



TI24X7

CURSOS ONLINE EM TECNOLOGIA

Curso de XNA Desenvolvimento de Jogos

Módulo 2

Autor: Fernando Amaral

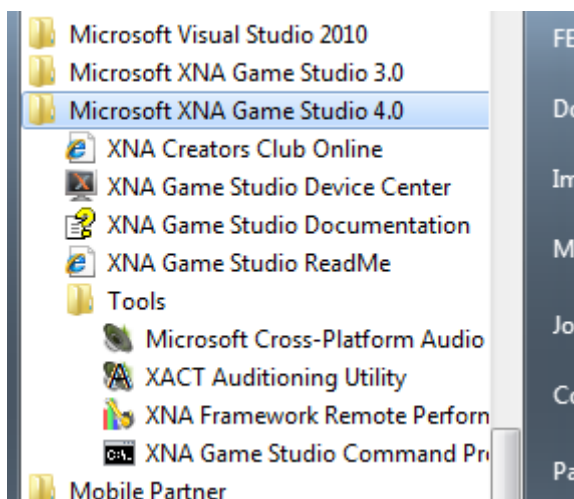
2: Conhecendo um Projeto de Jogo XNA 4.0

Sei que você está louco para escrever código e ver seu primeiro jogo em XNA nascendo. O momento está próximo, e eu garanto! Será recompensador. Porém antes, você deve conhecer a estrutura de um projeto XNA, do que é composta a solução e seus projetos e como é o ciclo de vida de uma jogo XNA. Acredite, este capítulo vai tornar o processo de aprendizado mais fácil e intuitivo e conseqüentemente os resultados virão de forma mais natural. Por isto, perca alguns minutos conhecendo a arquitetura de um jogo em XNA, para, no próximo capítulo, começar a diversão!

Criando um projeto de Jogos no Visual Studio

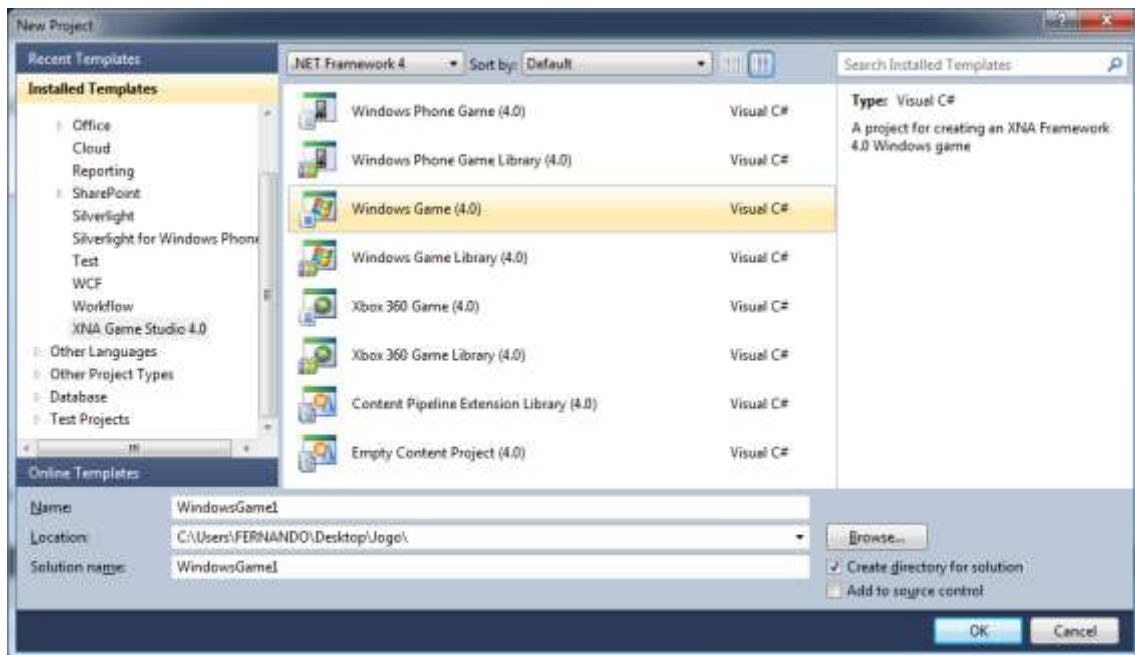
O processo de instalação do Microsoft Visual Studio e do XNA Game Studio é bastante simples e intuitivo, por isso vamos pular o procedimento de instalação e ir direto ao que interessa.

