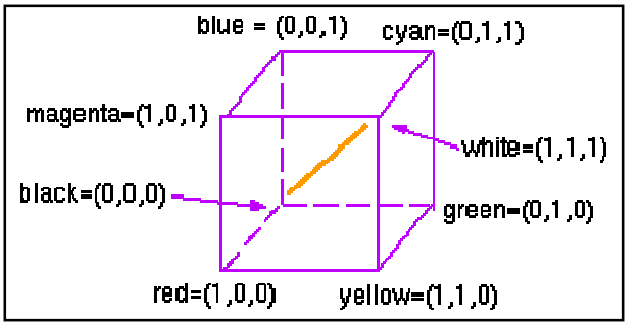
يبدو المخطط اللوني كالآتي:



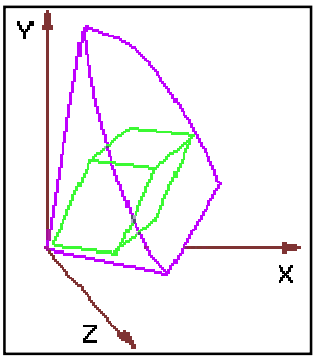
الشكل (14)

- **مكعب اللون RGB colour cube:**

نموذج الألوان الجمعي المستخدم في التخطيط بالحاسوب يمثل **بمكعب اللون RGB colour cube** ، **حيث تمثل R و G و** B الألوان **المنتجة** عن **الفسفور النقي ألأحمر والأخضر والأزرق** على التوالي.



الشكل أدناه يوضح موقع مكعب الألوان في فضاء CIE للألوان.



الشكل (15)

الجدول أدناه يلخص خواص أنواع أحبار الطباعة الأربعة الأساسية.

لون الصبغة	يمتص	يعكس
Cyan	أحمر	أزرق وأخضر
Magenta (القرمزي)	أخضر	أزرق وأحمر
أصفر	أزرق	أحمر و أخضر
أسود	الكل	لا شئ

- **إنتاج ألوان الطباعة:**

لإنتاج الأزرق، يمزج الأحبار ذات اللون cyan و اللون magenta، حيث أن كلهيما يعكسان الأزرق وأحدها يمتص الأخضر والأخر يمتص الأحمر وهذا **يجعل** عملية **تحويل** ألوان العارض **لألوان طباعة مكافئة** يجعلها عملية تحتاج إلى **مجهود وتحدِ. يستخدم** الحبر الأسود للتأكد من إمكانية إنتاج أسود بدرجة عالية، **ويشار إليه باستخدام K**. وعليه تستخدم الطابعات **نموذج CMYK**.

أسئلة تقويم ذاتي

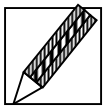


(1) ماذا نعني بتمثيل الألوان؟

(2) أين تتركز مسئولية إدراك الألوان في العين؟

(3) كيف يتم إنتاج ألوان الطباعة؟

تدريب(2)



(1) اشرح عملية إدراكنا للألوان؟

(2) كيف يتم إيجاد الكميات X و Y و Z المقابلة للون ما؟

(3) كيف يتم التعامل مع فضاء الألوان الثنائي الأبعاد؟

(4) رتب القائمة(أ) مع القائمة (ب) و(ج)

لون الصبغة	يمتص	يعكس
Cyan	الكل	أزرق وأخضر
Magenta (القرمزي)	أزرق	أحمر و أخضر
أصفر	أحمر	لا شى
أسود	أخضر	أزرق وأحمر

الخلاصة

اشتملت الوحدة علـــى تعلـــم مهـــارات أخـــذ العينـــات Sampling باســـتخدم استراتيجيات الترشيح المبكر prefilter، والترشيح المؤجل postfilter التي عادة تقلـــل التداخل بين الإشارات، كما شرحنا أن الضوء المرئي **يتكون** من **طيف(لون)** من **الطاقـــة الكهرومغناطيسية** لديها أطوال موجات في المدى 400–700 nm التي تمثل الألوان.

وأيضاً عرفنا أن إدراكنا للألوان **مرتبط بمحاكاة** ثلاثة أنواع من المخروطيـــات. **حيث** يتم إدراكنا للفضاء الثلاثي الأبعاد **بمزج ثلاثة ألـــوان رئيســـة** مثـــل **الأحمـــر** R و**الأخضر** G و**الأزرق** B ،ويمكننا **إنتاج** إي طول موجة باستخدام مزيج محدد من R G B، وتعلمنا كيفية إنتاج ألوان الطباعة بمزج الألوان.

لمحة مسبقة عن الوحدة التالية

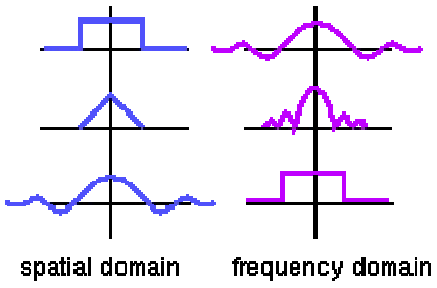
من خلال دراستنا للوحدة الثانية علمنا أن هنالك عدداً من المكتبات المرئية التي توفر ابتدائيات الرسم لتكوين المشاهد المختلفة، أكثر هذه المكتبات استخداما على مستوى الحاسوب الشخصي هي مكتبة OpenGL، وفي الوحدة العاشرة التاليـــة ســـوف نطبـــق بعض الأمثلة عملياً على الحاسوب لتكتسب مهارة تخطيطات الحاسوب باســـتخدام لغـــة البرمجة C .

