تفريغ مقرر بيانيات الحاسب| BCG601

Slash team

برنامج الهندسة المعلوماتية – الجـــــامعة الافتراضية السورية

تفريغ الطالب/ـــة:	ميادة ميرو
المدرّس المحاضر:	د. فاتن مشتا
عنوان المحاضرة	**الألوان Colors**
رقم المحاضرة	6
تاريخ بث المحاضرة	2023/1/7

الألوان COLORS

مقدمة:

الهدف من المحاضرة هو العثور على تمثيل الألوان.

(المجال المرئي لرؤية الألوان لدى البشر من 400 الى 700)

- كل لون يتألف من عدة أطوال موجات تساعد في تشكيله.
- اللون الأبيض مركب من كل ألوان الطيف.

نحتاج إلى:

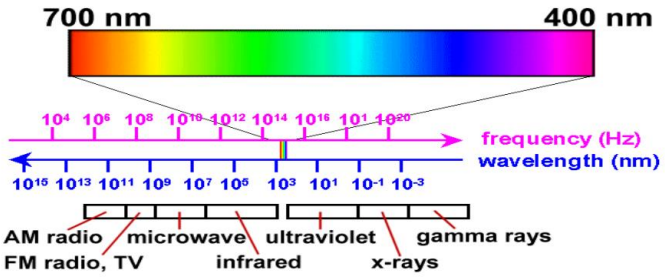
- توصيف اللون ب 3 معاملات حتى يتوافق مع الإحداثيات ثلاثية البعد.
- ربط كل نقطة من المشهد أو سطح بلون.
- توثيق الألوان بدقة.
- طريقة لقياس الألوان بدقة وتحويلها في الأوساط المختلفة.

الألوان ليس هدفها فقط إضافة جمال للغرض ولكنها تحمل أيضاً معلومات مهمة.

اللون هو complex subject (موضوع معقد) له أبعاد عديدة فيزيائية ونفسية وغيرها
وللآن لا يوجد نظرية موحدة لتمثيل الألوان بدقة.

الضوء الذي نستقبله من الشمس هو عبارة عن عدة ألوان أو حزمة ترددية ضيقة من الطيف الكهرطيسي التي تقابل أطوال الموجات من 370nm حتى 780nm.

$$\lambda = \frac{c}{f}$$



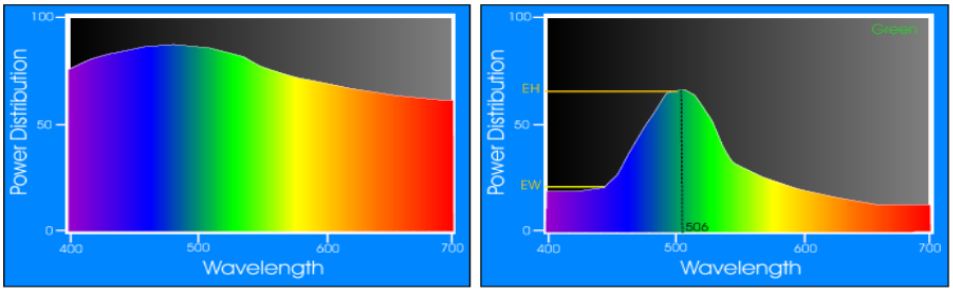
At the **low-frequency** end : **Reds colors (3.8*10^{14} Hz - 780 nm)**

At the **high-frequency** end: **Violet colors (7.9*10^{14}Hz - 380nm)**

Each wavelength value within the visible band corresponds to a distinct color.

ملاحظة: هذه الأطوال غير مطلوبة وإنما هي معلومات عامة فقط.

إذا حللنا كل لون الى مركباته الترددية نحصل على الشكل التالي:



اللون الأخضر اللون الأبيض

تابع الكثافة الترددية مهمته تحليل كل لون الى مركباته كأطوال موجات.

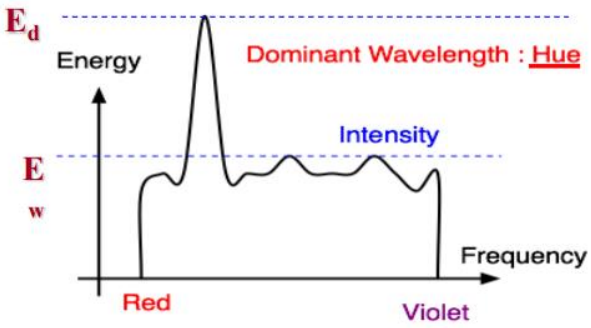
وهذه طريقة مناسبة لتمثيل الألوان ولكنها غير مناسبة لتمثيلها على الحاسب.