Após a instalação do XNA Game Studio uma nova pasta é adicionada ao seu menu iniciar. Nesta pasta você vai encontrar um atalho para o XNA Creators Club Online, a ajuda do Framework e diversas ferramentas de apoio. Veja que você não vai desenvolver jogos por aqui, já que, como falamos, o XNA funciona integrado ao Visual Studio, na forma de um complemento.



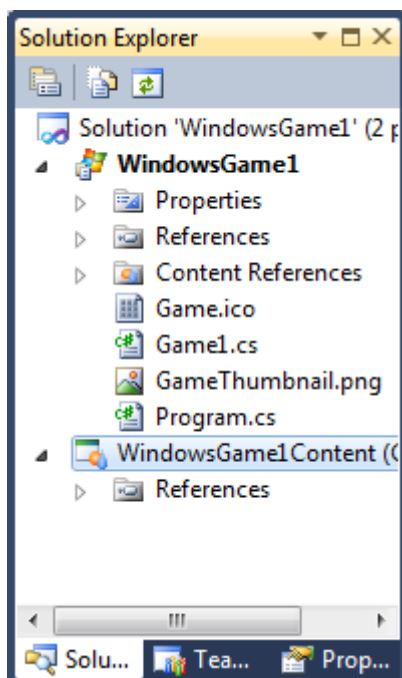
Então abra o Visual Studio 2010 – ou o C# 2010 Express Edition. Acesse File/New/Project. Na janela New Project, selecione XNA Game Studio 4.0. Você verá diversos templates para a criação de jogos em XNA. Um template é um projeto de aplicação com a estrutura básica necessária para o aplicativo, de forma que você iniciar imediatamente a codificação.

Curso de XNA Desenvolvimento de Jogos – Módulo 2



Nesta seção não vamos codificar nada, o objetivo será apenas entender a estrutura de um Jogo XNA. Selecione o Template Windows Game (4.0), mantenha o nome padrão e clique em Ok.

Veja que o Visual Studio criou uma solução com dois Projetos: Windows Game1 e WindowsGame1Content:



Curso de XNA Desenvolvimento de Jogos – Módulo 2

WindowsGame1 é aplicação do jogo propriamente dita. O Projeto WindowsGame1Content é um projeto de conteúdo, onde todos os recursos do jogo, como imagens, áudio, fontes entre outros, serão armazenados. Aliás, esta é uma das grandes diferenças entre o XNA 3.0 e o 4.0, quando o conteúdo era armazenado em uma pasta específica dentro do mesmo projeto.

Trataremos mais do projeto de conteúdo e do ContentPipeline mais adiante.

Veja no projeto principal que temos dois arquivos CS. Program.cs é arquivo de entrada da aplicação. Observe o código e veja que o procedimento principal da aplicação recebe um tipo Game1, executando o método Run. Você não vai precisar alterar este arquivo, a não ser em casos específicos em que, por exemplo, quer tratar algum parâmetro de entrada na execução de seu jogo. No mais, você precisa saber que este arquivo existe e qual a sua função.

Agora vamos analisar o arquivo Game1.cs. Este é o arquivo principal de nosso jogo, e por isso vamos gastar alguns minutos para analisá-lo.

Veja inicialmente que são incluídos diversos namespaces Microsoft.Xna.Framework. Cada um destes namespaces possuiu um conjunto específico de classes para uso no desenvolvimento de jogos. Quais funcionalidades você encontra em cada namespace deve estar óbvio pelo nome do mesmo:

```
using System;
using System.Collections.Generic;
using System.Linq;
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.Media;
```

Em nível de classe temos duas variáveis declaradas, uma do tipo GraphicsDeviceManager e outra do tipo SpriteBatch

```
GraphicsDeviceManager graphics;
SpriteBatch spriteBatch;
```

O tipo GraphicsDeviceManager terá as configurações da placa de vídeo onde o jogo estará sendo executado. Através de uma instancia deste tipo você também configurar o vídeo, como por exemplo, a resolução do jogo. O tipo SpriteBatch permite que um grupo de sprites sejam desenhados na tela ao mesmo tempo. Já falamos sobre o conceito de sprite: qualquer objeto animado de seu jogo. Estas variáveis vêm automaticamente no projeto porque, qualquer jogo terá que trabalhar com um dispositivo gráfico e manipular sprites.

