

بسم الله الرحمن الرحيم

المحاضرة الثانية

سنقوم في هذا المثال بتحميل Texture الى مثالنا

ملاحظة : لن يتم استعمال ال oop في هذه المحاضرة و لكن سنستعمله في المحاضرة التالية J
نبدأ بأذن الله :

اول شيء يجب ان تعرف ان لكل شيء في الXna كائن خاص

فهناك كائن للرسم و كائن لتحميل الtexture و هكذا و سيتم التعرف عليهم الان

اول شيء سنتكلم عنه هو الكائنات التي سنعرفها

اول شيء الكائن الحامل للTexture الذي سيقوم بتحميل الTexture الي مثالنا J

و اسمو باين يسمى Texture فسنقوم بتعريف عند الكلاس (خارج جميع الدوال)

فنقوم بالتعريف:

```
private Texture2D Texture;
```

ثاني شيء هو الكائن الذي سيقوم برسم هذا الtexture

و هو SpriteBatch

فسنقوم بتعريفه في نفس مكان الtexture :

```
private SpriteBatch spriteBatch;
```

و نعرف كائن لمكان الtexture و طبعا بما اننا نتعامل مع شيء 2d فسيكون نوع الكائن vector2 و مقصود هنا 2D

نقوم بتعريف هذا الكائن :

```
private Vector2 position;
```

واحد بقولي طب و ليه اتعب نفسي و افضل اكتب L private مع انو غير لازم

اديك فكرة بسيطة

كلما كان الكائن تعريفه اكثر سرية كلما كان اسرع في عمل الاشياء

يعني الكائن public ابطأ من الكائن private و لكنه ببطء خفيف لا يمكن ان تشاهده لأن هنا انت بتتعامل مع

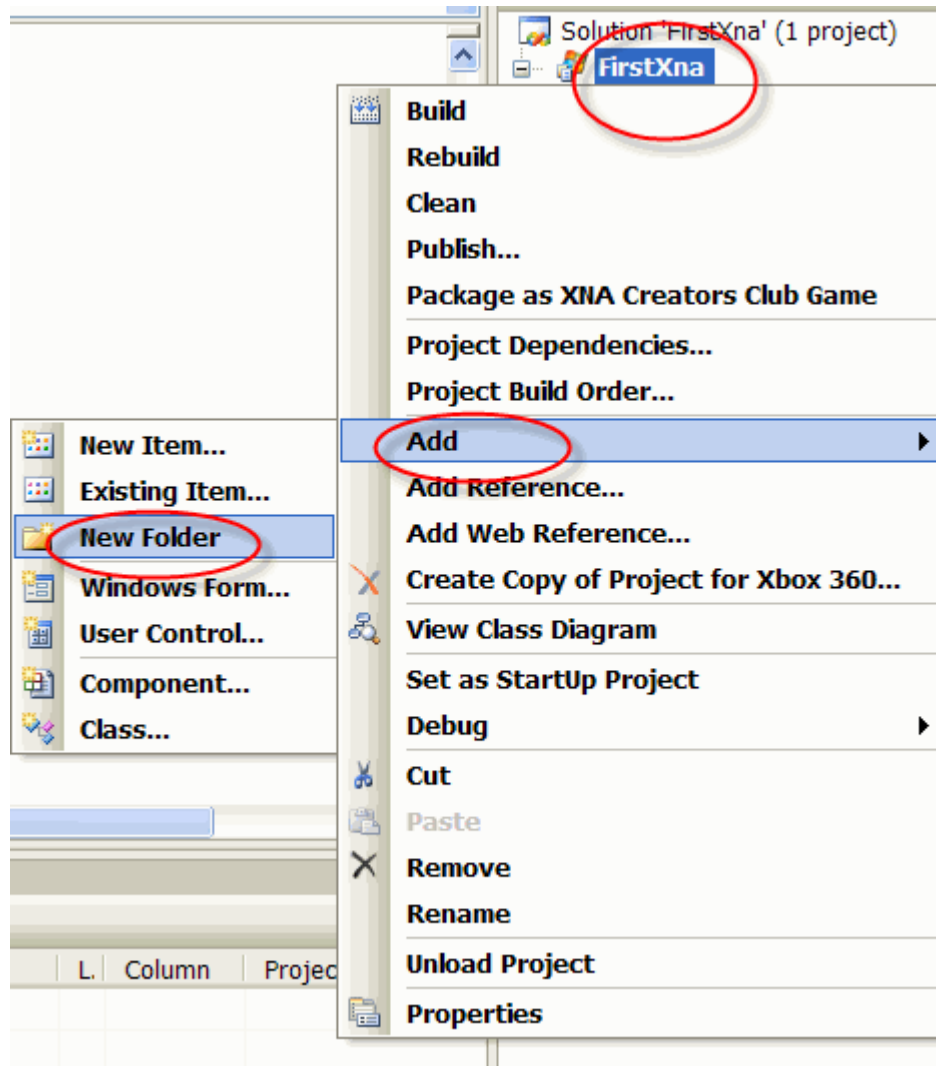
حاجات بالكيلو بايت و كمان صور لكن لو جيت تستورد مجسم 3D مشهد كامل فالسرعة هتفرق