لذلك تم اعتماد طريقة أخرة لتمثيل الألوان على الحاسب تعتمد على تحديد 3 معاملات لكل لون :

- Hue (موجة مسيطرة)
- Saturation نقاء اللون
- Luminance لمعان اللون

توزع الطاقة للون معين

نلاحظ وجود موجة مسيطرة (E_d) وباقي أطوال الموجات تساهم بنسبة ما بتكوين اللون تكون استطاعتهم (E_w).



- الموجة المسيطرة hue

تشكل spike (تصاعد أو قفزة) في المخطط.

- نقاء اللون saturation

كلما كانت قمة شاهقة أكثر كلما كون اللون نقي أكثر.

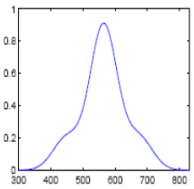
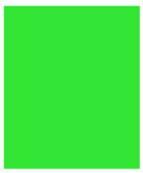
نسبة النقاء تتراوح بين ال 0 و 100

تكون 100 عندما يكون لدينا مركبة واحدة مشاركة في ترتيب اللون $E_w = 0$

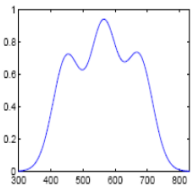
باللون الأبيض نسبة النقاء 0 $E_d = E_w$

كلما كانت E_d أكبر كلما كلن أنقى اللون.

مثال:

اللون الأول نقي بشكل عالي يوجد قمة واحدة.



اللون الثاني قليل النقاء يوجد عدة قمم.

- لمعان اللون luminance

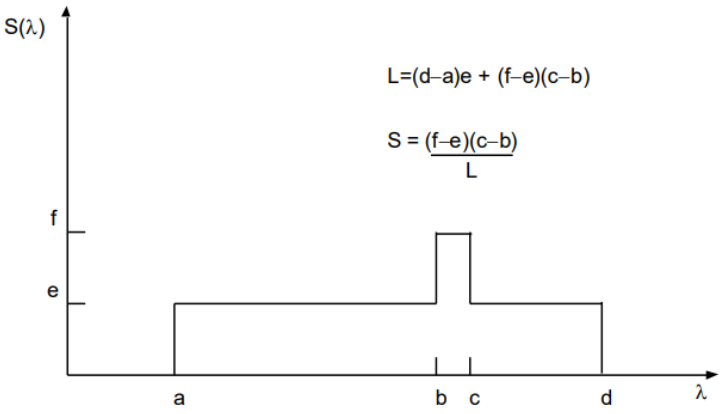
مستوى الإضاءة في اللون أو شدة اللون المنعكس على الغرض.

حيث L هو اللمعان luminance وهو

مساحة المستطيلين معاً.

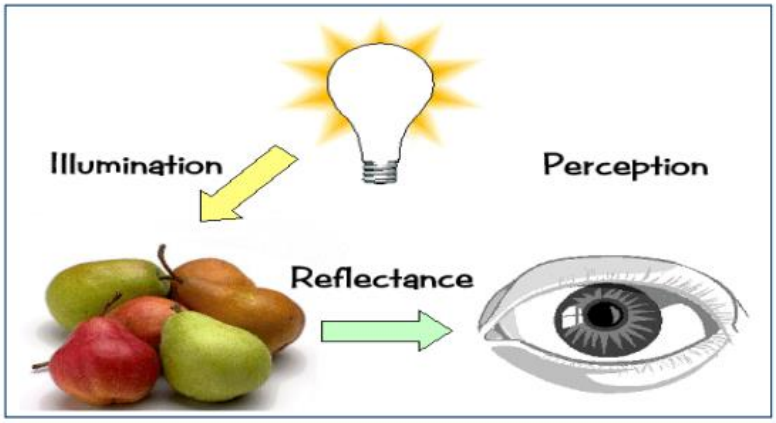
و S هو النقاء وهو نسبة مساحة المستطيل

الصغير على مساحة المستطيلين معاً.



مثال:

يوضح كيفية انعكاس الضوء على الأغراض



وكيفية رؤيتنا لها.

ملاحظة: رؤيتنا للأغراض لا تتعلق بلون الغرض نفسه فقط بل أيضاً بلون مصدر الإضاءة (الأغراض الحمراء لا نستطيع رؤيتها باللون الأزرق).

