

بسم الله الرحمن الرحيم

الحمد لله و الصلاة و السلام على رسول الله..

اما بعد ..

سوف أحاول بعون الله تعالى أن أقوم بترجمة سلسلة الدروس التعليمية بإسم 2D Xna Tutorial Using C#, "تعلم 2D Xna باستخدام C#".

رابط الدروس الأصلية <http://www.riemers.net/eng/Tutorials/XNA/Csharp/series2d.php>

مؤلف السلسلة يدعى Riemer Grootjans <http://www.riemers.net> و هو حاصل على شهادة

“Microsoft MVP Award 2007 - 2008 DirectX - XNA”

كما أنه مؤلف كتاب [XNA 2.0 Game Programming Recipes](#)

بسم الله نبدأ السلسلة التعليمية.

الهدف من هذه السلسلة هي تزويد المتعلم لل Xna بالمعلومات الأساسية التي تلزمه للوصول إلى مرحلة القدرة على بناء أي لعبة مبنية على البعدين “2D”.

من أجل الوصول إلى هذا الهدف سوف يتم على مدى السلسلة تطوير لعبة “Shooters” على مدى اللعبة, مما يعني أنك في النهاية سوف تصل إلى مرحلة إتقان ال Xna + القدرة على تطوير الألعاب بشكل جيد جدا إن شاء الله.

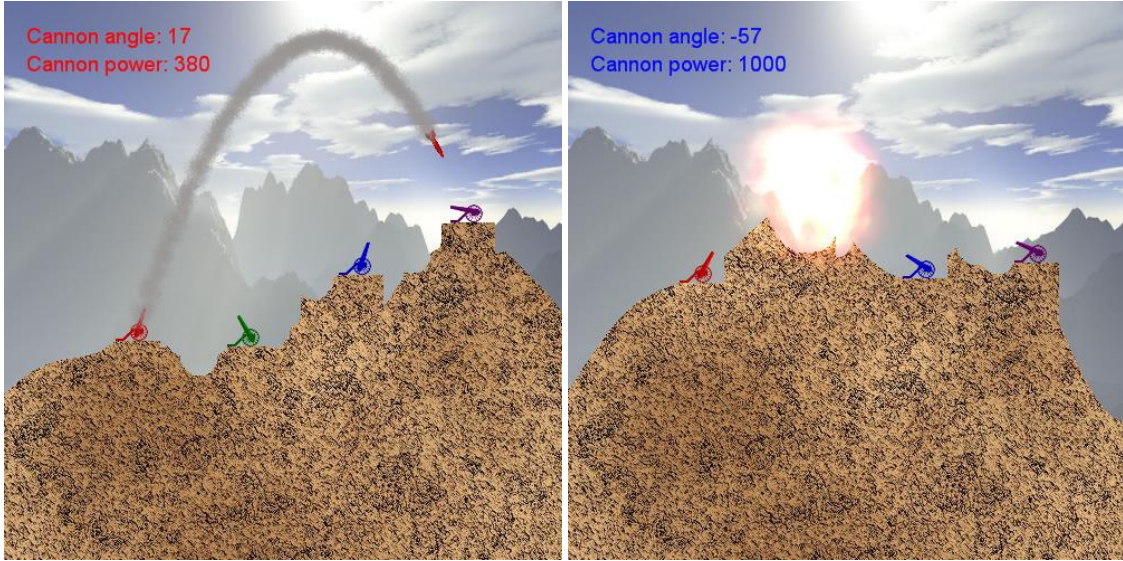
لن يتم التطرق إلى مفاهيم الألعاب بشكل نظري و لن يتم الحديث عن الجوانب النظرية المتعلقة بمكتبة ال Xna, طبعاً هذا لا يعني أن هذه السلسلة غير مخصصة للمبتدئين , لأنها مخصصة للمبتدئين فعلاً, الذين انتمى إليهم بطبيعة الحال (المترجم ☺).