Continuando a análise do código, vemos o construtor da classe Game1, que inicializa a variável graphics e define o diretório padrão de conteúdo.

```
public Game1()
{
```

Curso de XNA Desenvolvimento de Jogos – Módulo 2

```
graphics = new GraphicsDeviceManager(this);
Content.RootDirectory = "Content";
}
```

Os próximos cinco métodos formam o ciclo de vida do Jogo: Initialize, LoadContent, Update, Draw e Unload Content. Perca alguns minutos analisando o bloco de código abaixo:

```
protected override void Initialize()
{
    // TODO: Add your initialization logic here

    base.Initialize();
}

protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // TODO: use this.Content to load your game content here
}

protected override void UnloadContent()
{
    // TODO: Unload any non ContentManager content here
}

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
        ButtonState.Pressed)
        this.Exit();

    // TODO: Add your update logic here

    base.Update(gameTime);
}

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

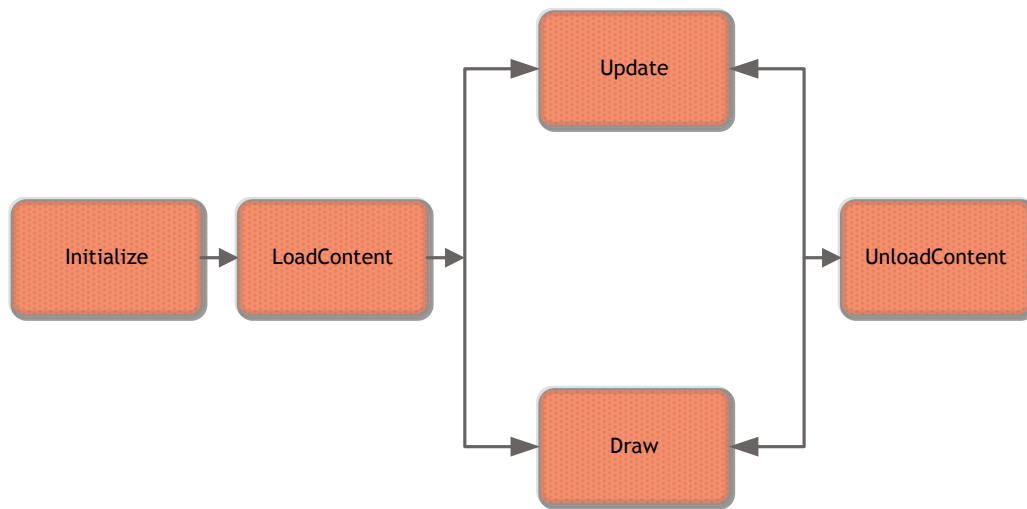
    // TODO: Add your drawing code here

    base.Draw(gameTime);
}
```

O método Initialize deve ser usado para inicializarmos todos os tipos que iremos usar em nosso jogo. Em LoadContent, o conteúdo, como sprites, áudio entre outros, devem ser carregado para posterior uso no jogo. Estes dois métodos serão executados uma única vez durante todo o jogo. Já Update e Draw serão executados intermitentemente enquanto durar o Jogo. Em update devemos escrever todo o código de atualização: Definir a nova posição do herói, verificar se ocorreu alguma colisão, atualizar o placar. Em Draw todos os elementos gráficos do jogo devem ser desenhados. Finalmente, um UnloadContent devemos liberar todo o conteúdo

Curso de XNA Desenvolvimento de Jogos – Módulo 2

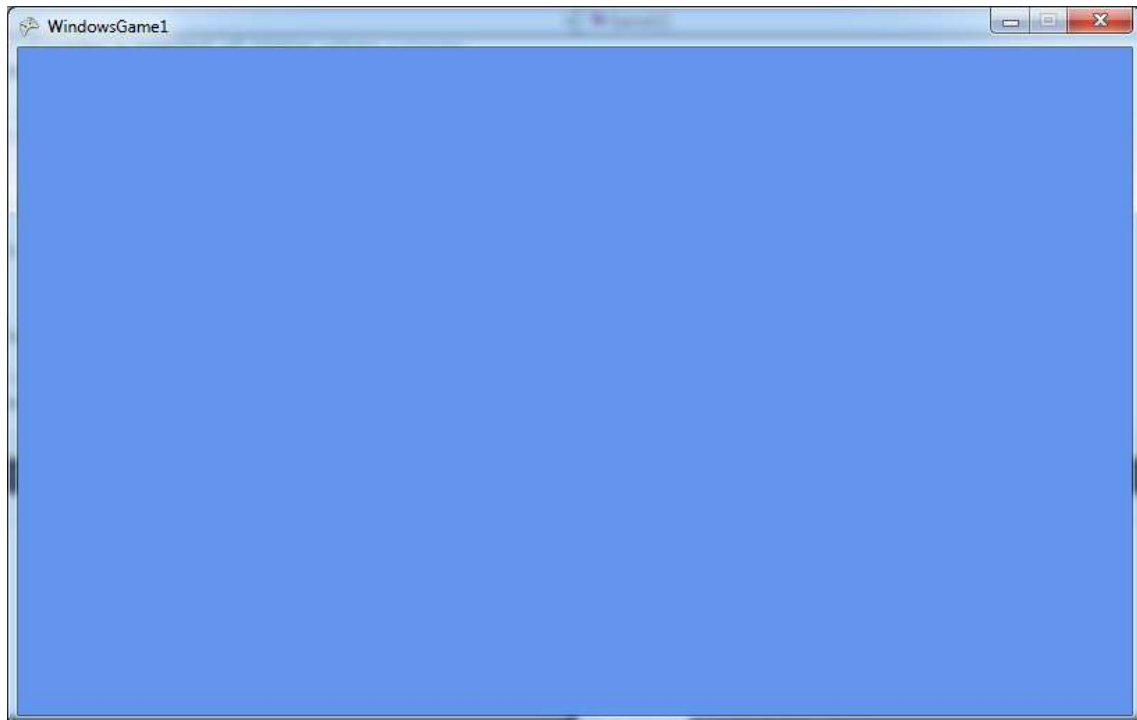
que usamos na execução do jogo, este método também será executado uma única vez, é o fim do jogo!



