



Content

Foreword.....	3
Preparing your development environment	4
Requirements.....	4
Contents	4
Installing Indiefreaks Game Profiler for Xna Games.....	4
Setting up your game project	5
Step 1: Add the assemblies for the right Xna target platform	5
Step 2: Add the Profiler() attribute	6
Step 3: Add the ProfilerGameComponent.....	7
Windows Phone 7 specific code.....	7
Step 4: Add the igp.exe call to your project post build event.....	8
Step 5: Build & Run your game	8
Using Indiefreaks Game Profiler for Xna games	9
Enable/Disable Controls (Windows & Xbox 360)	9
Enable/Disable Controls (Windows Phone 7)	9
Moving the cursor up & down	9
Expanding & Contracting the Stack	9
License Agreement	10

Foreword

The Indiefreaks Game Profiler for Xna games (or IGP to summarize) is composed of two elements: the compile time tool and the runtime assemblies.

When you want to profile your game with IGP, you'll use the compile time tool to inject code to your assembly so that the runtime components can gather and render realtime data.

I hope the Indiefreaks Game Profiler will help the Xna community finding where their game spends most of their processor time and which methods are generating potential memory leaks.

This document quickly goes through the process of setting up your game project to make use of the provided component.

If you find any bug, annoyance or potential improvement, feel free to report them back to me using the dedicated section on the Indiefreaks.com forums:

<http://indiefreaks.com/indiefreaks-game-framework/forums/?mingleforumaction=vforum&g=5.0>

Thanks

Philippe Da Silva

<http://indiefreaks.com>

Preparing your development environment

Requirements

Indiefreaks Game Profiler for Xna games works on any XNA game project (Windows, Xbox 360 or Windows Phone 7)

It only requires any Visual Studio 2010 SP1 edition and XNA 4.0 framework

Contents

The current distribution is composed of the following elements:

- The present document
- A Setup application

Installing Indiefreaks Game Profiler for Xna Games

Double click the Setup.exe file and follow the Setup Wizard steps.

Setting up your game project

Step 1: Add the assemblies for the right Xna target platform

The Indiebreaks Game Profiler for Xna games tool comes with a set of assemblies for each Xna target platform your project may be setup for.

You'll find these assemblies in the installation directory under the Assemblies folder. If you didn't change the default installation path, you'll find them under the following directory path:

- On 32 bits operating systems:

"C:\Program Files\Indiebreaks.com\Indiebreaks Game Profiler\Assemblies\"

- On 64 bits operating systems:

"C:\Program Files (x86)\Indiebreaks.com\Indiebreaks Game Profiler\Assemblies\"

Simply select the appropriate folder under the above path for your Xna game project and add references to the two provided assemblies:

- Indiebreaks.AOP.Profiler.dll
- Indiebreaks.Xna.Profiler.dll

Step 2: Add the Profiler() attribute

The Indiebreaks Game Profiler for Xna Games tool will only process an assembly if it contains a ProfilerAttribute instance.

It is really simple to set up. Simply add the following code in your AssemblyInfo.cs file:

```
[assembly: Profiler(GameClass = "MyGameNamespace.MyGame")]
```

As you'll notice, the Profiler attribute lets you define the main Microsoft.Xna.Framework.Game inherited class of your Xna project. This is only required on your main Xna assembly so that it updates and renders the profiled data.

If you have some custom referenced assemblies that you want to profile too, you may just set the Profiler attribute with:

```
[assembly: Profiler()]
```

The Profiler attribute also accepts a IgnoredNamespaces property that lets you place a set of comma separated namespaces for which you don't want classes to be profiled within your game project.

This can become useful when looking to profile specific parts of your game project.

Here is an example how you can configure it:

```
[assembly: Profiler(IgnoredNamespaces = "MyGame.SomeNamespace, MyGame.AnotherNamespace")]
```

Step 3: Add the ProfilerGameComponent instance

In order to update and render the profiled data, you must add the ProfilerGameComponent game component that simply gather all input events and renders the profiled classes and methods.

It works exactly as other Xna game components and setting it up is as simple as creating an instance on your game constructor class and adding it to the game Components property like so:

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    var profilerGameComponent = new ProfilerGameComponent(this, "ProfilerFont");
    ProfilingManager.Run = true;
    Components.Add(profilerGameComponent);
}
```

The ProfilerGameComponent game component requires you to give him a SpriteFont instance path in its constructor (in the above example it is set to "ProfilerFont"). You'll most likely want it to be a 14 sized SpriteFont to avoid rendering artifacts.

You can control in code the ProfilingManager.Run static property to activate or deactivate the profiling process if you want.

Windows Phone 7 specific code

If you are profiling a Windows Phone 7 Xna game project, you'll need to pass an instance of the current TouchPanel TouchCollection state so that it answers to touch events.

