Review of C# and XNA

What is C#?

C# is an object-oriented programming language developed by Microsoft. It is developed within .NET environment and designed for Common Language Infrastructure.

Visual C# 2010 Express

To use C# programming language, a C# language development environment is required. The environment usually consists of editor and compiler. Visual C# 2010 Express is an Integrated Development Environment (IDE) specially designed for C# programming language. Visual C# 2010 Express will be referred as VC# Express in short.
To start a programming in VC# Express, a solution needs to be created. A solution has more than one project. Each project includes information what files are related to a program and how to compile the source files.

**The first C# program**

Let’s create a simple program that shows your name in console.

Click on Windows button, and click on the Microsoft Visual C# 2010 Express button as shown below.

![Figure 2. Start Microsoft Visual C# 2010 Express](image)

After the VC# Express shows up, click on the new project button in the IDE, and you will see the following new project dialog window.
A project type should be selected based on the program that a programmer would like to create. At this time, a simple program that shows your name on a console screen will be created. A “Console Application” is a project type for this application.

After selecting “Console Application” as a project type, change the project name to “hello” then click OK button.

The initial program will show up as below.
Figure 4. Initial Visual C# 2010 Express IDE screen with Console Application type selected.

The VC# Express will populate the program with initial program codes based on the application type you selected. The program should be compiled without any further changes. The program, however, does nothing when it is compiled and executed.

In between the code “static void Main(string[] args) {” and “}”, type the following two lines with your name for example “John Doe”.

```csharp
Console.WriteLine(“Hello, your name”);
Console.ReadLine();
```

After you type the code, the program will be read like the following
A C# program should be compiled before being executed. In VC# Express, a program is compiled and executed when a programmer click on “Start debugging (F5)” button as shown in the following figure.

After VC# Express compile the program, the VC# Express will start the program. You will see a screen as shown below.

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace hello
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, Digispired ii students");
            // Console.ReadLine();
        }
    }
}
```
This program is the simplest program, but it includes all required program structures and is a complete program. Any text output program can be created with a little modification of this program.

**Application of this skills and knowledge**

Try to create programs that can display:

```
Welcome to Digispired ii!
```

```
C# programming is easy and fun.
```

```
CCC  #  #
C  C  #######
C    #  #
C  C  #######
CCC  #  #
```
Simple Animation using XNA

What is XNA?

“XNA Game Studio 4.0 is a programming environment that allows you to use Visual Studio to create games for Windows Phone, the Xbox 360 console, and Windows-based computers.” – msdn.microsoft.com

Simple XNA program

First, start Visual C# 2010 Express and click on “New Project” button.

In the “New Project” dialog box, select “Windows Game (4.0)” and type “xna_prog_1” in the Name text box as shown in below.

Figure 8. Windows Game (4.0) project
The source file in the project already filled up with basic codes, which is ready to be compiled. When the project compiled and run, a blank window will show up. It is the time to put some program codes to show as below.

![Expected xna_prog_1 program output](image)

**Naming functions/variables**

Computer programming can be a very simple to very complex process. It is a good idea to put meaningful names for functions and variables, so a programmer can remember and use them easily.

First you need to change the program’s main class name as “xna_prog_1”. Under “namespace xna_prog_1 {” you can find “public class Game1 : Microsoft.Xna.Framework.Game”. Change the “Game1” to “xna_prog_1”, and change “public Game1()” to “public xna_prog_1()” as shown below.
Also change the main class file name in the solution window from “Game1.cs” to “xna_prog_1.cs” as shown below.

When you changing a name, all related program code should be changed accordingly. In this case, the main class name has been changed to “xna_prog_1”. All names for the main class should be changed to “xna_prog_1”. In “Program.cs” file, change “using (Game1 game = new Game1());” to “using (xna_prog_1 game = new xna_prog_1());” as shown below.
**Figure 12. Change related codes in other files**

```csharp
namespace xna_prog_1
{
    #if WINDOWS || XBOX
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        static void Main(string[] args)
        {
            using (xna_prog_1 game = new xna_prog_1())
            {
                game.Run();
            }
        }
    }
    #endif
}
```

**Create a picture file for the project**

Let’s add a picture in the program. Paint program will be used to create a picture. In the paint program, click new, then click the file menu again and click “properties”. The image property dialog will show up as follows.

**Figure 13. Image property dialog window**
Change the width and height of the image as desired. In this program, 100 width and 100 height will be used. Draw a smiley face and save it as png file with named “SmileSprite”.

![Image of Paint and Windows Explorer](image)

**Figure 14. Save a picture as a png file format**

### Codes for adding picture in a XNA program

After saving the picture, back in the VC# Express and include the picture file in the project. In Solution Explorer, right click on the “xna_prog_1 Content (Content)” and click “Add” then click “Existing Item”. Find the created “SmileSprite.png” file and click on “Add” button to include the file in the project.

![Image of Solution Explorer](image)

**Figure 15. Adding a picture file in the project**
The picture has been included in the project, but not used in the program yet. To use the picture in the program:

- Create a texture function for the picture
- Load the picture in the program
- Draw the loaded picture in the program window

Type “Texture2D texSmileSprite {get; set;}” without double quotes in between “SpriteBatch spriteBatch;” and “public xna_prog_1()” in xna_prog_1.cs file as shown below.