إجابات التدريبات

تدريب (1)

الحل:

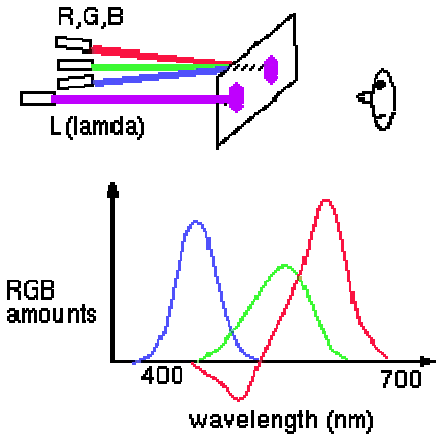
الشكل أدناه يوضح **عملية الترشيح في نطاق المكان spatial** و**التردد frequency** .



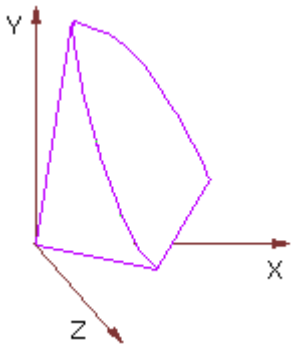
تدريب (2)

الحل:

(1) إدراكنا للألوان **مرتبط بمحاكـــاة** ثلاثة أنواع من المخروطيـــات. **حيـــث** يـــتم إدراكنا للفضاء الثلاثي الأبعاد **بمزج ثلاثة ألوان رئيسة** مثل **الأحمر** R (طول الموجة، λ = 700 nm) و**الأخضر** G (طول الموجـــة، λ = 546 nm) و**الأزرق** B (طـــول الموجة، λ = 436 nm)، وبهذه الطريقة يمكننا **إنتاج** إي طول موجة باستخدام مـــزيج محدد من R G B و قد **تحدث مشكلة** مثلا لا بد من **إضافة اللون الأحمر** للهدف قبـــل الحصول على المزيج المطلوب، يوضح هذا في الرسم البياني بجعل **شدة اللون الأحمر R** (في هذا المثال) تأخذ **قيمة سالبة**.



(2) لنفترض الآن بأنه لدينا طيف (لون) ونود إيجاد الكميات X و Y و Z المقابلة لهذا اللون. فيمكننا إيجاد هذه الكميات بتكامل حاصل ضرب القدرة الطيفية spectral power مع كل من منحنيات الملائمة الثلاثة لكل أطوال الموجات wavelengths. أوزان X و Y و Z لفضاء XYZ المحدد في CIE موضحة أدناه.



(3) عادة من السهل **التعامل مع فضاء الألوان الثنائي الأبعاد** 2D colour space. **يتم** هذا في العادة **بإسقاط فضاء الألوان الثلاثي الأبعاد** 3D colour space **على السطح X+Y+Z=1 لإيجاد مخطط لونية CIE** CIE chromaticity diagram **تحدد الإسقاطات** بالتالي:

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} \quad = 1 - x - y$$

(4)

لون الصبغة	يمتص	يعكس
Cyan	أحمر	أزرق وأخضر
Magenta (القرمزي)	أخضر	أزرق وأحمر
أصفر	أزرق	أحمر و أخضر
أسود	الكل	لا شئ

مسرد المصطلحات

- **الترشيح المبكر prefilter:**

  يمثل أخذ عينات من مساحة دون أوزان.

- **الترشيح المؤجل:**

  هو أخذ عدد من العينات لكل عنصر صورة.

- **تمثيل الألوان Colour Representation:**

  يتكون الضوء المرئي من **طيف(لون)** من **الطاقة الكهرومغناطيسية** لديها أطوال موجات في المدى 400–700 nm. طيف الموجات الكهرومغناطيسية

- **إدراك الألوان:**

هو مزج ثلاثة ألوان رئيسة مثل **الأحمر** R (طول الموجة، λ = 700 nm) و**الأخضر** G (طول الموجة، λ = 546 nm) و**الأزرق** B (طول الموجة، λ = 436 nm).

- **مكعب اللون RGB colour cube:**

  نموذج الألوان الجمعي المستخدم في التخطيط بالحاسوب.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983


الوحدة العاشرة
أمثلة تطبيقية لاستخدام مكتبة
التخطيط OpenGL

محتويات الوحدة

الموضوع	الصفحة
المقدمة	**241**
تمهيد	**241**
أهداف الوحدة	**241**
1. أمثلة لبرمجيات OpenGL	**242**
2. برنامج SimpleDraw	**243**
3. برنامج SimpleAmin	**251**
4. برنامج Solar	**260**
لمحة مسبقة عن الوحدة الدراسية التالية	**268**
المراجع	**270**

المقدمة

تمهيد

عزيزي الدارس في ختام هذا المقرر سنعطيك بعض الأمثلة هذه الوحدة لاستخدام مكونات مكتبة OpenGL مع لغة البرمجة C، وكما ذكر في الوحدة الثانية فإنه توجد عدد من مكتبات المرئي التي توفر ابتدائيات رسم لتكوين المشاهد المختلفة، و أكثر هذه المكتبات استخداما على مستوى الحاسوب الشخصي هي مكتبة OpenGL، **وسوف تعطيك ملفات المصدر source files، والترويسة header files لكل مثال،** فيمكنك أخي للدارس **تطبيق** هذه الأمثلة على جهاز حاسوب شخصي يستخدم **نظام التشغيل** يونكس/لينكس Unix/Linux أو Windows لتصقل معرفتك بتخطيطات الحاسوب.

أهداف الوحدة



عزيزي الدارس! بعد فراغك من دراسة هذه الوحدة يتوقع منك أن تكون قادراً على أن:

تطبق البرامج على مكتبة التخطيط OpenGL مع لغة البرمجة C.

1. أمثلة لبرمجيات OpenGL

❖ **استخدام مكونات مكتبة OpenGL مع لغة البرمجة C:**

يمكن ترجمة ملفات المصدر **باستخدام مترجمات البرامج cc أو gcc** لمستخدمي اليونكس أو **Visual Studio C++ version 6.0** لمستخدمي windows، وفي كل الحالات فإن **تنفيذ** هذه التطبيقات يحتاج إلى **تنصيب مكتبة OpenGL و برامج GLUT** في الجهاز (ملفات الترويسة للترجمة وكائن المكتبة لتنفيذ البرنامج).