سيكون شكل الكود النهائي :

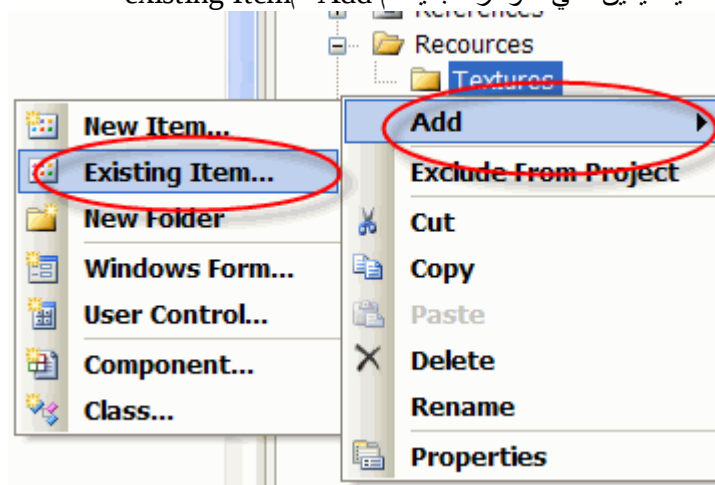
```
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
namespace FirstXna
{
    public class Engine : Microsoft.Xna.Framework.Game
    {
        private GraphicsDeviceManager graphics;
        private Texture2D Texture;
        private SpriteBatch spriteBatch;
        private Vector2 position;
        public Engine()
        {
            graphics = new GraphicsDeviceManager(this);
        }
        protected override void Initialize()
        {
            base.Initialize();
        }
        protected override void Draw(GameTime gameTime)
        {
            graphics.GraphicsDevice.Clear(Color.Black);
            base.Draw(gameTime);
        }
    }
}
```

انتهي هذا الجزء و سنبدأ بجزء تحميل الصور من الخارج

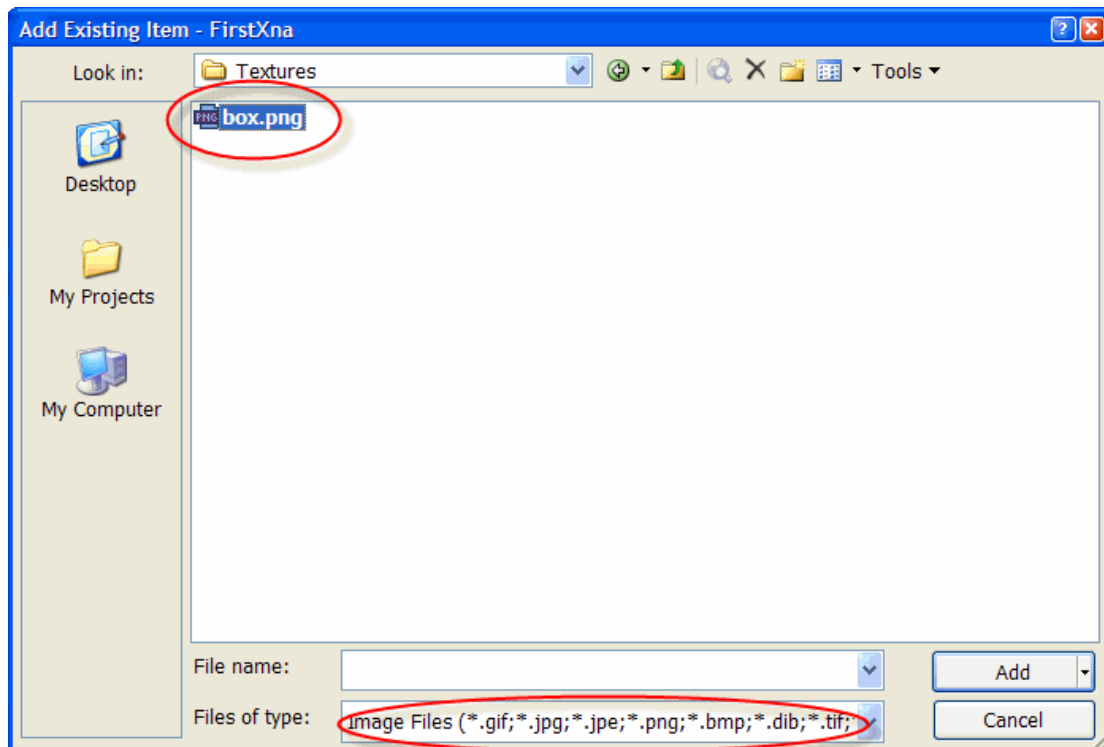
اول شيء افتح ال solution explorer و اضغط على اسم المشروع : FirstXna بلكليك يمين ثم add ثم New Folder



