**بسم الله الرحمان الرحيم**

انشاء نافذة و الرسم عليها

الكود الذي طبقناه سابقا لا يقوم باي شيئ ,و لكننا استعملناه للتأكد من سلامة تنصيب الـ SDL . سنشرع الان في كتابة كود لاظهار نافذة و العمل عليها .

تهيئة الـSDL للعمل :

اغلب المكتبات الاضافية تتطلب مرحلة تهيئ حتى تتمكن هذه الاخيرة من تحميل بعض المعلومات الضرورية لعملها في الذاكرة , الدالة المسؤولة عن تهيئة الـ SDL معرفة كما يلي : int SDL\_Init(Uint32 flags) ;

هذه الدالة تستقبل علم (flag) و الذي يحدد ما الذي يجب تهيئته , الاعلام المتاحة هي :

|  |  |
| --- | --- |
| يهيئ نظام الوقت | SDL\_INIT\_TIMER |
| يهيئ نظام الصوت | SDL\_INIT\_AUDIO |
| يهيئ نظام الرسم و العرض | SDL\_INIT\_VIDEO |
| يهيئ نظام قارئ الاقراص المضغوطة | SDL\_INIT\_CDROM |
| يهيئ نظام ادوات التحكم بالالعاب | SDL\_INIT\_JOYSTICK |
| يهيئ كل الانظمة | SDL\_INIT\_EVERYTHING |

هناك أعلام أخرى لم يتم التطرق لها.

في هذا الدرس سوف نهيئ نظاماً واحداً فقط و هو نظام الرسم و العرض (SDL\_INIT\_VIDEO) .

**SDL\_Init(SDL\_INIT\_VIDEO);**

بما إننا هيئنا أنظمة الـ SDL و حجزنا موقعا في الذاكرة فهذا يتطلب منا تحرير الموقع المحجوز . يتم ذلك بواسطة الدالة التالية : void SDL\_Quit(void) ;

يتم استدعاء هذه الدالة بطريقتين اسهلها :

**atexit(SDL\_Quit);**

أما الطريقة الثانية فهي استدعاء الدالة قبل نهاية الدالة الأصلية Main

int main(int argc, char \*argv[]) {

**SDL\_Init(SDL\_INIT\_VIDEO);**

// الكود الخاص ببرنامجكم

SDL\_Quit() ;

Return 0 ;

}

لكن هناك مشكلة ،من يضمن لنا أن المكتبة تهيئت بالشكل المطلوب ؟ ماذا لو ان المساحة المتوفرة في الذاكرة ليست كافية ؟

سنحل هذه المشكلة بالتحقق من القيمة التي ترجعها SDL\_Init :

if ( SDL\_Init( SDL\_INIT\_VIDEO ) < 0 )

{

printf( "Unable to init SDL: %s\n", SDL\_GetError() );

return 1; // الخروج من البرنامج

}

بهذه الطريقة نكون متأكدين بان المكتبة تهيئت كما يجب .لان الدالة **SDL\_Init** ترجع قيمة موجبة إذا تمت التهيئة بنجاح ،أما إذا فشلت التهيئة فان الدالة سترجع قيمة سالبة .

**ملاحظة** : الدالة **SDL\_GetError()** تخبرنا عن سبب عدم تهيئة المكتبة,بصفة عامة الدالة تخبرنا عن مصدر الاخطاء .

بما أن المكتبة مهيأة كما يجب سنبدا الان بكتابة الكود الخاص بالنافذة .

حتى نحصل على نافذة يجب علينا تعريف مؤشر من نوع خاص بالمكتبة و هو SDL\_Surface ( السطح الذي ستعمل عليه الـ SDL ).