أدناه مثال لبرامج الترجمة والربط *makefiles* التي قد تستخدم لترجمة وربط برامج المثال الأول **SamplDraw**، على نظام التشغيل يونكس أو لينكس، و**للأمثلة الأخرى اجر التعديلات الضرورية** لبرنامج makefile وذلك **بإضافة اسم برنامج المصدر c**. في سطر أمر الترجمة compiler command line بالترتيب الصحيح. وبرنامج الترجمة والربط هي:

- **برنامج الترجمة والربط للاستخدام مع برنامج الترجمةC**

```
INCDIRS = -I/usr/openwin/include -I/usr/local/include
LIBDIRS = -L/usr/openwin/lib -L/usr/local/lib

CC = gcc
CFLAGS = -g $(INCDIRS)
LIBS = -lX11 -lXi -lXmu -lglut -lGL -lGLU -lm

SimpleDraw: SimpleDraw.o
    $(CC) $(CFLAGS) -o SimpleDraw $(LIBDIRS) SimpleDraw.c
$(LIBS)
```

❖ **استخدام Visual C++:**

في **حالة استخدام Visual C++** هناك **احتمال كبير** أن تكون **مكتبة OpenGL** قد **تم تنصيبها مع تنصيب المترجم** و**الصفحة الإلكترونية الرسمية** والتي يمكن منها **تحميل أخر إصدارات البرنامج** موجودة على العنوان **http://www.xmission.com/~nate/glut.html.**، بعد التحميل **تتحصل** على الملفات glut32.dll و glut32.lib و glut.h و يجب **تنصيب** هذه البرامج على جهاز الحاسوب الشخصي، و**لمزيد من التفاصيل راجع الموقع أعلاه.**

2. برنامج SimpleDraw

❖ **برنـــامج Simple Draw** لشـــفرة **C** يوضـــح **اســـتخدام مكتبـــة التخطـــيط OpenGL. فالبرنامج يوضح طريقة**

- ♦ رسم النقاط points
- ♦ الخطوط lines
- ♦ إزالة الخطوط line strips
- ♦ حلقات الخطوط line loops.

❖ **يتكون البرنامج من ملفي مصدر هما:**

- ♦ SimpleDraw.c
- ♦ SimpleDraw.h

تدريب(1)



1. **ترجم وشغل** البرنامج، **استخدم** مسطرة المسافة space bar للتنقل بين المناظر الخمسة، و**اختبر** شفرة المصدر وأوجد كيف يتم رسم اى من هذه المناظر الخمسة في وتيرة routine ***drawScence()***.
2. **أزل** علامات التعليق uncomment من تجميعية الشفرة في ***initRendering()***، ثم أعد ترجمة وتشغيل البرنامج، (أزل التعليق من التجميعية الأولى أولا ثم من التجميعيتين). كيف يوثر هذا على النتائج؟ وكيف تبدو مناظر التخطيط على الحاسوب؟
3. **تفهم** عملية المنادة الراجعة التي تضبط بمنادة كل من ***glutKeyboardFunc(),glutReshapeFunc(), glutDisplayFunc()***.
4. **تفهم** كيف يستجيب البرنامج لمسطرة المسافة و مفتاح الخروج escape key.
5. **لا يستخدم** هذا البرنامج للتخزين الوسيط المزدوج double buffering. ما معنى هذا؟ ما هى وظيفة نداء **glutPostRedisplay()** بعد تغير الحالة؟