تابع يحسب منبع الإضاءة LIGHT SOURCE

- توزع إضادة المنبع $E(\lambda)$

يجب أن نميز طبيعة المادة:

1. مواد شفافة:

تمتص جزء بسيط من الضوء وتمرر الضوء بشكل غير مبعثر من خلالها (مثل النوافذ)

2. مواد نصف شفافة:

تمتص جزء من الضوء وتمرر جزء وتبعثر جزء.

3. مواد عاتمة:

تمتص كل الضوء.

- طبيعة المادة تعرف بتابع الاستطاعة للضوء المنعكس $S(\lambda)$

بالتالي نستطيع الحصول على الاستطاعة الطيفية للضوء المنعكس عن الغرض بالعلاقة التالية:

$$P(\lambda) = E(\lambda) * S(\lambda)$$

نظرية اللون الثلاثي TRICHROMATIC COLOR THEORY

على شاشة الحاسب نربط كل بكسل بلون معين وبالتالي كل بكسل سوف يبث 3 مركبات لونية (الأحمر – الأخضر - الأزرق) ولكن بنسب مختلفة.

يوجد نموذجين :

- نموذج RGBA

حيث يحوي 3 ألوان (أحمر – أخضر - أزرق) ويمكن إضافة معامل رابع A يمثل شفافية الغرض (شفاف – نصف شفاف - عاتم).

عند تخزين الصورة في هذا النموذج يكون لكل بكسل 3 مركبات.

- نموذج Color index

في هذا النموذج عوضاً عن تعريف كل لون ب 3 مركبات نستبدل الألوان برقم واحد index بحيث ننشئ جدول يسمى ال color map تترتب فيه الألوان

فمثلاً ال index 0 يمثل اللون الأبيض 1 1 1

Index 1 يمثل اللون الأبيض 0 0 0

بحيث عند تخزين صورة أعطي لكل بكسل index اللون فقط.

عندما يتم تنفيذ البرنامج يتم تحديد لون لكل رأس vertex

عندما نرسم الغرض نحدد لون لكل رأس ولكن عند عرضه على الشاشة يتم أيضاً حساب اللون لأنه كما سبق وذكرنا رؤيتنا للون الغرض لا تعتمد على لونه فقط بل أيضاً على لون الإضاءة فإذا فعلت الإضادة احتاج لإعادة حساب اللون عند كل بكسل.

اللون الجديد المحسوب يتعلق بكل من :

- موقع الغرض (بعده عن مصدر الإضاءة)
- الناظم على سطح الغرض (كيف ينعكس الضوء)
- نوع المادة (نسبه الضوء الذي تعكسه)

ملاحظة: لن نهتم بالتفاصيل الرياضية للحسابات السابقة ما يهمنا فقط أنه إذا تم تفعيل الإضاءة فنحن بحاجة الى إعادة حساب اللون في كل بكسل وإذا لم تفعل الإضاءة تبقة نفس الألوان السابقة.

مثال:

لو كان لدينا كرة حمراء ومصدر الضوء أزرق فسوف تظهر بمظهر مختلف لو عرضناها بدون الإضاءة.

RGB COLOR MODE

في نمط ال RGBA كل نمط مرتبط ب 3 مركبات.

يوجد عدد بتات محدد لتمثيل الألوان تدعى bit planes

فلو حددت أن لدي n بت لتمثيل الألوان فيكون لدي 2^n لون مختلف تقسم هذه الألوان بين المركبات اللونية الثلاثة (RGB)

فلو كان الحاسب مخصص $24bit$ فأنا استطيع عرض $2^{24} = 16.77million$ لون مختلف.

ملاحظة: هذه المعلومة مهمة جداً وسؤال محتمل للامتحان

Q- لو كان لدي 24bit للألوان فكم عدد الألوان التي استطيع تمييزها وعرضها على الشاشة؟

A- $2^{24} = 16.77million$

يتم تقسيم ال 24bit كالتالي:

8bit red – 8bit green – 8bit blue

أحياناً يكون لدينا فقط 8bit لتمثيل اللون فتتوزع كالتالي

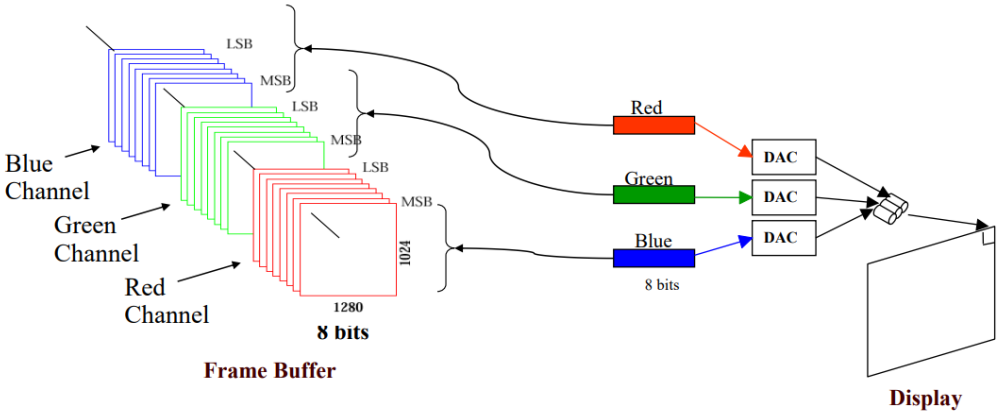
3bit red – 3bit green – 2bit blue

لمعرفة عددبتات ال bit planes في open gl نستعمل التعليمة:

glGetIntegerv() with GL_RED_BITS, GL_GREEN_BITS, GL_BLUE_BITS, GL_ALPHA_BITS, and GL_INDEX_BITS.

مع تحديد عدد البتات التي تمثل كل من الألوان (أحمر – أخضر - أزرق)

كيفية عرض الصور على الحاسب



24-bit Color Images

لدينا صورة حجمها 1024*1280 كل بكسل يتم تمثيله على 3byte(24bit)

بالتالي ال frame buffer يتم تمثيلها على 8bit أي كل مركبة لونية يتم تمثيلها على 8bit

ثم يتم تحويلهن لإشارات كهربائية تسقط على البكسل المطلوب في الشاشة وتعرض اللون المناسب.

ملاحظة: سؤال ممكن للامتحان

Q- ليكن لدي frame buffer طوله 8bit كم عدد الألوان المختلفة التي يمكن عرضها على الشاشة؟

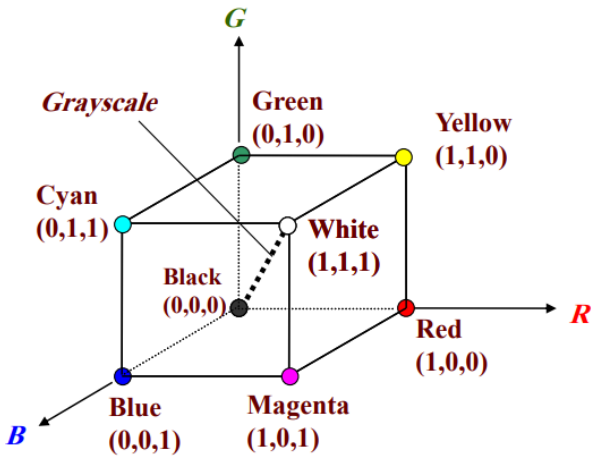
A- $2^{24} = 16.77 million$

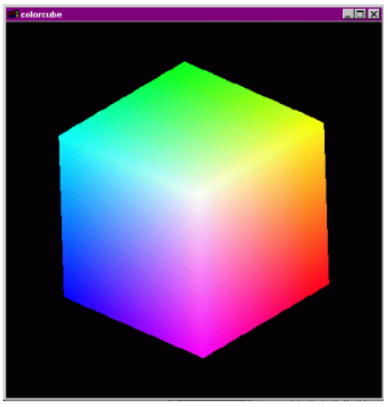
لأن frame buffer تعني مركبه لونية واحدة بالتالي يوجد 8bit للأخضر و 8bit للأزرق أي 24