SDL\_Surface\* screen = NULL ;

ثم نقوم باستدعاء الدالة : SDL\_Surface \***SDL\_SetVideoMode**(int width, int height, int bpp, Uint32 flags);

* Width : عرض النافذة
* Heigh : طول النافذة
* Bpp : عدد البتات الممثلة للبكسل (8 , 16 , 24 ,32)
* Flags : علم يرشد المكتبة إلى طريقة التعامل مع الذاكرة. العلم الذي سنستخدمه في هذه المرحلة هو SDL\_HWSURFACE و الذي يسمح لبرنامجنا باستغلال ذاكرة كارت العرض السريعة جدا و ذات المساحة المحدودة

screen = SDL\_SetVideoMode(640,480,32, SDL\_HWSURFACE) ;

تقوم هذه الدالة بإنشاء نافذة بالمواصفات التي يحددها المبرمج .و تقوم بحجز المساحة اللازمة لعمل النافذة .

كما حدث في مرحلة التهيئة لا شيئ يضمن لنا ان المساحة تم حجزها بنجاح لذلك يجب التاكد من ذلك :

if ( !screen ){

printf("Unable to set 640x480 video: %s\n", SDL\_GetError());

return 1;

>

}

الآن كل شيء على ما يرام يمكننا بناء البرنامج للتمتع بالنافذة .

....

هل رايت النافذة ؟ لا ؟ من حسن حظك و الا وجب عليك رمي جهازك في المزبلة .

السبب في عدم ظهور النافذة هو سرعة الكمبيوتر ,بمعنى اخر النافذة ظهرت و اختفت في لمح البصر...و ربما اسرع !

الحل الوحيد و المطبق في جميع البرامج هو الحلقات (Boucle) و بوجه الخصوص الحلقة while .

سأطلب منك إضافة هذا الكود الذي ساشرحه في مرحلة لاحقة :

void pause()

{

int continuer = 1;

SDL\_Event event;

while (continuer)

{

SDL\_WaitEvent(&event);

switch(event.type)

{

case SDL\_QUIT:

continuer = 0;

}

}

قم باستدعاء هذه الدالة في الدالة الأصلية Main كالأتي :

#include <stdlib.h>

#include <stdio.h>

#include <SDL/SDL.h>

void pause();

int main(int argc, char \*argv[])

{

//تهيئة المكتبة

if ( SDL\_Init( SDL\_INIT\_VIDEO ) < 0 ) {

printf( "Unable to init SDL: %s\n", SDL\_GetError() );

return 1;

}

//إنشاء النافذة

SDL\_Surface\* screen = SDL\_SetVideoMode(640,480,32, SDL\_HWSURFACE) ;

if ( !screen ){

printf("Unable to set 640x480 video: %s\n", SDL\_GetError());

return 1;

}

pause(); //تثبيط البرنامج

SDL\_FreeSurface(screen) ;// تسريح سطح النافذة

atexist( SDL\_Quit); // تسريح المساحة التي تم حجزها لعمل المكتبة

return 0; // انتهاء البرنامج

}

void pause()

{

int continuer = 1;

SDL\_Event event;

while (continuer)

{

SDL\_WaitEvent(&event);

switch(event.type)

{

case SDL\_QUIT:

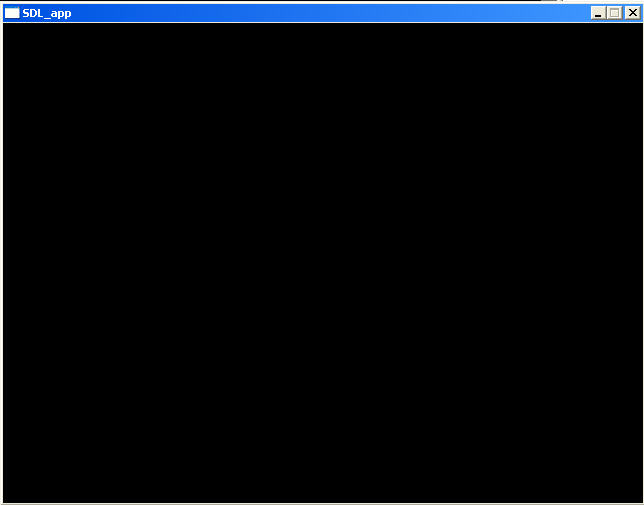
continuer = 0;

}

}

}

قم ببناء البرنامج و شغله ...مبارك لك نافذتك الأولى !