❖ **ملف SimpleDraw.c:**

```
/*
 * SimpleDraw.c
 *
 * Example program illustrating a simple use
 * of OpenGL to draw straight lines in 2D, and
 * to draw overlapping triangles in 3D.
 *
 * Author: Samuel R. Buss
 *
 * Software accompanying the book
```

```
 *              3D Computer Graphics: A Mathematical Introduction
with OpenGL,
 *              by S. Buss, Cambridge University Press, 2003.
 *
 * Software is "as-is" and carries no warranty.  It may be used
without
 *   restriction, but if you modify it, please change the filenames to
 *   prevent confusion between different versions.
 * Bug reports: Sam Buss, sbuss@ucsd.edu.
 * Web page: http://math.ucsd.edu/~sbuss/MathCG
 *
 * Version 1.1.  Updated September 28, 2003.
 *    1.1. 9/28/30: Added glEnable(GL_DEPTH_TEST).  Thanks to
Rob Wilkens.
 *    1.0. Original version released Spring 2003.
 *
 */

#include <stdlib.h>
#include <stdio.h>
#include <GL/glut.h>      // OpenGL Graphics Utility Library
#include "SimpleDraw.h"

// These variables control the current mode
int CurrentMode = 0;
const int NumModes = 5;

// These variables set the dimensions of the rectanglar region we
wish to view.
const double Xmin = 0.0, Xmax = 3.0;
const double Ymin = 0.0, Ymax = 3.0;

// glutKeyboardFunc is called below to set this function to handle
//              all "normal" ascii key presses.
// Only space bar and escape key have an effect.
void myKeyboardFunc( unsigned char key, int x, int y )
{
    switch ( key ) {

    case ' ':
            // Space bar
```

```
            // Increment the current mode, and tell operating system
screen needs redrawing
            CurrentMode = (CurrentMode+1)%NumModes;
            glutPostRedisplay();
            break;

    case 27:
            // "27" is theEscape key
            exit(1);

    }
}


/*
 * drawScene() handles the animation and the redrawing of the
 *             graphics window contents.
 */
void drawScene(void)
{
    // Clear the rendering window

    glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);

    // Set drawing color to white
    glColor3f( 1.0, 1.0, 1.0 );

    switch (CurrentMode)
    {

    case 0:
            // Draw three points
            glBegin(GL_POINTS);
            glVertex2f( 1.0, 1.0 );
            glVertex2f( 2.0, 1.0 );
            glVertex2f( 2.0, 2.0 );
            glEnd();
            break;

    case 1:
```

```
case 2:
case 3:
        if ( CurrentMode==1 ) {
                // Draw lines in GL_LINES mode
                glBegin( GL_LINES );
        }
        else if ( CurrentMode==2 ) {
                // Draw lines in GL_LINE_STRIP mode
                glBegin( GL_LINE_STRIP );
        }
        else {
                // Draw lines in GL_LINE_LOOP mode
                glBegin( GL_LINE_LOOP );
        }
        glVertex2f( 0.5, 1.0 );
        glVertex2f( 2.0, 2.0 );
        glVertex2f( 1.8, 2.6 );
        glVertex2f( 0.7, 2.2 );
        glVertex2f( 1.6, 1.2 );
        glVertex2f( 1.0, 0.5 );
        glEnd();
        break;

case 4:                 // Overlapping triangles
        glBegin( GL_TRIANGLES );
        glColor3f( 1.0, 0.0, 0.0 );
        glVertex3f( 0.3, 1.0, 0.5 );
        glVertex3f( 2.7, 0.85, 0.0 );
        glVertex3f( 2.7, 1.15, 0.0 );

        glColor3f( 0.0, 1.0, 0.0 );
        glVertex3f(2.53, 0.71, 0.5 );
        glVertex3f(1.46, 2.86, 0.0 );
        glVertex3f(1.2, 2.71, 0.0 );

        glColor3f( 0.0, 0.0, 1.0 );
        glVertex3f(1.667, 2.79, 0.5);
        glVertex3f(0.337, 0.786, 0.0);
        glVertex3f(0.597, 0.636, 0.0);
        glEnd();
}
```

```
    // Flush the pipeline.  (Not usually necessary.)
    glFlush();

}

// Initialize OpenGL's rendering modes
void initRendering()
{

    glEnable ( GL_DEPTH_TEST );

    // Uncomment out the first block of code below, and then the
second block,
    //                to see how they affect line and point drawing.
/*
    // The following commands should cause points and line to be
drawn larger
    //        than a single pixel width.
    glPointSize(8);
    glLineWidth(5);
*/

/*
    // The following commands should induce OpenGL to create
round points and
    //        antialias points and lines.  (This is implementation
dependent unfortunately).
    glEnable(GL_POINT_SMOOTH);
    glEnable(GL_LINE_SMOOTH);
    glHint(GL_POINT_SMOOTH_HINT, GL_NICEST);          //
Make round points, not square points
    glHint(GL_LINE_SMOOTH_HINT, GL_NICEST);           //
Antialias the lines
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA,
GL_ONE_MINUS_SRC_ALPHA);
*/

}
```

```
// Called when the window is resized
//            w, h - width and height of the window in pixels.
void resizeWindow(int w, int h)
{
    double scale, center;
    double windowXmin, windowXmax, windowYmin, windowYmax;

    // Define the portion of the window used for OpenGL rendering.
    glViewport( 0, 0, w, h );       // View port uses whole window

    // Set up the projection view matrix: orthographic projection
    // Determine the min and max values for x and y that should
appear in the window.
    // The complication is that the aspect ratio of the window may not
match the
    //                aspect ratio of the scene we want to view.
    w = (w==0) ? 1 : w;
    h = (h==0) ? 1 : h;
    if ( (Xmax-Xmin)/w < (Ymax-Ymin)/h ) {
            scale = ((Ymax-Ymin)/h)/((Xmax-Xmin)/w);
            center = (Xmax+Xmin)/2;
            windowXmin = center - (center-Xmin)*scale;
            windowXmax = center + (Xmax-center)*scale;
            windowYmin = Ymin;
            windowYmax = Ymax;
    }
    else {
            scale = ((Xmax-Xmin)/w)/((Ymax-Ymin)/h);
            center = (Ymax+Ymin)/2;
            windowYmin = center - (center-Ymin)*scale;
            windowYmax = center + (Ymax-center)*scale;
            windowXmin = Xmin;
            windowXmax = Xmax;
    }
    // Now that we know the max & min values for x & y that should
be visible in the window,
    //                we set up the orthographic projection.
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    glOrtho( windowXmin, windowXmax, windowYmin,
windowYmax, -1, 1 );
```

```
}

// Main routine
// Set up OpenGL, define the callbacks and start the main loop
int main( int argc, char** argv )
{
    glutInit(&argc,argv);

    // The image is not animated so single buffering is OK.
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |
GLUT_DEPTH );

    // Window position (from top corner), and size (width and hieght)
    glutInitWindowPosition( 20, 60 );
    glutInitWindowSize( 360, 360 );
    glutCreateWindow( "SimpleDraw - Press space bar to toggle
images" );

    // Initialize OpenGL as we like it..
    initRendering();

    // Set up callback functions for key presses
    glutKeyboardFunc( myKeyboardFunc );                    //
Handles "normal" ascii symbols
    // glutSpecialFunc( mySpecialKeyFunc );              // Handles
"special" keyboard keys

    // Set up the callback function for resizing windows
    glutReshapeFunc( resizeWindow );

    // Call this for background processing
    // glutIdleFunc( myIdleFunction );

    // call this whenever window needs redrawing
    glutDisplayFunc( drawScene );

    fprintf(stdout, "Press space bar to toggle images; escape button
to quit.\n");

    // Start the main loop.  glutMainLoop never returns.
```

```
    glutMainLoop( );

    return(0);        // This line is never reached.
}
```

❖ **ملف SimpleDraw.h:**

```
/*
 * SimpleDraw.h
 *
 * Author: Samuel R. Buss
 *
  */

// Function prototypes for SimpleDraw.c

void myKeyboardFunc( unsigned char key, int x, int y );

void drawScene(void);

void initRendering();
void resizeWindow(int w, int h);
```

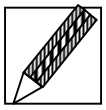
3. برنامج SimpleAmin

برنامج SimpleAnim **بشفرة C يوضح** استخدام مكتبة التخطيط OpenGL.