التخزين في الذاكرة يساوي حجم الصورة * عدد البتات

Memory requirement: 1280 x 1024 x 3 bytes (24 bits) ≈ 4 Mb

- مكعب الألوان RGB color cube





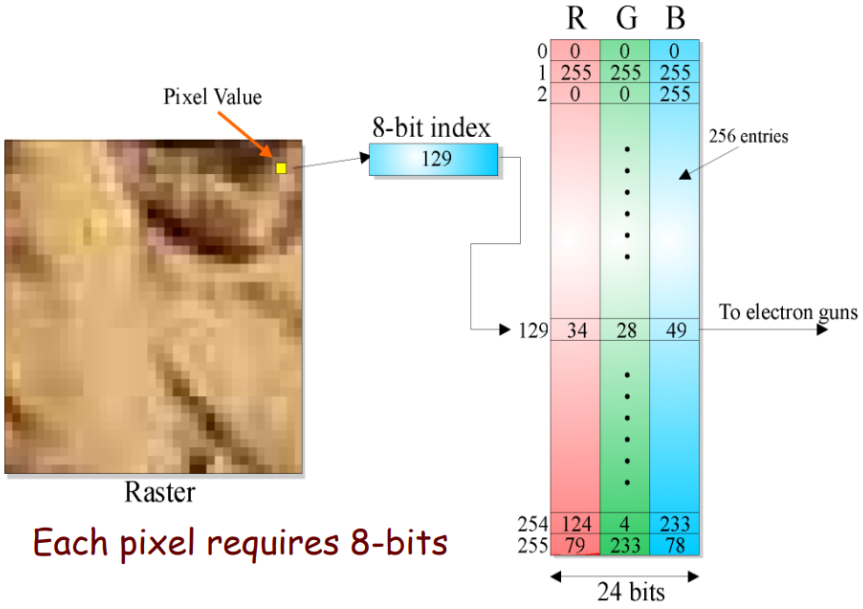
RGB color cube

يمكن أن نخزن عليه كل الألوان التي يمكن تمييزها بواسطة الألوان الثلاثة (RGB)

COLOR INDEX MODE

نربط كل لون ب index

نلجئ لهذه الطريقة عندما يكون حجم ال bit planes قليل أو محدود.



نربط كل رأس ب index اللون فقط بالتالي نوفر حجم التخزين.

لوكان لدي 8bit فيمكن تمثيل 256 لون مختلف.

حيث هنا في هذا النمط أستطيع تحديد الألوان التي في الصورة واسنادها الى color map بالتالي نحصل على صورة أوضح مما يمكن أن تكون عليه في نمط RGB.

في مكتبة OPEN GL

تعليمة GL COLOR();

نستطيع تحديد اللون من نمط RGB

```
void glColor3{b s i f d ub us ui} (TYPEr, TYPEg,
TYPEb);
void glColor4{b s i f d ub us ui} (TYPEr, TYPEg,
TYPEb, TYPEa);
void glColor3{b s i f d ub us ui}v (const TYPE*v);
void glColor4{b s i f d ub us ui}v (const TYPE*v);
```

1. حيث الأحرف الصغيرة هي اختصار للنمط

Byte – short – integer – float – double

unsigned byte – unsigned short – unsigned integer

ثم يليها المركبات الثلاثة للألوان RGB وفي حال كانت أربعة ف مع A للشفافية

كما يمكن إضافة الحرف V أي إضافة الألوان الثلاثة كشعاع

يمكن تمثيل هذه الألوان :

2. من 0 الى 1 بنمط FLOAT
3. من 0 الى 255 بنمط INTEGER

تعليمة GL INDEX();

لتحديد اللون بنمط Color index

```
void glIndex{sifd ub}(TYPE c);
void glIndex{sifd ub}v(const TYPE *c);
```

تعليمة GL CLEAR COLOR();

استطيع ادخال المركبات للون الذي أريده للخلفية ولكن لا يتم وضع لون الخلفية حتى تنفيذ تعليمة gl color();

عند وضع المركبات كلها على الواحد يعني أن الخلفية أبيض وفي حال وضعهم 0 تكون الخلفية باللون الأسود.

```
glClearColor (GLclampf red, GLclampf green, GLclampf blue,
GLclampf alpha):
```

```
glClearColor (1.0, 1.0, 1.0, 0.0);
```

ملاحظة: سؤال للامتحان

Q- ما لون الخلفية في حال تنفيذ التعليمة التالية:

```
glClearColor (1.0, 1.0, 1.0, 0.0);
```

A- تكون الخلفية باللون الأبيض.

تعليمة GL SHAPE MODEL();

```
glShadeModel( GLenum mode )
```

يوجد نمطين للشكل:

4. SMOOTH

نحسب اللون في كل بكسل ممكن أن يأخذ كل بكسل لون مختلف

5. FLAT

نحسب اللون عند أحد الرؤوس ونسنده الى كل البكسلات بالشكل

(كل البكسلات نفس اللون)

ملاحظة: المعلومة السابقة مهمة جداً وسؤال محتمل للامتحان التمييز بين النمطين.

نهاية المحاضرة الحادية عشر