كل هذا الجهد و العناء للحصول عل مجرد نافذة سوداء بشعة ذات عنوان تلقائي و... اعتذر، فقدت أعصابي !

هيا بنا لنجمل نافذتنا, أولاً شيء بلا اسم يعتبر لا شيء لهذا سنعطي عنوانا واضحا للنافذة و ذلك باستعمال الدالة: void **SDL\_WM\_SetCaption**(const char \*title, const char \*icon);

* Title : عنوان النافذة المراد إدراجه
* Icon : اسم الايقونة.

يتم استدعاء هذه الدالة ( و اغلب دوال الـ SDL ) بعد التهيئة .

if ( SDL\_Init( SDL\_INIT\_VIDEO ) < 0 ) {

......

}

SDL\_WM\_SetCaption("My First Window" , NULL ) ;

أما بالنسبة للايقونة فسيتم التطرق إليها في المرحلة القادمة من الدرس ( حتى لا تتعقد الأمور).

بالنسبة لي فانا امقت اللون الأسود ، أفضل الألوان الفاتحة ،إذا كنت تشاطرني الرأي فسأدلك على الدالة السحرية التي تمكننا من تغير لون سطح النافذة .

الدالة هي : int **SDL\_FillRect**(SDL\_Surface \*dst, SDL\_Rect \*dstrect, Uint32 color);

* Dst : السطح المراد تلوينه .
* Dstrect : المساحة المراد تلوينها .
* Color : اللون المراد استعماله .

لتعلم أن الألوان تمثل بثلاث قيم ( في جميع أجهزة الكمبيوتر )،و هي :

* قيمة اللون الأحمر و التي يرمز لها بـ :R
* قيمة اللون الأخضر و التي يرمز لها بـ : G
* قيمة اللون الأزرق و التي يرمز لها بـ : B

للحصول على القيم الثلاثة الخاصة بلون ما يمكن استعمال برنامج رسم بسيط مثل Paint .

كل هذا لكي أنبهك إلى أن الدالة **SDL\_FillRect** لا تستقبل إلا قيمة واحدة ترمز للون المراد استعماله، كيف يتم ذلك ؟

دون التعمق في التفاصيل الدالة التي تحول القيم الثلاثة الخاصة باللون إلى قيمة واحدة هي : Uint32 **SDL\_MapRGB**(SDL\_PixelFormat \*fmt, Uint8 r, Uint8 g, Uint8 b);

* Fmt : نوع البيكسل الذي يتعامل معه سطح النافذة
* R : قيمة اللون الأحمر
* G : قيمة اللون الأخضر
* B : قيمة اللون الأزرق

استعمال هذه الدالة بسيط جدا ،نرسل إليها نوع البيكسل و الذي نجده في المتغيرات ذات النوع SDL\_Surface كما يلي : screen->format ،ثم نحدد اللون و انتهى الأمر .

SDL\_FillRect( screen , NULL , SDL\_MapRGB(screen->format , 170 , 240 ,120 ) ) ;

Pause() ;

اذا قمت ببناء برنامجك و تشغيله فستلاحظ ان النافذة ما زالت سوداء رغم استدعائك للدالة التي من المفترض ان تلون لك نافذتك .

قبل كل شيء لا تلم دالة التلوين فقد قامت بعملها على أكمل وجه، المشكلة أننا لم ننزع اللون الأسود من سطح النافذة، و لكي يظهر اللون الذي حددناه يجب مسح و تحديث سطح النافذة و ذلك باستدعاء الدالة التالية :

int **SDL\_Flip**(SDL\_Surface \*screen);

هذه الدالة تهيئ سطح النافذة للرسم عليها و كأنها فارغة، بعبارة أخرى هذه الدالة تفرغ سطح النافذة من جميع البكسلات الموجودة على سطحها .

