

بسم الله الرحمن الرحيم

الحمد لله و الصلاة و السلام على رسول الله..

اما بعد ..

بعد الإنتهاء بحمد الله من ترجمة سلسلة الدروس "تعلم برمجة الألعاب الثنائية الأبعاد بإستخدام Xna" سوف أحاول بعون الله تعالى أن أقوم بترجمة أولى السلاسل التعليمية الخاصة ببرمجة الألعاب الثلاثية الأبعاد.

رابط الدروس الأصلية <http://www.riemers.net/eng/Tutorials/XNA/Csharp/series1.php>

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

“Microsoft MVP Award 2007 - 2008 DirectX - XNA”

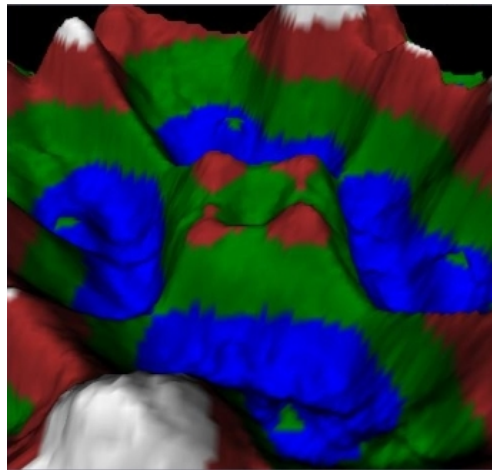
كما أنه مؤلف كتاب [XNA 2.0 Game Programming Recipes](#)

المترجم: خلدون خالد، الفريق العربي للبرمجة <http://arabteam2000-forum.com/>

بسم الله نبدأ السلسلة التعليمية : سلسلة دروس تعلم برمجة الألعاب ثلاثية الأبعاد في ال Xna (السلسلة الأولى).

هذا الجزء من الدروس يبين مدى سهولة تصميم و تشغيل برنامج بإستخدام ال Xna! هذه السلسلة التعليمية تستهدف الأشخاص الذين لم يسبق لهم التعامل مع برمجة الألعاب الثلاثية الأبعاد و يرغبون بمشاهدة النتائج بأسرع وقت ممكن. من أجل ذلك فإن ال Xna هي أفضل مكان لعمل ذلك. بإستخدام ال C# كلغة برمجة، كود ال C# يبدو شبيها جدا بالجافا، لذا أي شخص لديه خلفية في الجافا بإمكانه أن يبدأ بسهولة. إضافة إلى ذلك، هذه السلسلة كتبت بطريقة تجعل من السهل على أي شخص بغض النظر عن معرفته البرمجية فإنه بإمكانه أن يفهما و يتقن ما فيها بشكل تام!

هذه صورة للنتيجة النهائية التي سوف نصل لها في هذه السلسلة (السلسلة الأولى):



هذه السلسلة الأولى من دروس ال Xna سوف تعطيك مقدمة عامة عن البرمجة ثلاثية الأبعاد في ال Xna. تم تقسيمها إلى دروس، كما هو موضح في الأسفل. في كل درس سوف تتعلم ميزة بسيطة جديدة في ال Xna:

- بدء مشروع جديد: تجهيز و استخدام بيئة التطوير
- ملف التأثير "effect file": المؤثرات تلزم من أجل رسم الأشياء على الشاشة
- المثلث الأول: تعريف الرؤوس، و عرضها باستخدام Xna
- الفضاء: تعريف نقطة في الفضاء الثلاثي الأبعاد، تحديد موقع الكاميرا
- نقل و تدوير: تدوير و تحريك المشهد
- الفهارس "Indices": حذف معلومات الرؤوس المكررة من أجل تقليل كمية نقل البيانات في ال AGP/PCIX
- أساسيات التضاريس: استيراد الإرتفاعات إلى داخل البرنامج
- التضاريس من ملف: إنشاء التضاريس من ملف صورة
- لوحة المفاتيح: قراءة مدخلات المستخدم من لوحة المفاتيح باستخدام ال Xna
- إضافة الألوان: إضافة ألوان بسيطة إلى التضاريس
- أساسيات الإضاءة: الإضاءة ممكن أن تكون معقدة؛ لكي تستطيع فهمها بشكل تام، هناك درس كامل عن موضوع الإضاءة.
- إضاءة التضاريس: هنا سوف نقوم بتطبيق ما تعلمناه في الدرس السابق، بحيث سوف نقوم بتطبيق الإضاءة على التضاريس