```
profilerGameComponent.UpdateTouches(TouchPanel1.GetState());
```

Since the TouchPanel.GetState() method removes all the touch input events from the current frame state, if you are already gathering the TouchCollection returned by this method elsewhere in your game, you may want to pass it directly to the ProfilerGameComponent.UpdateTouches() method.

```
profilerGameComponent.UpdateTouches(myInputHandlerSystem.CurrentFrameTouchCollection);
```

If you don't do so, you won't be able to navigate through the profiled methods.

Step 4: Add the igp.exe call to your project post build event

This last step is the piece that will transform your assembly injecting IL code around your classes' methods to profile their processing time and the memory allocated.

In order to do so, in your Xna Project properties, click on the "Build Events" tab and add the following line to the Post-build event command line text box:

```
"{path to the IGP installation directory}\igp.exe" "$(TargetPath)"
```

This will make sure that the output assembly for the current project will be processed by the executable in charge of injecting the .Net IL code required to profile your game.

If you selected the default installation directory, it should look like this:

On a 32 bits operating system:

```
"C:\Program Files\Indiefreaks.com\Indiefreaks Game Profiler\igp.exe"
```

On a 64 bits operating system:

```
"C:\Program Files (x86)\Indiefreaks.com\Indiefreaks Game Profiler\igp.exe"
```

Step 5: Build & Run your game

Once all previous steps are setup, you simply build your project and run your game.

Using Indiefreaks Game Profiler for Xna games

Enable/Disable Controls (Windows & Xbox 360)

You can enable or disable the Indiefreaks Game Profiler by hitting a combination of input buttons/keys.

- On Windows, you simply press "Left Ctrl+F12";
- On Windows & Xbox 360, you'll use "Start button + Dpad Up";

Enable/Disable Controls (Windows Phone 7)

In order to enable or disable the Indiefreaks Game Profiler on a Windows Phone 7 Xna game project, you must set the ProfilingManager.Run static method to true or false directly in your code.

It's up to you to create the gesture or touch input events that will do it.

Moving the cursor up & down

- On Windows, you can use the Up and Down arrows of your keyboard;
- On Windows & Xbox 360, you can use the Dpad Up and Dpad Down;
- On WP7, you'll tap on the Top middle zone of the screen to move up and Bottom middle to move down

Expanding & Contracting the Stack

When you'll see a "[+]" in front of a method, that means that you can expand the stack on this method.

- On Windows, you can use the Right arrow key on your keyboard to expand a method and the Left arrow to get back one level;
- On Windows & Xbox 360, you can use the Dpad Right and Dpad Left;
- On WP7, you'll tap on the Right middle zone of the screen to expand and Left middle zone to contract;

License Agreement

Indiefreaks.com End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Indiefreaks.com for the Indiefreaks.com software product(s) identified above which may include associated software components, media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. This license agreement represents the entire agreement concerning the program between you and Indiefreaks.com, (referred to as "licenser"), and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE.

The SOFTWARE PRODUCT is licensed as follows:

(a) Installation and Use.

Indiefreaks.com grants you the right to install and use copies of the SOFTWARE PRODUCT on your computer running a validly licensed copy of the operating system for which the SOFTWARE PRODUCT was designed.

(b) Backup Copies.

You may also make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

(a) Maintenance of Copyright Notices.

You must not remove or alter any copyright notices on any and all copies of the SOFTWARE PRODUCT.

(b) Distribution.

You may not distribute registered copies of the SOFTWARE PRODUCT to third parties.

(c) Prohibition on Reverse Engineering, Decompilation, and Disassembly.

You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

(d) Rental.

You may not rent, lease, or lend the SOFTWARE PRODUCT.

(e) Support Services.

Indiefreaks.com may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA.

(f) Compliance with Applicable Laws.

You must comply with all applicable laws regarding use of the SOFTWARE PRODUCT.

3. TERMINATION

Without prejudice to any other rights, Indiefreaks.com may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT in your possession.

4. COPYRIGHT

All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by Indiefreaks.com or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by Indiefreaks.com.

5. NO WARRANTIES

Indiefreaks.com expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT is provided 'As Is' without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, noninfringement, or fitness of a particular purpose. Indiefreaks.com does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the SOFTWARE PRODUCT. Indiefreaks.com makes no warranties respecting any harm that may be caused by the transmission of a computer virus, worm, time bomb, logic bomb, or other such computer program. Indiefreaks.com further expressly disclaims any warranty or representation to Authorized Users or to any third party.

6. LIMITATION OF LIABILITY

In no event shall Indiefreaks.com be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if Indiefreaks.com has been advised of the possibility of such damages. In no event will Indiefreaks.com be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. Indiefreaks.com shall have no liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors

or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information