O código inicial do método Update faz com o jogo seja encerrado no XBOX 360. No método Draw, a tela é pintada de azul. É no método Draw que a magia do jogo acontece. A tela de seu jogo é atualizada 60 vezes por segundo, cada atualização é conhecida tecnicamente como Frame. A taxa de atualização, é o FrameRate de seu jogo. Um jogo XNA atualiza a tela 60 vezes por segundo, ou seja, o FrameRate do seu jogo será, em média, de 60 fps (frames per second). Para nossos olhos, esta atualização é imperceptível, movendo sprites entre frames dará a nossos olhos a ilusão de uma animação!

Você pode rodar a aplicação neste momento, o resultado será uma tela azul:

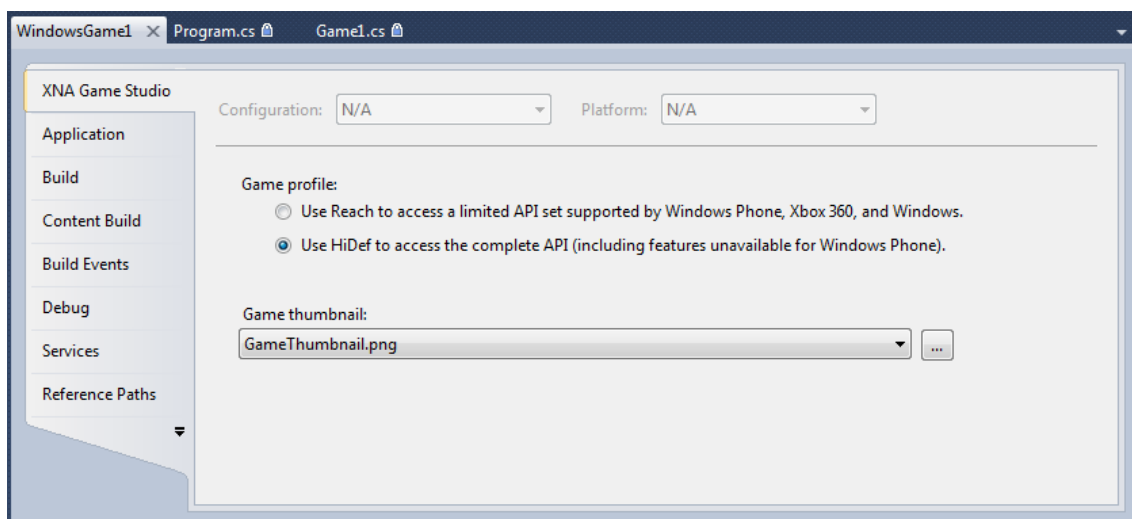
Curso de XNA Desenvolvimento de Jogos – Módulo 2



Não podíamos esperar mais do que isto, ainda não digitamos uma linha de código!