```csharp
namespace xna_prog_1
{
    /// <summary>
    /// This is the main type for your game
    /// </summary>
    public class xna_prog_1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;

        Texture2D texSmileSprite { get; set; }

        public xna_prog_1()
        {
            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";
        }
    }
}
```

**Figure 16. Create texture function for a picture**

The created function is going to be used to loading the picture file. To load the picture, type `texSmileSprite = this.Content.Load<Texture2D>("SmileSprite");` without single quotes after the “// TODO: use this.Content to load your game content here” in xna_prog_1.cs file.

```csharp
/// <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
    texSmileSprite = this.Content.Load<Texture2D>("SmileSprite");
}
```

**Figure 17. Loading a picture to a program**
The following code is for drawing the loaded picture in the program window.  

```csharp
spriteBatch.Begin();
spriteBatch.Draw(texSmileSprite, new Vector2(10f, 10f), Color.White);
spriteBatch.End();
```

Put the code after “// TODO: Add your drawing code here” in the “protected override void Draw(GameTime gameTime)” function in xna_prog_1.cs file as shown below.

```csharp
/// <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
    spriteBatch.Begin();
spriteBatch.Draw(texSmileSprite, new Vector2(10f, 10f), Color.White);
spriteBatch.End();

    base.Draw(gameTime);
}
```

**Figure 18. Code for drawing a picture in a program window**

Now it is time for put preferred window size in the program. Type “graphics.PreferredBackBufferWidth = 400; graphics.PreferredBackBufferHeight = 300;” in between “graphics = new GraphicsDeviceManager(this);” and ‘Content.RootDirectory = "Content";’ in “public xna_prog_1()” function in xna_prog_1.cs file as shown below.

```csharp
public xna_prog_1()
{
    graphics = new GraphicsDeviceManager(this);
    graphics.PreferredBackBufferWidth = 400;
    graphics.PreferredBackBufferHeight = 300;
    Content.RootDirectory = "Content";
}
```

**Figure 19. Code for setting preferred window size**
**Put a text in a XNA program**

Let’s add a text in the program. To add a text in a XNA program, the following need to be done.

- Create a Sprite Font
- Edit the font property as needed
- Create a function for the text font
- Load the font in the program
- Draw the font in the preferred location in the program window

To create a Sprite Font, right click on “xna_prog_1Content(Content)” in Solution Explorer, click “Add” and then click “New Item” button.

![Image](image.png)

*Figure 20. Adding new content item in a project*

In the “Add New Item” dialog window, select “Sprite Font”, change name to “MySpriteFont.spritefont”, then click “Add” button.
Change "<FontName>Segoe UI Mono</FontName>" to "<FontName>Comic Sans MS</FontName>" and Change "<Size>14</Size>" to "<Size>24</Size>" in the "MySpriteFont.spritefont" file.

```xml
  <Asset Type="Graphics:FontDescription">
    <!-- Modify this string to change the font that will be imported. -->
    <FontName>Comic Sans MS</FontName>
    <!-- Size is a float value, measured in points. Modify this value to change the size of the font. -->
    <Size>24</Size>
  </Asset>
</XnaContent>
```

Back to the xna_prog_1.cs file, type "SpriteFont sfMySpriteFont { get; set; }") without double quotes below the “Texture2D texSmileSprite { get; set; }” line in xna_prog_1 class as shown below.
The Sprite Font style should be loaded before used in the program. To load the style, type
'sfMySpriteFont = this.Content.Load<SpriteFont>("MySpriteFont");' without single quote in the
LoadContent() function in xna_prog_1.cs file.

Next step is type a code for drawing a text with the loaded Sprite Font style. Put
'spriteBatch.DrawString(sfMySpriteFont, "Welcome back \n Digispired ii students!", new Vector2(10f, 160f), Color.Yellow);' without single quotes in Draw() function in xna_prog_1.cs file.
**Change Game profile**

If you get the following error when you compiled and run the program, you need to change game profile of the project.

![Figure 26. Graphics card support error](image1.png)

To change game profile of the project, right click on the xna_prog_1 in Solution Explorer, then click “Properties” button.

![Figure 27. Project property menu](image2.png)
Change Game profile to "Use Reach to access a limited API set supported by Windows Phone, Xbox 360, and Windows."

Now it is ready to run. Click on “Start Debugging (F5)” button. The program will be compiled and executed.
C# for Digispired ii Academy, July 2011.