



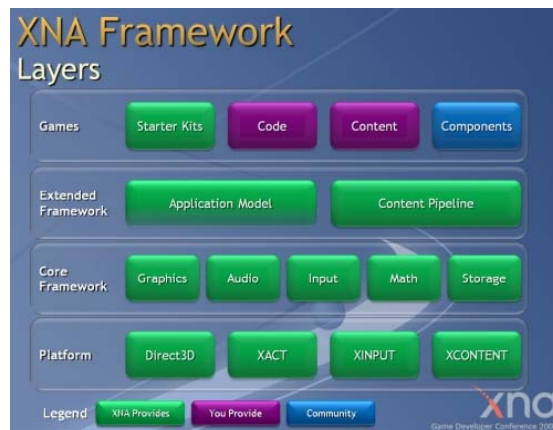
# XNA Initialization

305890  
Spring 2011  
3/14/2011  
Kyoung Shin Park

## XNA

- Microsoft XNA is a set of tools with a managed runtime environment for computer game development and management.
- XNA Framework is based on native implementation of .NET Compact Framework 2.0 for Xbox 360 development and .NET Framework 2.0 on Windows.
- XNA games can run on any platform that supports the XNA Framework with minimal or no modification.
- But, only C# in XNA Game Studio Express IDE and all versions of Visual Studio 2008 and 2010 (as of XNA 4.0) are officially supported.

## XNA Framework



## XNA 3.0



## Installing XNA

- Visual Studio 2008 Installation & Rebooting
- DirectX9.0c Runtime Installation
  - `directx_jun2010_redist.exe`
  - <http://www.softwarepatch.com/windows/directx.html>
- XNA Game Studio 3.0 Installation
  - <http://www.microsoft.com/downloads/en/details.aspx?FamilyId=7D70D6ED-1EDD-4852-9883-9A33C0AD8FEE&displaylang=en>
- Create a Platformer project & Run
  - Visual Studio 2008 메뉴에서 File->New->Projects
  - Visual C#->XNA Game Studio 3.0->Platformer Starter Kit (3.0) 선택
  - 프로젝트 이름 지정

5

## Installing XNA

- Platformer1 실행화면



## Compiling XNA Programs

- Visual Studio 2008 VC# 실행
- 프로젝트 새로 만들기
  - 메뉴에서 File->New->Projects
  - Visual C#->XNA Game Studio 3.0->Windows Game (3.0) 선택
  - 프로젝트 이름 지정
- 빌드 (build) (F7)와 실행 (execute) (F5)

## XNA Example

- 파란색 배경을 가진 프로그램 예제

```
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;  
using Microsoft.Xna.Framework.Net;  
using Microsoft.Xna.Framework.Storage;
```

```

/// <summary>
/// This is the main type for your game
/// </summary>
public class Game1 : Microsoft.Xna.Framework.Game
{
    GraphicsDeviceManager graphics;
    SpriteBatch spriteBatch;

    public Game1()
    {
        graphics = new GraphicsDeviceManager(this);
        Content.RootDirectory = "Content";
    }
    /// <summary>
    /// Allows the game to perform any initialization it needs to before starting to run.
    /// This is where it can query for any required services and load any non-graphic
    /// related content. Calling base.Initialize will enumerate through any
    /// components and initialize them as well.
    /// </summary>
    protected override void Initialize()
    {
        // TODO: Add your initialization logic here
        base.Initialize();
    }
}

```

```

/// <summary>
/// LoadContent will be called once per game and is the place to load
/// all of your content.
/// </summary>
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
}

/// <summary>
/// UnloadContent will be called once per game and is the place to unload
/// all content.
/// </summary>
protected override void UnloadContent()
{
    // TODO: Unload any non ContentManager content here
}

```

```

/// <summary>
/// Allows the game to run logic such as updating the world,
/// checking for collisions, gathering input, and playing audio.
/// </summary>
/// <param name="gameTime">Provides a snapshot of timing values.</param>
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);
}
/// <summary>
/// This is called when the game should draw itself.
/// </summary>
/// <param name="gameTime">Provides a snapshot of timing values.</param>
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    // TODO: Add your drawing code here
    base.Draw(gameTime);
}
} // end of Game1 class

```

## XNA Example

### □ 500x500 화면크기와 "Test1" 타이틀 추가

```

protected override void Initialize()
{
    // TODO: Add your initialization logic here
    graphics.PreferredBackBufferWidth = 500;
    graphics.PreferredBackBufferHeight = 500;
    graphics.IsFullScreen = false;
    graphics.ApplyChanges();
    Window.Title = "Test1";

    base.Initialize();
}

```

## XNA Example

---

- ESC-key를 누르면 프로그램을 종료하기 추가

```
// keyboard & gamepad variables
private KeyboardState currentKeyboardState = new KeyboardState();
private GamePadState currentGamePadState = new GamePadState();

// add HandleInput
protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
    HandleInput();

    // TODO: Add your update logic here
    base.Update(gameTime);
}
```

## XNA Example

---

```
// ESC-key 또는 게임패드의 버튼1이 눌렀을 경우 프로그램 종료
#region Handle Input
/// <summary>
/// Handles input for quitting the game.
/// </summary>
private void HandleInput()
{
    currentKeyboardState = Keyboard.GetState();
    currentGamePadState = GamePad.GetState(PlayerIndex.One);

    // Check for exit.
    if (currentKeyboardState.IsKeyDown(Keys.Escape) ||
        currentGamePadState.Buttons.Back == ButtonState.Pressed)
    {
        this.Exit();
    }
}
#endregion
```

## XNA Example

---