وهو يقوم بدورات ثلاثة مثلثات متشابكة overlapping. ثم إنه يتكون من ملفي مصدرهما:

- SimpleAmin.c ♦
- SimpleAmin.h ♦

تدريب(2)



1. **ترجم وشغل** البرنامج، **حـــاول** متحكمـــات لوحـــة المفـــاتيح keyboard controls. **بالضغط على "r"** يبدأ وينتهـــي تحريـــك الرســـم animation. **بالضغط على "s"** تتحصل على خطوة واحدة من الحركة، **الأســـهم** ألا علـــى والأسفل تتحكم في سرعة الدوران.
2. **لاحظ مشكلة** الاسم المستعار aliasing مع حافات المثلثات، لعمل هذا كبر النافذة لمشاهدة أفضل، ثم قلل سرعة الدوران حتى تلف المثلثات ببطء. علـــى حافات المثلثات ترى حافات خشنة غير سوية أو حركة عناصر الصورة.
3. هذا البرنامج **يستخدم** التخزين الوســـيط المـــزدوج double buffering **لتحسين** الحركة، اختبر هذا بجعل البرنامج يستخدم مصداً واحـــداً. لاســـتخدام وصد واحد غير العامل **GLUT_DOUBLE** إلـــى **GLUT_SINGLE** وأضف علامة التعليق comment مناداة **glutSwapBuffers()**. كيف تبدو الحركة في حالة استخدام مصد واحد
4. **بالقرب من بداية الوتيرة** drawScence() الزاوية الحالية مثبتة على أقل من 360^0 ما هو **الخطاء** الذي سوف يحدث إن أزيلت من الشفرة. **حاول** إجراء هذا التغير وحدد الخطاء. (للملاحظة بسرعة اجعل المحاكي يعمل بوتيرة عالية لفترة زمنية قصيرة ثم يعمل بوتيرة منخفضة، قد تحتاج للتشغيل لفترة طويلـــة حتى تصبح أخطـــاء الفيضـــان/الانحســـار overflow/underflow errors ملموسة.
5. **تفهم** عمل كل من **glRotatef** and **glTranslatef**.