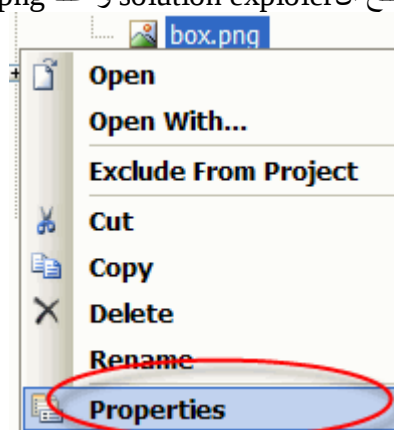
و نقوم بأعطائه اسم Resources
نضغط كليك يمين علي هذا ال Folder ثم نضغط على Add ثم New Folder و نعطيها اسم Textures
نضغط كليك يمين علي الفولدر الجديد ثم Add ثم existing Item



ستظهر لك هذه النافذة و اختار نوع الفايلات فايل لصورة و اختار الصورة المرفقة مع الدرس



الان نحن لم نفعل شيء سوى عمل فولدر جديد فيه هذه الصورة و لكننا لم نحمل هذه الصورة الى برنامجنا بعد هناك خطوة مهمة جدا حتي ينجح الكود الاتي افتح ال solution explorer و عند box.png اضغط كليك يمين ثم properties



ستظهر نافذة تحت نافذة ال solution explorer و هي نافذة الخصائص فهي تحمل خصائص الكائن المختار حول الخاصية copy to output من copy always الى Do not copy سيجعل هذا الخيار الفولدر resources يتم نقلوا بجانب مشروعنا سنكتب الان دالة سنقوم فيها بتحميل اى شيء يحتاج الى تحميل مثل ال textures

نأتي بعد دالة Initialize و نكتب protected بعدها override سنجد بعض الخيارات ظهرت نختار الخيار LoadGraphicsContent سنجد كود اتكتب او توماتيك كالاتي :

```
protected override void LoadGraphicsContent(bool loadAllContent)
{
    base.LoadGraphicsContent(loadAllContent);
}
```

نمسح الكود الذي بداخله و نكتب :

```

        if (loadAllContent)
        {
            Texture = Texture2D.FromFile(graphics.GraphicsDevice,
"Resources\\Texture\\box.png");
        }

```

ماذا فعلنا هنا؟؟

نحن الان وضعنا جمل شرطية اذا كان قيمة هذا البارامتر true فقم بتحميل هذا الtexture طبقا سيتم التحميل من ملف خارجي لهذا كتبنا FromFile و حددنا الgraphics device فى اول بارامتر و ثاني بارامتر وضعنا فيه مكان الفايل طبقا احنا كنا واضعين فولدر يسمى Resources و بداخله فولدر يسمى Texture و جعلناه يعمل كوبي لنفسوا لمكان التشغيل

لو نظرنا لرأينا ان هذه الدالة بها بارامتر من نوع bool يعني true او false نحن الان وضعنا جمل شرطية اذا كان قيمة هذا البارامتر true فقم بتحميل هذا الtexture حسنا هل هذا الكود قيمته true؟؟ انت لم تسند اليه قيمة L حسنا اذهب الى الدالة initialize و اكتب فى اول مكان فى الدالة :

```

this.LoadGraphicsContent(true);

```

سيكون شكل الكود الان

```

using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
namespace FirstXna
{
    public class Engine : Microsoft.Xna.Framework.Game
    {
        private GraphicsDeviceManager graphics;
        private Texture2D Texture;
        private SpriteBatch spriteBatch;
        private Vector2 position;
        public Engine()
        {
            graphics = new GraphicsDeviceManager(this);
        }
        protected override void Initialize()
        {
            LoadGraphicsContent(true);
            base.Initialize();
        }
        protected override void LoadGraphicsContent(bool
loadAllContent)
        {
            if (loadAllContent)
            {
                Texture = Texture2D.FromFile(graphics.GraphicsDevice,
"Resources\\Textures\\box.png");
            }
        }
        protected override void Draw(GameTime gameTime)
        {
            graphics.GraphicsDevice.Clear(Color.Black);
            base.Draw(gameTime);
        }
    }
}

