

الدرس الثاني

أهلاً بكم في الدرس الثاني من سلسلة دروس تعلم الـ 3D Xna السلسلة الأولى، في هذا الدرس سوف نتحدث عن ملف التأثيرات "Effect File".

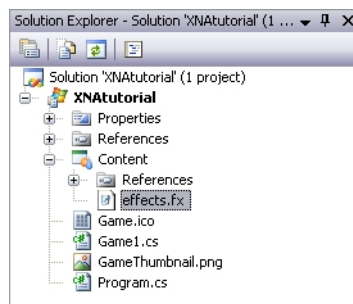
أحد الفروق الأساسية بين الـ DirectX 9 و الـ Xna أنه يلزمنا في الـ Xna تأثير "effect" لكل شيء نرسمه. إذا ماذا نعني بالتأثير بالضبط؟

في برمجة الألعاب ثلاثية الأبعاد، كل العناصر يتم تمثيلها بمثلثات. حتى الكرات بالإمكان تمثيلها باستخدام المثلثات، إذا استخدمنا كمية كافية منها. التأثير هو الكود الذي يقوم بإرشاد المكونات الفيزيائية في الجهاز (بطاقة الرسومات) كيف سوف تقوم بعرض هذه المثلثات. ملف التأثير يحتوي على طريقة (تقنية) "technique" أو عدة طرق، على سبيل المثال الطريقة A و الطريقة B. رسم المثلثات باستخدام الطريقة A سوف يؤدي إلى رسمهم بشكل نصف شفاف على سبيل المثال، بينما رسمهم بالطريقة B سوف يؤدي إلى رسمهم بالألوان الأزرق-الرمادي فقط، كما هو الحال في بعض أفلام الرعب.

لا تقلق كثيراً حيال هذا الموضوع، صحيح أن هذا متقدم قليلاً حيث سوف يتم التطرق له في السلسلة الثالثة. بكل الأحوال، الـ Xna يلزمه ملف تأثير من أجل رسم أبسط المثلثات، لذلك لقد قمت بكتابة ملف تأثير يحتوي على بعض الطرق البسيطة للرسم. بإمكانك أن تقوم بتنزيله من [هنا](#) (أو من الملفات المرفقة). اضغط بالزر الأيمن على الرابط و اختر "حفظ باسم" "Save As". يجب عليك أن تضع الملف في نفس الموقع إلى جانب ملفات الكود الخاصة بك. إذا كنت لا تعلم أين تتواجد ملفات الكود، إذهب أولاً إلى الكود، و من القائمة "File"، و اختر "Save[Project Name] As" لكي تشاهد الموقع المخزن فيه مشروعك. بشكل افتراضي، فإن ملفات الكود سوف تكون محفوظة في المسار:

```
C:\Documents and Settings\[user name]\My Documents\Visual Studio  
2005\Projects\[project name]\[project name]
```

الآن وقد قمت بتنزيل ملف التأثير في نفس مكان ملفات الكود، سوف نقوم بإستيراد الملف داخل مشروع الـ Xna. في داخل الـ "Solution Explorer" على الجهة اليمنى لشاشة الـ Xna، إبحث عن مجلد الـ "Content" و اضغط عليها بالزر الأيمن. و اختر Add->Existing Item و إنتقل إلى مكان وجود ملف الـ fx ، بعد ذلك اضغط على OK ، سوف يتم إضافة ملف الـ fx إلى مجلد الـ Content كما هو موضح في الصورة التالية:



بعد ذلك، سوف نقوم بربط ملف التأثير بمتغير، بحيث يصبح بإمكاننا إستعماله فعلياً داخل الكود. سوف نقوم بتعريف كائن من النوع Effect، إذا قم بوضع الكود التالي في بداية الـ Class:

```
Effect effect;
```

في داخل الدالة LoadContent، قم بإضافة السطر التالي لكي تقوم الـ Xna بتحميل ملف الـ fx. إلى متغير الـ effect:

```
effect = Content.Load<Effect> ("effects");
```

الإسم "effects" يشير إلى إسم الملف بدون الإمتداد "fx".