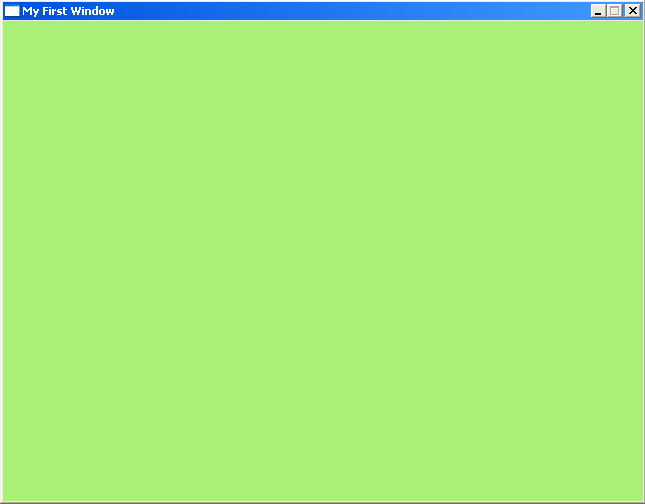
استعمال هذه الدالة غابة في البساطة :

SDL\_FillRect( screen , NULL , SDL\_MapRGB(screen->format , 170 , 240 ,120 ) ) ;

SDL\_Flip(sreen) ;

Pause() ;

اعد بناء البرنامج ... النافذة أجمل مما كانت عليه :



إنشاء عدة سطوح:

بعد أن أصبحت نافذتنا أجمل بشكل ملحوظ من النافذة الأولى سيكون من السذاجة التوقف عند هذا الحد ... لما لا نجعل نافذتنا أكثر جمالا و أكثر احترافية حتى تتمكن من إدهاش أصدقائك ؟

لننطلق باسم الله .

كما ذكرت سابقا كل شيء يظهر على النافذة عبارة عن سطح ، و لحد ألان استعملنا سطحا واحدا و هو سطح النافذة الرئيسي .سنقوم ألان بإنشاء عدة سطوح و لصقها على السطح الرئيسي .

حسنا إذا قلنا سطح فهذا يستلزم تعريف مؤشر نحو النوع SDL\_Surface .

SDL\_Surface\* Smal\_surface = NULL

ثم نحجز مساحة في الذاكرة لهذا السطح الجديد، الدالة التي تقوم بذلك هي :

SDL\_Surface \***SDL\_CreateRGBSurface**(Uint32 flags, int width, int height, int depth, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask);

* Flags : أعلام تبين كيفية استعمال الذاكرة لهذا السطح ، الذي يهمنا بين هذه الأعلام هو : SDL\_HWSURFACE لكن توجد عدة أعلام
* Width : عرض السطح
* Height : طول السطح
* Depth : عدد البتات الممثلة للبكسل
* Rmask ,Gmasl,Bmask, Amask : لن يتم التطرق لها ، سنكتفي بتصفيرها .

هذه الدالة ترجع مؤشرا لللسطح الذي تم حجزه إذا تم الحجز بنجاح و إلا فإنها ترجع NULL في حالة فشل عملية الحجز.

Smal\_surface = SDL\_CreateRGBSurface(SDL\_HWSURFACE , 320 , 240 , 32 ,0,0,0,0);

If ( smal\_surface == NULL ) exit(1);

و كما تعلم إذا وجد حجز سيوجد تسريح ،و الدالة التي تسمح لنا بتسريح السطوح (SDL\_Surface) هي:

) ; void SDL\_FreeSurface(SDL\_Surface \*surface

يتم استدعاء دالة التسريح عند انتهاء البرنامج و قبل استدعاء الدالة : SDL\_Quit()

الآن يمكن تطبيق دالة التلوين على السطح بنفس الطريقة التي استعملناها لتلوين السطح الرئيسي للنافذة .

SDL\_FillRect( screen , NULL , SDL\_MapRGB(screen->format , 170 , 240 ,120 ) ) ;

**SDL\_FillRect(Smal\_surface , NULL , SDL\_MapRGB(screen->format , 250 , 250 ,250 ) ) ;**

SDL\_Flip(sreen) ;

لست مضطراً لإعادة مسح السطوح التي تقوم بإنشائها، السطح الوحيد الذي يستلزم مسحه هو السطح الرئيسي للنافذة.

الآن نافذتنا جاهزة،ما علينا سوى لصقها على السطح الرئيسي .

