

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 الثلاثية الأبعاد بإستخدام Xna السلسلة الأولى" سوف أقوم بعون الله تعالى بترجمة ثاني السلاسل التعليمية الخاصة ببرمجة الألعاب الثلاثية الأبعاد.

رابط الدروس الأصلية <http://www.riemers.net/eng/Tutorials/XNA/Csharp/series2.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/>

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 أقوم بوضع المصطلح الإنجليزي إلى جانب العربي في علامتي تنصيص ""

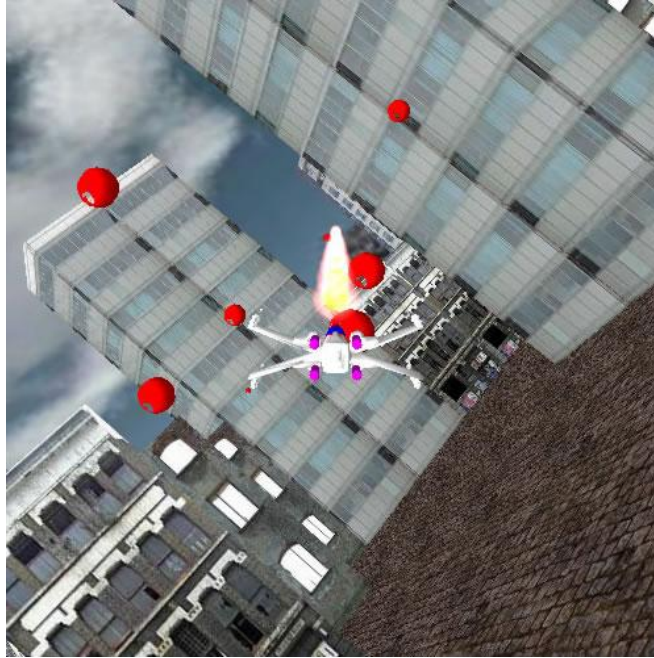
التوكل على الله ...

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

أهلاً بكم في السلسلة الثانية من دروس ال 3D Xna مع ال C#! في السلسلة الأولى، تعرفت على بعض الخواص البسيطة في ال Xna. هذه الخواص سوف يتم مناقشتها بشكل أكثر إستفاضة في هذه السلسلة، لذا بعد إكمالك لهذه السلسلة سوف تكون قد أنشئت لعبة ال 3D الخاصة بك!! 😊

في هذه السلسلة (الثانية)، سوف تتعلم كيفية إنشاء محاكي طائرة كامل. سوف يشمل ذلك جعل المركبة تطير في مدينة فعلية ثلاثية الأبعاد، كما ستقوم الطائرة بإطلاق الرصاص على الأهداف!

إليك صورة لما سوف تقوم بإنشاءه:



مره أخرى، الهدف الرئيسي من هذه السلسلة هو تغطية مفاهيم و خواص ال Xna. هذا يعني أن النموذج الفيزيائي للطائرة لن يحتوي على أي تأثير للجاذبية، لكنه يتيح لك فقط عمل المناورات بإستخدام طائرتك.

فيما يلي قائمة بالخواص التي سوف تتعلمها في هذه السلسلة (الثانية) من دروس ال Xna.