بعد أن تم تحميل كل المتغيرات اللازمة، بإستطاعتنا أن نركز على الدالة Draw. سوف تلاحظ أن السطر الأول يبدأ ب graphics.GraphicsDevice. بإستطاعتنا إستبدال ذلك بسهولة. هذا السطر يقوم بمسح الذاكرة الخاصة بالشاشة بلون معين. دعنا نحدد ذلك باللون DarkSlateBlue، للمتعة فقط 😊:

```
device.Clear(Color.DarkSlateBlue);
```

تستعمل ال Xna الذاكرة من أجل الرسم، بدلا من الرسم مباشرة إلى النافذة. في نهاية الدالة Draw، سوف يتم رسم محتويات تلك الذاكرة على الشاشة مرة واحدة. بهذه الطريقة سوف لن ترمش الشاشة كما في حالة لو رسمنا كل جزء من المشهد على حدة على الشاشة.

قم بتشغيل الكود سوف تظهر لك شاشة مثل الظاهرة في الأسفل، ولكن أريد أولا أن أقوم بإضافة بعض الكود إلى الدالة Draw أيضا. كما ناقشنا في الأعلى، من أجل رسم شيء ما على الشاشة سوف نحتاج أولا لتحديد تقنية لكائن ال Effect. سوف نقوم مباشرة بتفعيل التأثير، بحين يكون بإستطاعتنا في الدرس القادم أن نرسم شيئا على الشاشة مباشرة!

```
effect.CurrentTechnique = effect.Techniques["Pretransformed"];  
effect.Begin();
```

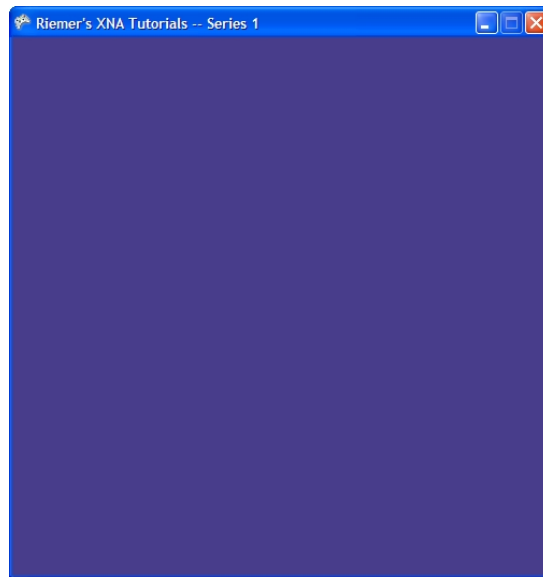
كما ترى فقد إختارنا تقنية ال Pretransformed من ملف ال effects.fx. هذه التقنية سوف نستخدمها و سناقشها في الدرس القادم. السطر الأخير يطلب من التأثير و من بطاقة الرسومات الإستعداد من أجل بعض العمل.

يتم عمل التقنية بإستخدام عدة مسوحات (مروررات) "Multiple Passes"، إذن سوف نحتاج لأن نمر عدة مرات. قم بإضافة الكود التالي تحت الكود الذي وضعته قبل قليل:

```
foreach (EffectPass pass in effect.CurrentTechnique.Passes)  
{  
    pass.Begin();  
  
    pass.End();  
}  
effect.End();
```

بإستطاعتك أن تلاحظ أن إن كل مرور يحتاج لأن نستدعي كل من الدالتين Begin و End. يجب أن يتم رسم المشهد بين هذين الإستدعائين، في النهاية، نحتاج أن نضع إستدعاء للدالة effect.End لكي نخبر المؤثر الخاص بنا اننا لا نريد أن نرسم عناصر أخرى بعد بإستخدام هذه التقنية.

أخيرا، بعد مرورنا على قسم التجهيز! إذا أحسست أنك لم تفهم أي شيء عن المؤثرات و التقنيات، لا يجب عليك أن تقلق لأننا سوف نشرح كل ذلك بشكل مفصل في السلسلة الثالثة إن شاء الله. بعد وضع كل الكود السابق، نحن جاهزون أخيرا لكي نبدأ برسم الأشياء على الشاشة، وهو ما سوف نقوم بعمله في الدرس القادم بإذن الله.



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;
        Effect effect;

        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(GraphicsDevice);

        effect = Content.Load<Effect> ("effects");
    }

    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)
    {
        device.Clear(Color.DarkSlateBlue);

        effect.CurrentTechnique = effect.Techniques["Pretransformed"];
        effect.Begin();
        foreach (EffectPass pass in effect.CurrentTechnique.Passes)
        {
            pass.Begin();

            pass.End();
        }
        effect.End();

        base.Draw(gameTime);
    }
}

```