❖ ملف SimpleAmin.c:

```
/*
 * SimpleAnim.c
 *
 * Example program illustrating a simple use
 * of OpenGL to animate motion.  Creates rotating
 * overlapping triangles.
 *
 * Author: Samuel R. Buss
 *
 * Software accompanying the book
 *            3D Computer Graphics: A Mathematical Introduction
with OpenGL,
 *            by S. Buss, Cambridge University Press, 2003.
 *
 * Software is "as-is" and carries no warranty.  It may be used
without
 *   restriction, but if you modify it, please change the filenames to
 *   prevent confusion between different versions.
 * Bug reports: Sam Buss, sbuss@ucsd.edu.
 * Web page: http://math.ucsd.edu/~sbuss/MathCG
 *
 * USAGE:
 *    Press "r" key to toggle (off and on) running the animation
 *    Press "s" key to single-step animation
 *    The up arrow key and down array key control the
 *                    time step used in the animation rate.  Each key
 *                    press multiplies or divides the times by a factor
 *                    of sqrt(2).
 *     Press ESCAPE to exit.
 *
 */

#include <math.h>                    // For math routines (such as sqrt
& trig).
#include <stdio.h>
#include <stdlib.h>                  // For the "exit" function
#include <GL/glut.h>      // OpenGL Graphics Utility Library
#include "SimpleAnim.h"
```

```
int RunMode = 1;            // Used as a boolean (1 or 0) for "on" and
"off"

// The next global variable controls the animation's state and speed.
float CurrentAngle = 0.0f;                  // Angle in degrees
float AnimateStep = 3.0f;                   // Rotation step per update

// These variables set the dimensions of the rectanglar region we
wish to view.
const double Xmin = 0.0, Xmax = 3.0;
const double Ymin = 0.0, Ymax = 3.0;

// glutKeyboardFunc is called below to set this function to handle
//            all "normal" key presses.
void myKeyboardFunc( unsigned char key, int x, int y )
{
    switch ( key ) {
    case 'r':
            RunMode = 1-RunMode;            // Toggle to opposite
value
            if ( RunMode==1 ) {
                    glutPostRedisplay();
            }
            break;
    case 's':
            RunMode = 1;
            drawScene();
            RunMode = 0;
            break;
    case 27:        // Escape key
            exit(1);
    }
}

// glutSpecialFunc is called below to set this function to handle
//            all "special" key presses.  See glut.h for the names of
//            special keys.
void mySpecialKeyFunc( int key, int x, int y )
{
    switch ( key ) {
```

```
    case GLUT_KEY_UP:
            if ( AnimateStep < 1.0e3) {                 // Avoid
overflow problems
                    AnimateStep *= sqrt(2.0);           // Increase
the angle increment
            }
            break;
    case GLUT_KEY_DOWN:
            if (AnimateStep>1.0e-6) {           // Avoid underflow
problems.
                    AnimateStep /= sqrt(2.0);   // Decrease the
angle increment
            }
            break;
    }
}

/*
 * drawScene() handles the animation and the redrawing of the
 *             graphics window contents.
 */
void drawScene(void)
{
    // Clear the rendering window
   glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);

   if (RunMode==1) {
            // Calculate animation parameters
      CurrentAngle += AnimateStep;
            if ( CurrentAngle > 360.0 ) {
                    CurrentAngle -=
360.0*floor(CurrentAngle/360.0);        // Don't allow overflow
            }
    }

    // Rotate the image
    glMatrixMode( GL_MODELVIEW );               // Current
matrix affects objects positions
    glLoadIdentity();                                   //
Initialize to the identity
```

```
    glTranslatef( 1.5, 1.5, 0.0 );                          //
Translate rotation center from origin
    glRotatef( CurrentAngle, 0.0, 0.0, 1.0 );               // Rotate
through animation angle
    glTranslatef( -1.5, -1.5, 0.0 );                        // Translate
rotation center to origin

    // Draw three overlapping triangles of different colors
    glBegin( GL_TRIANGLES );
    glColor3f( 1.0, 0.0, 0.0 );
    glVertex3f( 0.3, 1.0, 0.5 );
    glVertex3f( 2.7, 0.85, 0.0 );
    glVertex3f( 2.7, 1.15, 0.0 );

    glColor3f( 0.0, 1.0, 0.0 );
    glVertex3f(2.53, 0.71, 0.5 );
    glVertex3f(1.46, 2.86, 0.0 );
    glVertex3f(1.2, 2.71, 0.0 );

    glColor3f( 0.0, 0.0, 1.0 );
    glVertex3f(1.667, 2.79, 0.5);
    glVertex3f(0.337, 0.786, 0.0);
    glVertex3f(0.597, 0.636, 0.0);
    glEnd();

    // Flush the pipeline, swap the buffers
  glFlush();
  glutSwapBuffers();

    if ( RunMode==1 ) {
            glutPostRedisplay();   // Trigger an automatic redraw for
animation
    }

}

// Initialize OpenGL's rendering modes
void initRendering()
{
  glShadeModel( GL_FLAT );    // The default value of
GL_SMOOTH is usually better
```

```
   glEnable( GL_DEPTH_TEST );          // Depth testing must be
turned on
}

// Called when the window is resized
//            w, h - width and height of the window in pixels.
void resizeWindow(int w, int h)
{
    double scale, center;
    double windowXmin, windowXmax, windowYmin, windowYmax;

    // Define the portion of the window used for OpenGL rendering.
    glViewport( 0, 0, w, h );          // View port uses whole window

    // Set up the projection view matrix: orthographic projection
    // Determine the min and max values for x and y that should
appear in the window.
    // The complication is that the aspect ratio of the window may not
match the
    //                aspect ratio of the scene we want to view.
    w = (w==0) ? 1 : w;
    h = (h==0) ? 1 : h;
    if ( (Xmax-Xmin)/w < (Ymax-Ymin)/h ) {
            scale = ((Ymax-Ymin)/h)/((Xmax-Xmin)/w);
            center = (Xmax+Xmin)/2;
            windowXmin = center - (center-Xmin)*scale;
            windowXmax = center + (Xmax-center)*scale;
            windowYmin = Ymin;
            windowYmax = Ymax;
    }
    else {
            scale = ((Xmax-Xmin)/w)/((Ymax-Ymin)/h);
            center = (Ymax+Ymin)/2;
            windowYmin = center - (center-Ymin)*scale;
            windowYmax = center + (Ymax-center)*scale;
            windowXmin = Xmin;
            windowXmax = Xmax;
    }

    // Now that we know the max & min values for x & y
    //                that should be visible in the window,
```

```
    //                  we set up the orthographic projection.
   glMatrixMode( GL_PROJECTION );
   glLoadIdentity();
   glOrtho( windowXmin, windowXmax, windowYmin, windowYmax,
-1, 1 );

}


// Main routine
// Set up OpenGL, define the callbacks and start the main loop
int main( int argc, char** argv )
{
     glutInit(&argc,argv);

     // We're going to animate it, so double buffer
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB |
GLUT_DEPTH );

     // Window position (from top corner), and size (width% and
hieght)
   glutInitWindowPosition( 10, 60 );
   glutInitWindowSize( 360, 360 );
   glutCreateWindow( "SimpleAnim" );

     // Initialize OpenGL as we like it..
   initRendering();

     // Set up callback functions for key presses
     glutKeyboardFunc( myKeyboardFunc );                    //
Handles "normal" ascii symbols
     glutSpecialFunc( mySpecialKeyFunc );              // Handles
"special" keyboard keys

     // Set up the callback function for resizing windows
   glutReshapeFunc( resizeWindow );

     // Call this for background processing
     // glutIdleFunc( myIdleFunction );

     // call this whenever window needs redrawing
```

```
   glutDisplayFunc( drawScene );

    fprintf(stdout, "Arrow keys control speed.  Press \"r\" to run,
\"s\" to single step.\n");

    // Start the main loop.  glutMainLoop never returns.
    glutMainLoop(  );

   return(0);          // This line is never reached.
}
```

❖ **ملف SimpleAnim.h:**

```
/*
 * SimpleAnim.h
 *
 * Author: Samuel R. Buss
 *
 * Software accompanying the book
 *             3D Computer Graphics: A Mathematical Introduction
with OpenGL,
 *             by S. Buss, Cambridge University Press, 2003.
 *
 * Software is "as-is" and carries no warranty.  It may be used
without
 *   restriction, but if you modify it, please change the filenames to
 *   prevent confusion between different versions.
 * Bug reports: Sam Buss, sbuss@ucsd.edu.
 * Web page: http://math.ucsd.edu/~sbuss/MathCG
 */

// Function prototypes for SimpleAnim.cpp

void myKeyboardFunc( unsigned char key, int x, int y );
void mySpecialKeyFunc( int key, int x, int y );

void drawScene(void);

void initRendering();
void resizeWindow(int w, int h);
/*
```

```
 * SimpleDraw.h
 *
 * Author: Samuel R. Buss
 *
 */

// Function prototypes for SimpleDraw.c

void myKeyboardFunc( unsigned char key, int x, int y );

void drawScene(void);

void initRendering();
void resizeWindow(int w, int h);
```

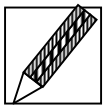
4. برنامج Solar

برنامج Solar **بشفرة C** يوضح استخدام مكتبة التخطـيط OpenGL. وهـو **يقوم** بتمثيل حركة نظام شمسي مكون من الشمس وكوكب واحد وقمر واحد.