أثناء هذه السلسلة سوف تحتاج لتنزيل المصادر التالية:

- ملف التأثير: effect.fx
- التضاريس من ملف: heightmap.bmp

أولويات الترجمة سوف تتم بالترتيب التالي:

- المصطلحات المترجمة من قبل فريق الترجمة في المنتدى سوف يتم الإعتماد عليها أولاً. من الرابط التالي:
<http://www.arabteam2000-forum.com/index.php?showtopic=149732>
- سأحاول استخدام الترجمات العامة للمصطلحات (الترجمات الأكثر إنتشاراً)
- 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 3D Xna الأولى، هذه السلسلة من الدروس تستهدف الأشخاص الذين لم يسبق لهم أي معرفة في برمجة الألعاب الثلاثية الأبعاد 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 الـ DirectX، بحيث تسهل برمجة الألعاب بعدة طرق.

المتطلبات البرمجية (مجانية) :

البرمجيات اللازمة لكي تبدأ كتابة كود الـ Xna و 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 قمت 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 ستوديو ، بإمكانك فتح مشروع بالطريقه العاديه و إختيار Xna Game Studio 2.0 من قائمة المشاريع على اليسار. نحتاج مشروع من نوع 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 و هما "Game1.CS" و "Program.CS" تستطيع مشاهدة ما يحتويه من كود عن طريق الضغط بالزر الأيمن للماوس و إختيار "View Code". عندما تريد ان تشغل المشروع سوف يبدأ البرنامج التنفيذ من ملف الـ "Program.CS" ، في الدالة "Function" الرئيسية "Main". ببساطة هذه الدالة تقوم بإستدعاء الكود الموجود في ملف الـ "Game1.CS". بكل الأحوال لا يلزمنا تغيير أي 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 تركيبة 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 الXna .
- 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 الثانية. 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 لا يلزمنا كود من أجل فتح شاشة معينة (مثل ال DirectX) , حيث سوف تتم هذه الخطوة بشكل تلقائي. عندما نقوم 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" يمنحك 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 لمليء هذا المتغير, و ذلك من خلال وضع السطر التالي في الدالة "LoadContent" :

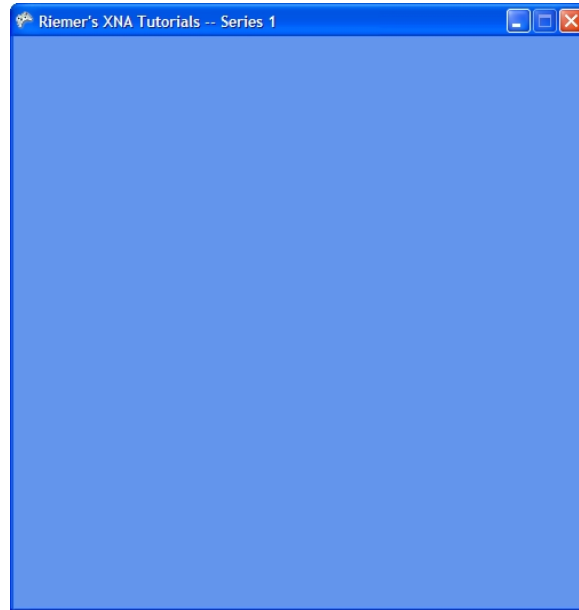
```
device = graphics.GraphicsDevice;
```

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

```
graphics.PreferredBackBufferWidth = 500;
graphics.PreferredBackBufferHeight = 500;
graphics.IsFullScreen = false;
graphics.ApplyChanges();
Window.Title = "Riemer's XNA Tutorials -- Series 1";
```

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 التمارين، لمساعدتك على ممارسة ما قمت بتعلمه في الدرس. بعد التمارين 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 بتحديدده عند قيامك بإنشاء مشروع جديد.

- بإمكانك تجربة حل المسائل التاليه , بناء على ما تعلمته من الدرس:
- غير حجم الشاشة إلى 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 XNA Tutorials -- Series 1";

            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);
        }
    }
}
```