سأقوم بسررد بعض من المفاهيم التي سوف تتعلمها بإذن الله في نهاية هذه السلسلة من الدروس:

- تنزيل و تثبيت ال Xna و البدء بمشروعك الأول
- إضهار وتصيير “Render” صور ثنائية البعد على الشاشة
- تحجيم و تدوير و تغيير مواضع الصور الثنائية البعد.
- مدخلات لوحة المفاتيح
- تشغيل مؤثرات صوتيه في ال Xna
- معالجة الخامات Per-Pixel
- إنشاء التضاريس العشوائية
- مزج ألفا “Alpha Blending”

- إكتشاف التصادمات (اكثر المواضيع صعوبة التي سوف تناقش هنا هي Per-Pixel Transformed)
- إضافة إلى شرح كامل لمحرك الجزيئات ثنائي الأبعاد "2D Particle Engine" من أجل الانفجارات.

صور من اللعبة التي سوف يتم تطويرها بإذن الله.



-من المترجم-

سوف أحاول ان اترجم المصطلحات القابلة للترجمة حسب الأولويات التالية:

- المصطلحات التي تم ترجمتها في موضوع "دعوة عامة لتوحيد المصطلحات"
- <http://www.arabteam2000-forum.com/index.php?showtopic=149732>
- سأحاول استخدام الترجمات العامة للمصطلحات (الترجمات الأكثر إنتشارا)
- إضافة إلى كل ذلك سوف اقوم بوضع المصطلح الإنجليزي إلى جانب العربي في علامتي تنصيص ""

التوكل على الله ...

الدرس الأول

أهلاً بكم في الدرس الأول من سلسلة دروس تعلم الـ Xna, هذه السلسلة من الدروس تستهدف الأشخاص الذين لم يسبق لهم اي معرفة في الـ Xna, و يرغبون بالحصول على نتائج بأسرع وقت ممكن.

الـ Xna هي "بيئة" برمجية , تم إصدارها في ديسمبر 2006 بواسطة مايكروسوفت , هذه التقنية مبنية "حول" على الـ DirectX , بحيث تسهل برمجة الألعاب بعدة طرق.

المتطلبات البرمجية (مجانية) :

البرمجيات اللازمة لكي تبدأ كتابة كود الـ Xna هي مجانية تماماً , و بإمكانك تنزيلها:

- Microsoft XNA Game Studio 2.0, the programming environment (free) ([link](#))
- Microsoft Visual Studio C# Express (free) ([link](#)). XNA Game Studio 2.0 will also work with the full version of Visual Studio 2005.

تأكد من إختيارك لنسخة الـ C# المعلمة بالأخضر.

إبدأ مشروع Xna جديد:

باستخدام هذه البرمجيات التي قمنا بتنزيلها , تستطيع أن تبدأ بكتابة برامجك في الـ Xna ,

تستطيع بدء الـ Xna Game Studio 2.0 , الموجوده في قائمة إبدأ , وبعدها إختار قائمة ملف => و اختر مشروع جديد (New Project) .

إذا كنت تملك نسخه كاملة الفيجوال ستديو , بإمكانك فتح مشروع بالطريقه العاديه و إختيار Xna Game Studio 2.0 من قائمة المشاريع على اليسار. نحتاج نوع مشروع (Windows Game (2.0.

نشر اللعبة إلى الـ Xbox360 سوف يتم مناقشته فيما بعد. الآن أكتب اسم اللعبة في خانة إسم المشروع انا اخترت "XNAtutorial" بعد ذلك إضغط موافق OK !

هكذا تم إنشاء مشروع Xna . في متصفح المشروع "Solution Explorer" على يمين الشاشة تستطيع ان ترى أن المشروع يحتوي على ملفين للكود هما "Game1.CS" و "Program.CS" تستطيع مشاهدة ما يحتويانه من كود عن طريق الضغط على بالزر الأيمن للماوس و إختيار "View Code". عندما تريد ان تشغل المشروع سوف يبدأ البرنامج التنفيذ من ملف الـ "Program.CS" , في الدالة "Function" الرئيسيه "Main". ببساطة هذه الدالة تقوم بإستدعاء الكود الموجود في ملف الـ "Game1.CS". بكل الأحوال لا يلزمنا تغيير اي شيء في ملف الـ "Program.CS" .