أريد أن أنبه أن السطوح في الـ SDL و في جميع المكتبات الأخرى تعمل تماما كمعلم الرياضيات الذي يتم إنشاء الأشكال عليه ... الم يقل لك أستاذك يوما ما اوجد x,y إحداثيات النقطة a ... كان عليك الانتباه أثناء الدرس !

حسنا لو لم تسمع كلمة إحداثيات من قبل فإنها (أي الإحداثيات)هي المكان الذي تتواجد فيه النقطة . و بما أن سطح النافذة عبارة عن معلم فان السطح الذي نريد لصقه عليها يجب أن تكون له إحداثيات حتى تحدد الـ SDL مكان لصقه.

انس أمر الرياضيات ألان و احفظ هذا النوع الجديد SDL\_Rect

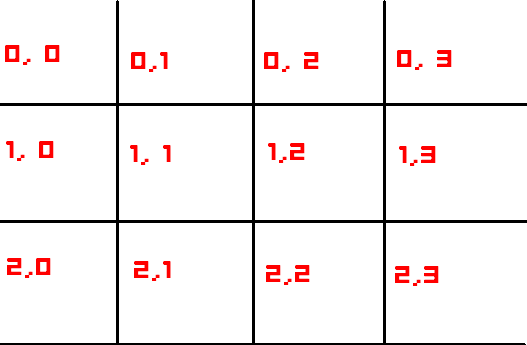
يحتوي هذا النوع على عدة معلومات و هي :

* X : فاصلة البكسل الأول من السطح المراد لصقه
* Y : ترتيبة البكسل الأول من السطح المراد لصقه
* W :عرض الإطار المراد لصقه
* H : طول الإطار المراد لصقه .

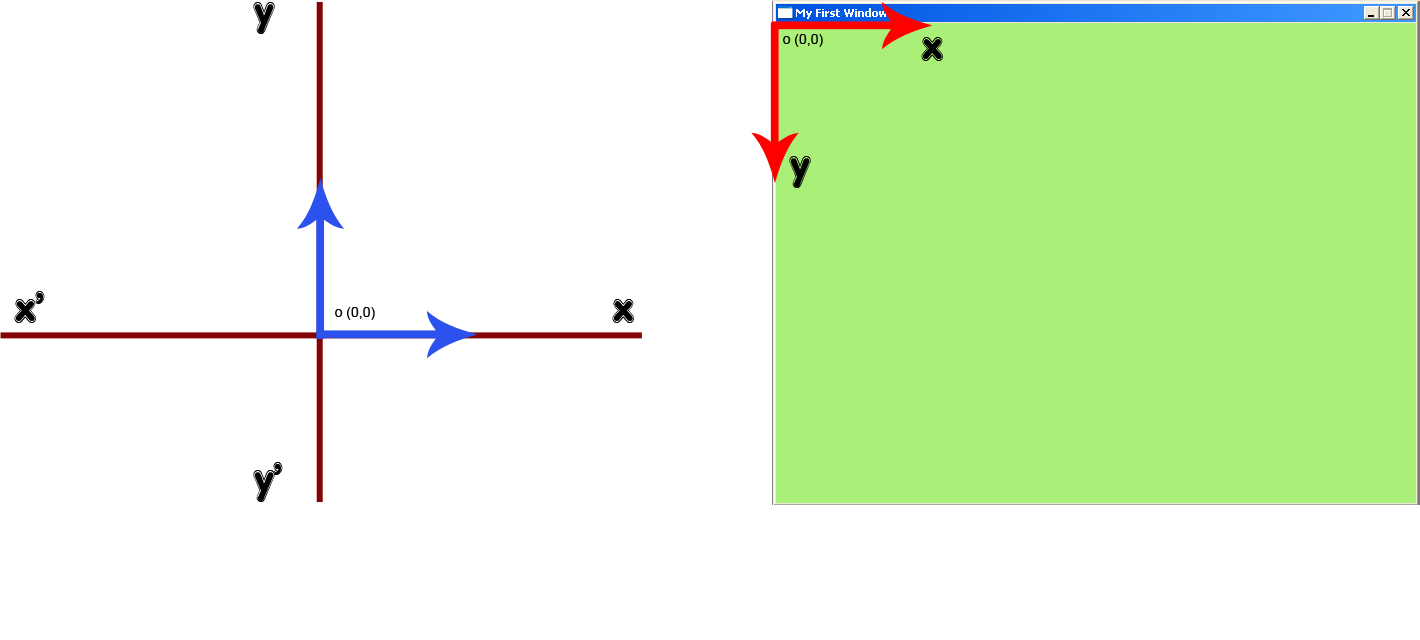
سوف يتم التطرق إلى العنصرين w ,h في مرحلة قادمة .