Perfil do Jogo

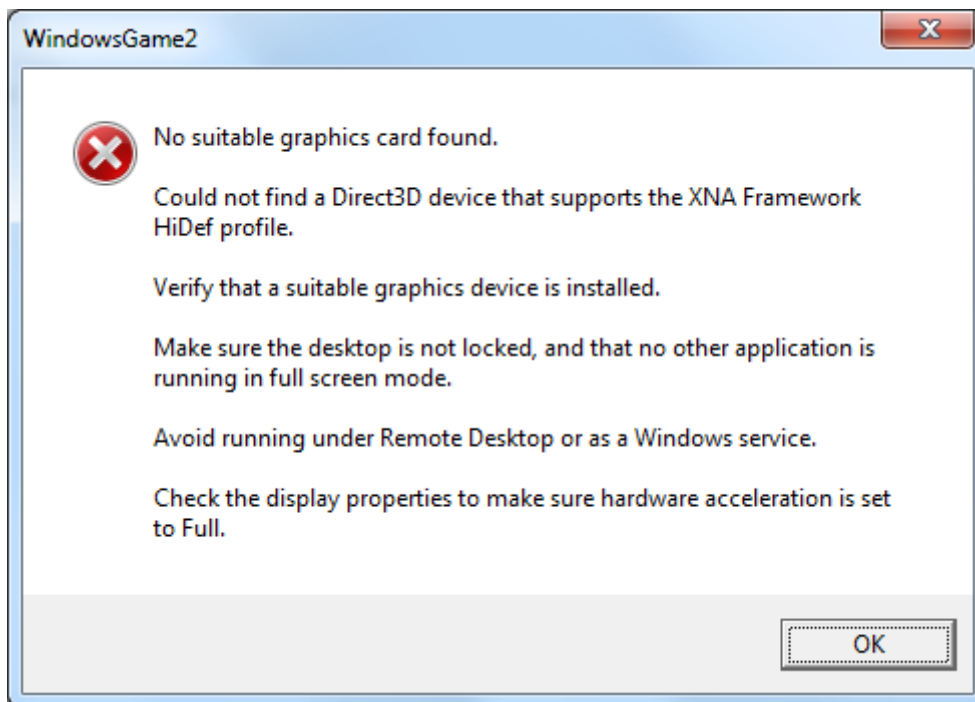
Vamos conhecer um pouco melhor a estrutura de uma aplicação XNA antes de iniciarmos a codificação. Clique com o botão direito sobre o projeto WindowsGame1, e selecione a opção Properties. Você verá as configurações abaixo:



Curso de XNA Desenvolvimento de Jogos – Módulo 2

Uma configuração importante, é que você deve conhecer, é o Game Profile. O Perfil trata das características de hardware que o jogo vai utilizar. Você terá que optar por um perfil, conforme as configurações ou o próprio equipamento em que seu jogo vai ser executado. O XNA 4.0 traz dois tipos de perfis: Reach e HiDef. O primeiro suporta equipamento de alta performance, o segundo, utiliza menos recursos e deve ser compatível com qualquer equipamento. Se você está desenvolvendo para PC com suporte a DirectX10 ou equivalente ou ainda para XBOX 360, use o perfil HiDef. Se o PC tem suporte inferior ou você quer desenvolver para Windows Phone 7, use o perfil Reach.

O perfil padrão quando você cria um projeto do tipo Windows Game é o HiDef. Caso você rode um jogo em um equipamento que não tem suporte a este perfil, verá um alerta como na mensagem abaixo:



Você poderá se surpreender com a quantidade de equipamentos, mesmo recentes, que não conseguem rodar um jogo XNA usando o perfil HiDef!

Content Pipeline

Não existe um jogo sem conteúdo: Imagens, modelos 3D, sons, fontes etc. O Content Pipeline é o mecanismo de gestão deste conteúdo, transformando o mesmo em um formato acessível ao jogo em tempo de execução. Você já observou que quando criou o projeto de seu jogo, um novo projeto é criado, chamando pelo nome do projeto do jogo mais o sufixo Content. Neste projeto devemos organizar todo o conteúdo que utilizaremos no jogo, eles serão automaticamente localizados no projeto através de uma referencia que é criada automaticamente quando criamos a aplicação.

XACT

Curso de XNA Desenvolvimento de Jogos – Módulo 2

XACT, acrônimo para Microsoft Cross-Platform Audio Creation Tool, é a ferramenta para organização e configuração de áudio para um jogo XNA. Através desta ferramenta você pode organizar e também modificar arquivos de áudio, disponibilizando os mesmos ao jogo através de um único arquivo. Vamos saber mais sobre o XACT no capítulo 6, quando vamos aprender a adicionar áudio aos nossos jogos.

Conclusão

Neste capítulo estudamos a estrutura de um jogo em XNA. Agora estamos pronto para digitarmos algumas linhas de código e desenhar elementos na tela. Até o capítulo 3.