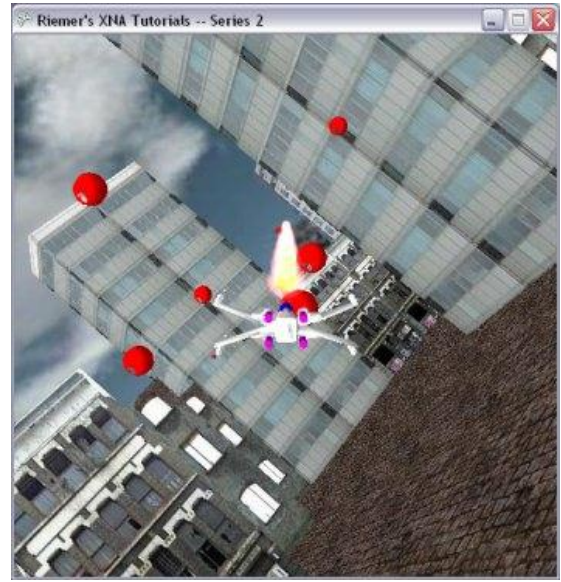
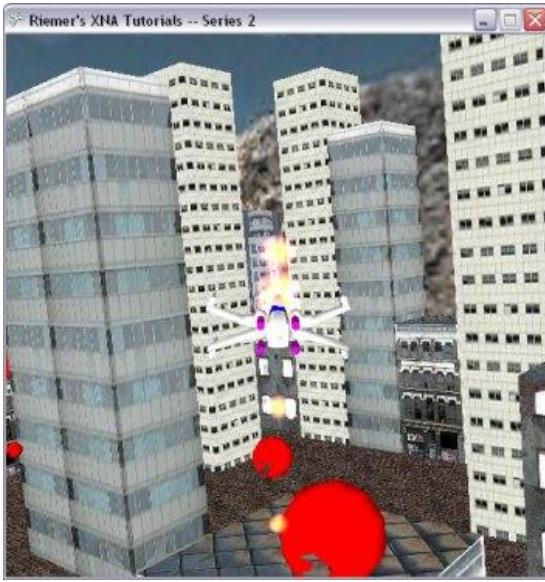
- إضافة الخامات إلى المثلثات.
- إنشاء بيئة مدينة ثلاثية الأبعاد بشكل تلقائي.
- إضافة مكعب السماء "Skybox" من أجل التخلص من الخلفية السوداء.
- نموذج طيران بسيط ولكن دقيق.
- تحريك الكاميرا.
- أشباح "Sprites" نقطية، و "Billboarding" بسيط.
- مزج ألوان.

أثناء هذه السلسلة سوف تحتاج لتنزيل المصادر (الملفات) التالية، يمكنك الحصول عليها من الملفات المرفقه أيضاً:

- نقطة البداية: effects.fx

- الخامات: [riemertexture.bmp](#)
- تحميل الأرضيات: [texturemap.jpg](#)
- نموذج الطائرة: [xwing.x](#)
- نموذج الهدف: [target.x](#)
- صورة النقطة (الرصاصية): [bullet.jpg](#)
- مكعب السماء: [skybox.x](#)
- خامات مكعب السماء: [skyboxtextures.zip](#)

فيما يلي يمكنك أن تشاهد بعض اللقطات للمشروع.



الدرس الأول

أهلاً بالجميع! مسرور لوصولكم معي إلى السلسلة الثانية من دروس ال Xna. سوف نقوم بإذن الله بتغطية بعض الخصائص الجديدة في ال Xna، جمع هذه الخواص معا في مشروع واحد، و سننتهي بمحاكي طيران حقيقي! مره أخرى، الهدف الأساسي هو شرح بعض المبادئ في ال Xna، لذا لا تتوقع الحصول على برنامج يحتوي على فيزيائيات الطيران الحقيقي، مثل الجاذبية و أشياء أخرى. لأن ذلك سيؤدي إلى إضافة كمية كبيره من أكواد الرياضيات، وسوف يشتت إنتباهنا عن ال Xna.