ملاحظة : يجب أن تعلم أن أول بيكسل من السطح يرمز له بـ (0,0) و هو البيكسل الذي يمثل الإحداثيات (x ,y) أي ان السطح سيتم لصقه انطلاقا من هذا البيكسل .

الصورة أدناه توضح تتابع البكسلات في جميع السطوح (و حتى الصور لها نفس الترتيب و سترون لاحقا إن الصور عبارة عن سطوح )



إن محور التراتيب (y,y’) في الـ SDL معكوس،بمعنى أخر إذا أردنا إضافة قيمة موجبة ل y فان السطح سيتجه نحو الأسفل والعكس صحيح ، هذا الشكل يبين السبب :



**في الرياضيات و العلوم بصفة عامة**

**SDL**

حسنا ، الآن يمكننا تعريف متغير من نوع SDL\_Rect :

SDL\_Rect Position ;

Position.x = 320 ;

Position.y = 240 ;

ثم نقوم باستدعاء الدالة التي ستقوم بعملية اللصق(Blit) و هي :

int **SDL\_BlitSurface**(SDL\_Surface \*src, SDL\_Rect \*srcrect, SDL\_Surface \*dst, SDL\_Rect \*dstrect);

* Src : السطح المراد لصقه
* Srcrect : الجزء المراد لصقه من السطح
* Dst : السطح الذي سيتم عليه اللصق
* Dstrect : الإحداثيات.

نستدعي هذه الدالة قبل دالة المسح (ليس شرطا إجباريا لكنه مستحسن). :

//تلوين السطح

SDL\_FillRect(Smal\_surface , NULL , SDL\_MapRGB(screen->format , 250 , 250 ,250 ) ) ;

//لصق السطح

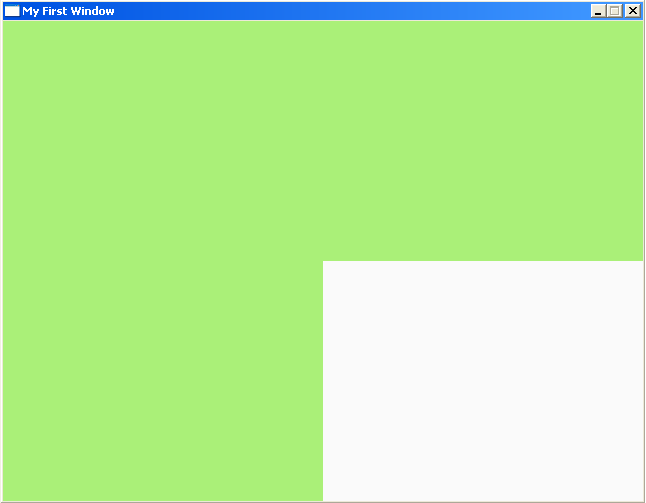
SDL\_BlitSurface( Smal\_surface, NULL , screen, &Position );

// تحديث البكسلات الخاصة بالسطح الرئيسي

SDL\_Flip(screen);

pause();

اعد بناء برنامجك ...شغله ... النتيجة :