والبرنامج يتكون من ملفي مصدرهما:

- Solar.c ♦
- Solar.h ♦

تدريب(3)



1. **ترجم وشغل** البرنامج، حاول متحكمات لوحة المفاتيح keyboard controls، في البداية البرنامج يظهر الكوكب ولكنه لا يدور ولكنه دائما يواجه الشمس، استخدام الأسهم في لوحة المفـاتيح لمشـاهدة الحركة. أوضح كيف تعمل شفرة الحركة.
2. **اجعل** البرنامج يستخدم مصدر واحد كما فعلنا في التطبيق السـابق ولاحظ التأثير.
3. **أنشئ** نظاما شمسيا أكثر تعقيدا بإضافة كواكب وأقمار أخرى.

❖ **ملف solar.c:**

```
/*
 * Solar.c
 *
 * Program to demonstrate how to use a local
 * coordinate method to position parts of a
 * model in relation to other model parts.
 *
 * Draws a simple solar system, with a sun, planet and moon.
 * Based on sample code from the OpenGL programming guide
 *          by Woo, Neider, Davis.  Addison-Wesley.
 *
 * Author: Samuel R. Buss
 *
 * Software accompanying the book
 *          3D Computer Graphics: A Mathematical Introduction
with OpenGL,
 *          by S. Buss, Cambridge University Press, 2003.
 *
 * Software is "as-is" and carries no warranty.  It may be used
without
 *   restriction, but if you modify it, please change the filenames to
 *   prevent confusion between different versions.
 * Bug reports: Sam Buss, sbuss@ucsd.edu.
 * Web page: http://math.ucsd.edu/~sbuss/MathCG
 *
 * USAGE:
 *    Press "r" key to toggle (off and on) running the animation
 *    Press "s" key to single-step animation
 *    The up arrow key and down array key control the
 *                  time step used in the animation rate.  Each key
 *                  press multiplies or divides the times by a factor
 *                  of two (2).
 *     Press ESCAPE to exit.
 *
 */


#include "Solar.h"
```

```
#include <stdlib.h>

#include <GL/glut.h>      // OpenGL Graphics Utility Library

static GLenum spinMode = GL_TRUE;
static GLenum singleStep = GL_FALSE;

// These three variables control the animation's state and speed.
static float HourOfDay = 0.0;
static float DayOfYear = 0.0;
static float AnimateIncrement = 24.0;  // Time step for animation
(hours)

// glutKeyboardFunc is called below to set this function to handle
//             all normal key presses.
static void KeyPressFunc( unsigned char Key, int x, int y )
{
    switch ( Key ) {
    case 'R':
    case 'r':
            Key_r();
            break;
    case 's':
    case 'S':
            Key_s();
            break;
    case 27:        // Escape key
            exit(1);
    }
}

// glutSpecialFunc is called below to set this function to handle
//             all special key presses.  See glut.h for the names of
//             special keys.
static void SpecialKeyFunc( int Key, int x, int y )
{
    switch ( Key ) {
    case GLUT_KEY_UP:
            Key_up();
            break;
    case GLUT_KEY_DOWN:
```

```
            Key_down();
            break;
    }
}


static void Key_r(void)
{
    if ( singleStep ) {                 // If ending single step mode
            singleStep = GL_FALSE;
            spinMode = GL_TRUE;             // Restart animation
    }
    else {
            spinMode = !spinMode;       // Toggle animation on and
off.
    }
}

static void Key_s(void)
{
    singleStep = GL_TRUE;
    spinMode = GL_TRUE;
}

static void Key_up(void)
{
   AnimateIncrement *= 2.0;                 // Double the
animation time step
}

static void Key_down(void)
{
   AnimateIncrement /= 2.0;                 // Halve the
animation time step

}

/*
 * Animate() handles the animation and the redrawing of the
 *          graphics window contents.
 */
```

```
static void Animate(void)
{
    // Clear the redering window
  glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);

  if (spinMode) {
          // Update the animation state
    HourOfDay += AnimateIncrement;
    DayOfYear += AnimateIncrement/24.0;

    HourOfDay = HourOfDay - ((int)(HourOfDay/24))*24;
    DayOfYear = DayOfYear - ((int)(DayOfYear/365))*365;
          }

    // Clear the current matrix (Modelview)
  glLoadIdentity();

    // Back off eight units to be able to view from the origin.
  glTranslatef ( 0.0, 0.0, -8.0 );

    // Rotate the plane of the elliptic
    // (rotate the model's plane about the x axis by fifteen degrees)
    glRotatef( 15.0, 1.0, 0.0, 0.0 );

  // Draw the sun -- as a yellow, wireframe sphere
    glColor3f( 1.0, 1.0, 0.0 );
  glutWireSphere( 1.0, 15, 15 );

  // Draw the Earth
    // First position it around the sun
    //          Use DayOfYear to determine its position
  glRotatef( 360.0*DayOfYear/365.0, 0.0, 1.0, 0.0 );
  glTranslatef( 4.0, 0.0, 0.0 );
  glPushMatrix();                                    // Save
matrix state
    // Second, rotate the earth on its axis.
    //          Use HourOfDay to determine its rotation.
    glRotatef( 360.0*HourOfDay/24.0, 0.0, 1.0, 0.0 );
    // Third, draw the earth as a wireframe sphere.
  glColor3f( 0.2, 0.2, 1.0 );
```

```
    glutWireSphere( 0.4, 10, 10);
    glPopMatrix();                                  // Restore
matrix state

     // Draw the moon.
     //         Use DayOfYear to control its rotation around the earth
     glRotatef( 360.0*12.0*DayOfYear/365.0, 0.0, 1.0, 0.0 );
    glTranslatef( 0.7, 0.0, 0.0 );
    glColor3f( 0.3, 0.7, 0.3 );
    glutWireSphere( 0.1, 5, 5 );

     // Flush the pipeline, and swap the buffers
    glFlush();
    glutSwapBuffers();

     if ( singleStep ) {
             spinMode = GL_FALSE;
     }

     glutPostRedisplay();            // Request a re-draw for animation
purposes

}

// Initialize OpenGL's rendering modes
void OpenGLInit(void)
{
    glShadeModel( GL_FLAT );
    glClearColor( 0.0, 0.0, 0.0, 0.0 );
    glClearDepth( 1.0 );
    glEnable( GL_DEPTH_TEST );
}

// ResizeWindow is called when the window is resized
static void ResizeWindow(int w, int h)
{
    float aspectRatio;
     h = (h == 0) ? 1 : h;
     w = (w == 0) ? 1 : w;
     glViewport( 0, 0, w, h );         // View port uses whole window
     aspectRatio = (float)w/(float)h;
```

```
    // Set up the projection view matrix (not very well!)
   glMatrixMode( GL_PROJECTION );
   glLoadIdentity();
   gluPerspective( 60.0, aspectRatio, 1.0, 30.0 );

    // Select the Modelview matrix
   glMatrixMode( GL_MODELVIEW );
}


// Main routine
// Set up OpenGL, hook up callbacks, and start the main loop
int main( int argc, char** argv )
{
    // Need to double buffer for animation
    glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB |
GLUT_DEPTH );

    // Create and position the graphics window
   glutInitWindowPosition( 0, 0 );
   glutInitWindowSize( 600, 360 );
   glutCreateWindow( "Solar System Demo" );

    // Initialize OpenGL.
   OpenGLInit();

    // Set up callback functions for key presses
    glutKeyboardFunc( KeyPressFunc );
    glutSpecialFunc( SpecialKeyFunc );

    // Set up the callback function for resizing windows
   glutReshapeFunc( ResizeWindow );

    // Callback for graphics image redrawing
   glutDisplayFunc( Animate );

    // Start the main loop.  glutMainLoop never returns.
    glutMainLoop( );
```

```
   return(0);                    // Compiler requires this to be here.
(Never reached)
}
```