```

الان تم الانتهاء من مرحلة تحميل الملفات الان نذهب الى مرحلة اعطاء كل متغير قيمته مثل
position و spriteBatch

سنذهب لدالة Initalize

الان ما بين السطرين

```
LoadGraphicsContent(true);
```

و

```
base.Initialize();
```

و نأتي لنعرف قيمة كل كائن

اول شيء هو ال spritebatch

فنقوم بتعريفه

```
spritebatch = new SpriteBatch(graphics.GraphicsDevice);
```

طبعا الكائن Graphics معرف في اول دالة

السؤال الان لماذا يحتاج ال spritebatch الى بارامتر من هذا النوع الان اذا قلنا له ارسم لي احد ال textures

فيجب عليك تحديد في أي graphics device سيتم الرسم و كأنك تقول في اي مشهد سيتم الرسم

ثاني شيء المتغير position فسيتم اعطائه قيمة من نوع 2d لأننا نرسم اشياء 2d و هو اساسا من نوع

vector2

```
position = new Vector2(100, 100);
```

تم الانتهاء من هذا الجزء و سيكون شكل الدالة النهائي :

```
protected override void Initialize()
{
    LoadGraphicsContent(true);
    spritebatch = new SpriteBatch(graphics.GraphicsDevice);
    position = new Vector2(100, 100);
    base.Initialize();
}
```

نأتي الى جزء الرسم و هو اهم جزء لأننا لو ضغطنا على F5 الان فلن نجد نتيجة لكل هذه

الجهود

نذهب الى الدالة Draw

الان ال spritebatch من المفترض قبل ان نستعمل دالة الرسم التي فيه علينا ان نجعله يبدأ بأستعمال Begin

يعني بعد الكود الى كتبناه : graphics.GraphicsDevice.Clear(Color.Black);

نقوم بكتابة

```
spritebatch.Begin();
```

و هذه الدالة يجب ان تقفل يعني مينفعش نجعلها تعمل begin مرتين على من دون ما نقفلها

ثم نستخدم دالة الرسم : Draw فنكتب

```
spritebatch.Draw(Texture, position, Color.White);
```

اول بارامتر يتم تحديد التكتشر المراد رسمه و ثاني بارامتر هو مكان التكتشر و ثالث بارامتر هو اللون و هنا

WHITE لا يعني ابيض بل يعني انه لن يتم اضافة لون على الصورة فحاول ان تضع مكان هذا اللون مثلا

Red فستري ان لون الصورة صار اقرب الى الاحمر



يمكننا تحديد parameters اكثر و لكن هذا لو كنا سنستعمل الدوران او هنقطع جزء من الصورة او هكذا و لكن

في هذا المثال لن تحتاج كل هذا

الان اضغط على F5 لترى نتيجة عملك J

حصل Error ^_

برأيك لما هذا ال error لو عرفت الاجابة فأعرف انك متميز

ستقول لي لأننا لم نقفل الدالة يعني لما تضع الجملة spriteBatch.End اقول لك هذا صحيح و لكن ال error يحصل لو استخدم begin مرتين و لكن هنا تم استعمالوا مرة واحدة فلم الخطأ ؟ حاول ان تعرف لو لم تعرف فلا تزعل فربما انك نسيت قليلا و لكن الاجابة في المحاضرة الاولى

٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧
٧

حسننا سأقول لك قلت لك في المحاضرة الاولى ان الدالة Draw تستدعي نفسها كل جزء من الثانية فسيتم اعادة الكود مرة اخري فسيتم استدعاء الدالة begin مرة اخري فينتج عن هذا خطأ
الحل انت تعرفه كل ما عليك هو عمل end
فنكتب في اخر الكود spriteBatch.End();
يكون شكل الدالة النهائي

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.Black);
    spriteBatch.Begin();
    spriteBatch.Draw(Texture, position, Color.White);
    spriteBatch.End();
    base.Draw(gameTime);
}
```

الان اضغط على F5 و شوف نتيجة عملك J J J



حسنًا ربما تحس أنك لم تفعل شيء و لكن طريق الالف ميل يبدأ بخطوة و هذه هي اول خطوة عليك ان تفرح بأنجازك و ان شاء الله سنتقدم فى هذه الدورة
تم الانتهاء من المحاضرة الثانية و اسأل الله التوفيق لى و لكم

للتواصل Gindisgindis@hotmail.com