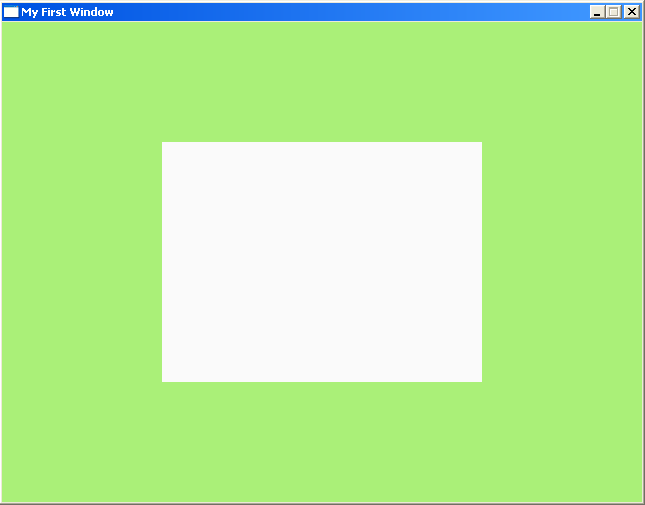


إن مكان السطح ليس جميلا لهذا سنقوم بلصق السطح وسط النافذة بعمليات بسيطة جدا ، جدا :

Position.x = ( (screen->w/2) - (Smal\_surface->w/2) ) ;

Position.y = ( (screen->h/2 ) - (Smal\_surface->h/2) ) ;

النتيجة :



الكود الكلي للبرنامج (بدون تعليقات عليه لان التعليقات موجودة في الدرس) :

#include <stdlib.h>

#include <stdio.h>

#include <SDL/SDL.h>

void pause();

int main(int argc, char \*argv[])

{

if ( SDL\_Init( SDL\_INIT\_VIDEO ) < 0 ) {

printf( "Unable to init SDL: %s\n", SDL\_GetError() );

return 1;

}

SDL\_Surface\* screen = SDL\_SetVideoMode(640,480,32, SDL\_HWSURFACE) ;

if ( !screen ){

printf("Unable to set 640x480 video: %s\n", SDL\_GetError());

return 1;

}

SDL\_WM\_SetCaption("My First Window" , "ates.ico" ) ;

SDL\_Surface\* Smal\_surface = SDL\_CreateRGBSurface(SDL\_HWSURFACE , 320 , 240 , 32 ,0,0,0,0);

SDL\_Rect Position ;

Position.x = ( (screen->w/2) - (Smal\_surface->w/2) ) ;

Position.y = ( (screen->h/2) - (Smal\_surface->h/2) ) ;

SDL\_FillRect( screen , NULL , SDL\_MapRGB(screen->format , 170 , 240 ,120 ) ) ;

SDL\_FillRect(Smal\_surface , NULL , SDL\_MapRGB(screen->format , 250 , 250 ,250 ) ) ;

SDL\_BlitSurface( Smal\_surface, NULL , screen, &Position );

SDL\_Flip(screen);

pause();

SDL\_FreeSurface(Smal\_surface) ;

atexit( SDL\_Quit);

return 0;

}

void pause()

{

int continuer = 1;

SDL\_Event event;

while (continuer)

{

SDL\_WaitEvent(&event);

switch(event.type)

{

case SDL\_QUIT:

continuer = 0;

}

}

}

انتهت هذه المرحلة من الدرس،و في انتظار المرحلة القادمة سأعطيكم بعض الوظائف التي سأراقبها قي المرحلة القادمة و يا ويل من لم يقم بوظيفته !

الوظيفة الأولى اكتب برنامجا يقوم بفتح نافذة ذات الأبعاد التالية : 320،240،32

اجعل برنامجك قادرا على رسم تدرج لوني بين الأبيض و الأسود بهذه الكيفية :



توجد طريقتين الأولى سهلة لكن غير فعالة و الثانية سهلة و فعالة .

التطبيق الثاني : اكتب برنامجا مماثلا لبرنامج التطبيق السابق،لكن بدل رسم تدرج عمودي للألوان ،هذه المرة يجب رسم تدرج أفقي .

التطبيق الثالث : اكتب برنامجا يرسم تدرج لوني بلونين غير الأبيض و الأسود .

التصحيح سيكون في موضوع منفصل إن شاء الله .

المرحلة القادمة ستكون أكثر إثارة ...لكنها ممنوعة عن اللذين لم ينجزوا وظائفهم !!!