تركيب البرنامج :

افتح ملف ال "Game1.CS". ستجد انه يحتوي على العديد من الملاحظات "Comments" باللون الأخضر , (بإمكانك حذفها), بإمكاننا الآن إستكشاف تركيب برنامج لعبة ال Xna .

- الدالة المنشئة "Constructor Function" المسماه "Game1()" و التي يتم إستدعائها عند بدء التشغيل. بشكل عام تستعمل هذه الدالة لتحميل بعض المتغيرات اللازمة بواسطة إطار عمل ال Xna .
- الدالة الإستهلالية "Initialize()" و التي يتم إستدعائها ايضا عند بدء تشغيل اللعبة. هذه هي الدالة التي تلزم لنضع فيها الكود الخاص بالتجهيز للعبة "Initialization".
- الدالة "LoadContent()", و التي تستخدم لإستيراد الموارد اللازمة للعبة (مثل الصور و العناصر و الأصوات) إضافة إلى البيانات المتعلقة ببطاقة الرسومات .
- الدالة "UnLoadContent" هي المكان المناسب لإلغاء تحميل كل ما ذكر سابقا (إن لزم).
- الدالة "Update()" هي الدالة التي يتم إستدعائها في كل إطار من إطارات سير اللعبة "Frame" و التي تساوي بالضبط 60 مره. في هذه الدالة سوف نضع الكود الخاص بنا اللازم لتحديث محتوى اللعبة أثناء تشغيلها, على سبيل المثال الكود الذي يقرأ من لوحة المفاتيح, و الكود الذي يقوم بتحديث العناصر الهندسية في المشهد.
- الدالة "Draw()" و التي يتم إستدعائها كلما سمح الكمبيوتر (أو بشكل أخص بطاقة الرسومات) . حيث يتم في هذه الدالة وضع الكود اللازم لعملية رسم محتويات المشهد على الشاشة بشكل فعلي.

كما ترى فإنه لايلزمنا كود من أجل فتح شاشة معينة (مثل ال DirectX), حيث سوف تتم هذه الخطوة بشكل تلقائي. عندما تقوم بتشغيل البرنامج من خلال الضغط على "F5", سوف ترى شاشة زرقاء جميلة و جاهزه ☺ .

دعنا الآن ننقل لمناقشة جهاز الرسومات "Graphics Device". بشكل ملخص , جهاز الرسومات هو رابط مباشر إلى بطاقة الرسومات او ال "Graphical Adapter". حيث يمثل ال "Graphics Device" كائن "Object" يمكنك وصول مباشر إلى قطع من ال "Hardware" داخل جهازك. نستطيع تعريف المتغير بسهولة في الكود كمتغير من نوع "GraphicsDevice", و لأننا سوف نستخدم هذا المتغير بشكل كبير سوف نقوم بعمل إختصار لهذا المتغير. اولا و نقوم بتعريف هذا المتغير, من خلال إضافة السطر التالي في بديء الصنف "Class" بالضبط فوق الدالة "Game1()":

```
GraphicsDevice device;
```

بشكل واضح نحتاج لملى هذا المتغير, و ذلك من خلال وضع السطر التالي في الدالة "LoadContent":