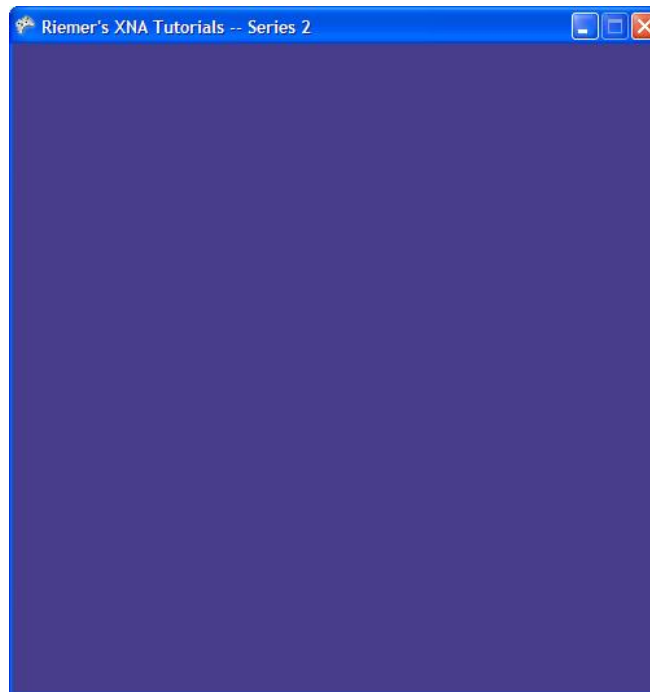
الهدف الوحيد من هذا الدرس (الأول) هو تجهيز كود البداية للمشروع. الكود يجب أن يكون بسيط جدا بالنسبة لك إذا كنت قد أنهيت السلسلة الأولى. حيث يقوم هذا الكود بـ:

- تحميل التأثير "effect"
- تحديد موضع الكاميرا
- مسح الشاشة و ذاكرة العمق "Z buffer" في الدالة Draw()

لذا قم بفتح مشروع جديد من النوع Windows Game (2.0) كما هو موضح في الدرس الأول من السلسلة الأولى. بالنسبة لي أعطيت المشروع إسم XNAseries2. أنت حر بإعطائك أي إسم مختلف للمشروع، ولكن ربما يتوجب عليك إستبدال ال namespace الموجود في الكود الخاص بي بإسم مشروعك (في حالة أنك تقوم بنسخ الكود مباشرة من الدرس). فضاء الأسماء "namespace" موجود في السطر الأول من الكود بعد عمليات الإستيراد "using".

إذا لم تقم بذلك إصلا، تستطيع تنزيل ملف التأثير الإفتراضي الذي سوف نستخدمه من [هنا](#)، حيث ستحتاج لإستيراده إلى مشروع ال Xna الذي قمت بإنشاءه، بنفس الطريقة التي تم شرحها في الدرس [الأول](#) من السلسلة الأولى. يحتوي ملف التأثير ذلك على كل التقنيات "techniques" التي سوف نحتاجها في هذه السلسلة. تذكر أنك سوف تتعلم كل شيء ستحتاج أن تعرفه عن ملف التأثير في السلسلة الثالثة بإذن الله. الآن ببساطة قم بنسخ و لصق الكود الموجود في الأسفل في ملف ال Game1.cs الخاص بك.

ترجمة و تشغيل الكود يجب أن تعطي شاشة فارغة، ملونة بلون من إختيارك بواسطة ال XNA:



كود البداية للمشروع الثاني:

```
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 XNAseries2
{
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        GraphicsDevice device;

        Effect effect;
        Matrix viewMatrix;
        Matrix projectionMatrix;

        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 2";

            base.Initialize();
        }

        protected override void LoadContent()
        {
            device = graphics.GraphicsDevice;

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

        private void SetUpCamera()
        {
            viewMatrix = Matrix.CreateLookAt(new Vector3(0, 0, 30), new
Vector3(0, 0, 0), new Vector3(0, 1, 0));
            projectionMatrix =
Matrix.CreatePerspectiveFieldOfView(MathHelper.PiOver4,
```

```
device.Viewport.AspectRatio, 0.2f, 500.0f);
    }

    protected override void UnloadContent()
    {
    }

    protected override void Update(GameTime gameTime)
    {
        base.Update(gameTime);
    }

    protected override void Draw(GameTime gameTime)
    {
        device.Clear(ClearOptions.Target |
ClearOptions.DepthBuffer, Color.DarkSlateBlue, 1.0f, 0);

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