❖ **ملف Solar.h:**

```
/*
 * Solar.h
 *
 * Author: Samuel R. Buss
 *
 * Software accompanying the book
 *             3D Computer Graphics: A Mathematical Introduction
with OpenGL,
 *             by S. Buss, Cambridge University Press, 2003.
 *
 * Software is "as-is" and carries no warranty.  It may be used
without
 *   restriction, but if you modify it, please change the filenames to
 *   prevent confusion between different versions.
 * Bug reports: Sam Buss, sbuss@ucsd.edu.
 * Web page: http://math.ucsd.edu/~sbuss/MathCG
 */

void OpenGLInit(void);

static void Animate(void );
static void Key_r(void );
static void Key_s(void );
static void Key_up(void );
static void Key_down(void );
static void ResizeWindow(int w, int h);

static void KeyPressFunc( unsigned char Key, int x, int y );
static void SpecialKeyFunc( int Key, int x, int y );
```

الخلاصة

عزيزي الدارس خلصنا من دراسة هذه الوحدة إلى أن هناك برامج SimpleAnim،SampleDraw وSolar بشفرة C توضح استخدام مكتبة التخطيط OpenGL، و أن برامج الترجمة والربط makefiles تستخدم مع برامج ال SampleDraw على نظام التشغيل يونكس أو لينكس الذي يوضح طريقة رسم النقاط points، الخطوطlines، وإزالة الخطوط line strips، وحلقات الخطوط line loops،ويتكون البرنامج من ملفي مصدر هما SimpleDraw.c وSimpleDraw.h، كما أن برنامج SimpleAmin يقوم بدورات ثلاثة مثلثات متشابكة overlapping ويتكون من ملفي مصدر هما SimpleAmin.c و SimpleAmin.h.ثم تعرفنا أخيراً على برنامج Solar الذي يقوم بتمثيل حركة نظام شمسي مكون من الشمس وككوب واحد وقمر واحد. والبرنامج يتكون أيضاً من ملفي مصدر هما Solar.c و Solar.h.

Open GL

Open Graphics **L**ibrary is a specification defining a cross-language cross-platform API for writing applications that produce computer graphics, it consists of over 250 different functions which can be used to draw complex computer graphics from simple primitives. From this specification, each hardware vendor creates implementations that are used as a broker between computer graphics software and hardware through operating system. So, Open GL guarantees that entire world speaks the same language, But there is a specific level of quality vendor's implementation must meet in order to be declared as standard Open GL. It supports many platforms just like Windows, Linux, many Unix platforms, PlayStation 3, and Mac OS X.

the OpenGL Architecture Review Board (ARB) –which consists of many different companies- is the principal responsible of administration.

There are several extensions that made to trace the technology evolution, sometimes they are officially adopted and become standard extension which later on become part of the core specification.

The OpenGL ARB released a series of manuals along with the specification which have been updated to track changes in the API:

- The Red Book - The OpenGL Programmer's guide: A readable tutorial and reference book - this is a 'must have' book for OpenGL programmers.

- The Blue Book - The OpenGL Reference manual: Essentially a hard-copy printout of the 'man' pages for OpenGL. Includes a poster-sized fold-out diagram showing the structure of an idealized OpenGL implementation.

- The Green Book - Programming OpenGL for the X Window System: A book about X11 interfacing and GLUT.

- The Alpha Book (white) - OpenGL Programming for Windows 95 and Windows NT: A book about interfacing OpenGL to Microsoft Windows. Then, for OpenGL 2.0 and beyond.

المراجع

1. Sam Buss. 3D Computer Graphics: A mathematical approach with OpenGL, Cambridge University Press, 2003.

2. E. Angel, Interactive Computer Graphics – A top down approach with OpenGL, , 2nd edition, Addison Wesley, 2000

3. Donald Hearn and M. Pauline Baker Computer Graphics (C Version),, Prentice Hall, 1997.

4. J. D. Foley et al, Computer Graphics: Principles and practice., 2nd edition, Addison Wesley, 1996

5. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics, Principles and Practice, Second Edition,Addison-Wesley, 1990.

6. Vera B. Anamd, Computer Graphics and Geometric Modeling for Engineers, John Willey, 1983