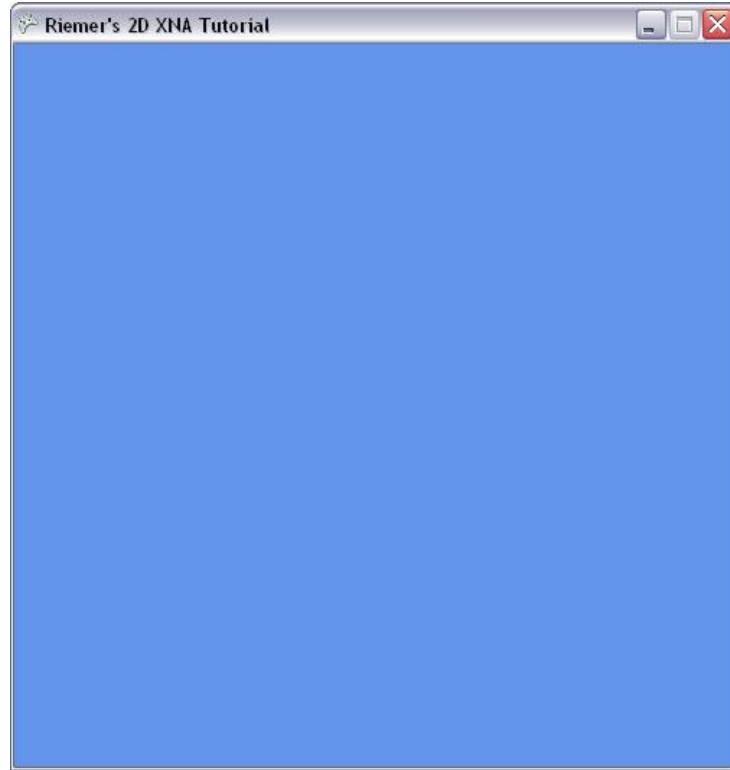
```
device = graphics.GraphicsDevice;
```

بعد ذلك نحتاج إلى تحديد أشياء إضافية متعلقة بالشاشة , مثل حجم الشاشة و عنوانها, طبعا ذلك يتم بوضع الكود التالي في الدالة "Initialize":

```
graphics.PreferredBackBufferWidth = 500;
graphics.PreferredBackBufferHeight = 500;
graphics.IsFullScreen = false;
graphics.ApplyChanges();
Window.Title = "Riemer's 2D XNA Tutorial";
```

في السطر الأول و الثاني يتم تحديد حجم الذاكرة الخلفية "BackBuffer" , التي تحتوي على ما سوف يتم رسمه على الشاشة. في السطر الثالث نقوم بتحديد فيما إذا كنا نريد تشغيل اللعبة في وضع ملئ الشاشة ام لا, بعد ذلك نقوم بتطبيق التغيرات. في السطر الأخير يتم تحديد عنوان الشاشة.

الآن قم بتشغيل البرنامج "F5" يجب ان يظهر لك شاشة بحجم 500 * 500 بكسل , مع العنوان الذي قمت بوضعه:



بعد كل درس سوف احاول ان اقترح بعض المسائل, لمساعدتك على ممارسة ما قمت بتعلمه في هذا الدرس. بعد المسائل سوف اقوم بوضع كامل الكود للمشروع مع التغيرات التي قمنا بها في الدرس. سوف اقوم بحذف الملاحظات "Comments" لذلك سوف يبدو الكود مضغوط قليلا.

ملاحظة هامة:

إذا قمت بعمل نسخ و لصق للكود في ملف ال "Game.CS" الخاص بك, تأكد من قيامك بتغيير اسم ال "Namespace" في ملف ال "Program.CS" الخاص بك إلى "XNAtutorial" (إذا لم يكن كذلك اصلا). ال "Namespace" هو نفس الأسم الذي قمت بتحديدك عند قيامك بإنشاء مشروع جديد.

- بإمكانك تجربة حل المسائل التاليه , بناء على ما تعلمته من الدرس:
- غير حجم الشاشة إلي 600 * 800.
- إجعل اللعبة تعمل في وضع ملئ الشاشة بدلا من وضع الشاشة العادية (إستخدم Alt + F4 لإيقاف البرنامج).

كود المشروع حتى الآن :

```
using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;

namespace XNAtutorial
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;
        GraphicsDevice device;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";
        }

        protected override void Initialize()
        {
            graphics.PreferredBackBufferWidth = 500;
            graphics.PreferredBackBufferHeight = 500;
            graphics.IsFullScreen = false;
            graphics.ApplyChanges();
            Window.Title = "Riemer's 2D XNA Tutorial";

            base.Initialize();
        }

        protected override void LoadContent()
        {
            device = graphics.GraphicsDevice;
            spriteBatch = new SpriteBatch(device);
        }

        protected override void UnloadContent()
        {
        }

        protected override void Update(GameTime gameTime)
        {
            if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
                this.Exit();

            base.Update(gameTime);
        }

        protected override void Draw(GameTime gameTime)
        {
            graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

            base.Draw(gameTime);
        }
    }
}
```

