

Kristoffer Olsson
Fredrik Håkansson
Projekt 1
2014-05-18
Arkadmaskinen - Systemet
20141-TS206 - Projekt 1

Användardokumentation för MAH Arcade Game Template / User instructions for MAH Arcade Game Template

v.2

Innehållsförteckning / Table of Contents

Användardokumentation för MAH Arcade Game Template / User instructions for MAH Arcade Game Template.....	1
v.2.....	1
Användarinstruktioner för MAH Arcade Game Template.....	3
1.Systemkrav.....	3
2.Installationsinstruktioner för MAH Arcade Game Template (Med hjälp av installer).....	4
3.Installationsinstruktioner för manuell installation av MAH Arcade Game Template till Visual Studio 2013.....	8
4.Tekniska likheter mellan Microsoft.Xna.Framework.Game och MAH ArcadeSystem.BaseGame.....	9
4.1. Fields och Properties.....	9
4.2. Metoder.....	9
5.Tekniska skillnader mellan Microsoft.Xna.Framework.Game och MAH ArcadeSystem.BaseGame.....	10
5.1 Fields och Properties.....	10
5.2.Metoder:.....	11
5.3.Drawing och Rendering.....	12
5.4.Input handling.....	12
User instructions for MAH Arcade Game Template - English version.....	14
1.Prerequisites.....	14
2.Install instructions of MAH Arcade Game Template (Using the Installer).....	15
3.Install instructions of MAH Arcade Game Template for Visual Studio 2013.....	19
4.Similarities between Microsoft.Xna.Framework.Game and MAH ArcadeSystem.BaseGame. .	20
4.1.Fields and Properties.....	20
4.2.Methods.....	20
5.Differences between Microsoft.Xna.Framework.Game and MAH ArcadeSystem.BaseGame. .	21
5.1.Fields and Properties:.....	21
5.2.Methods.....	22
5.3.Drawing and Rendering.....	22
5.4.Input handling.....	23

Användarinstruktioner för MAH Arcade Game Template

1.Systemkrav

För att templatén ska fungera behöver du ha XNA 4.0 eller senare version, samt .NET Framework 4.0 eller senare version installerad. Vi rekommenderar också att du installerar den senaste versionen av DirectX 11.

Du måste också ha Visual Studios 2010 eller senare installerat på ditt system.

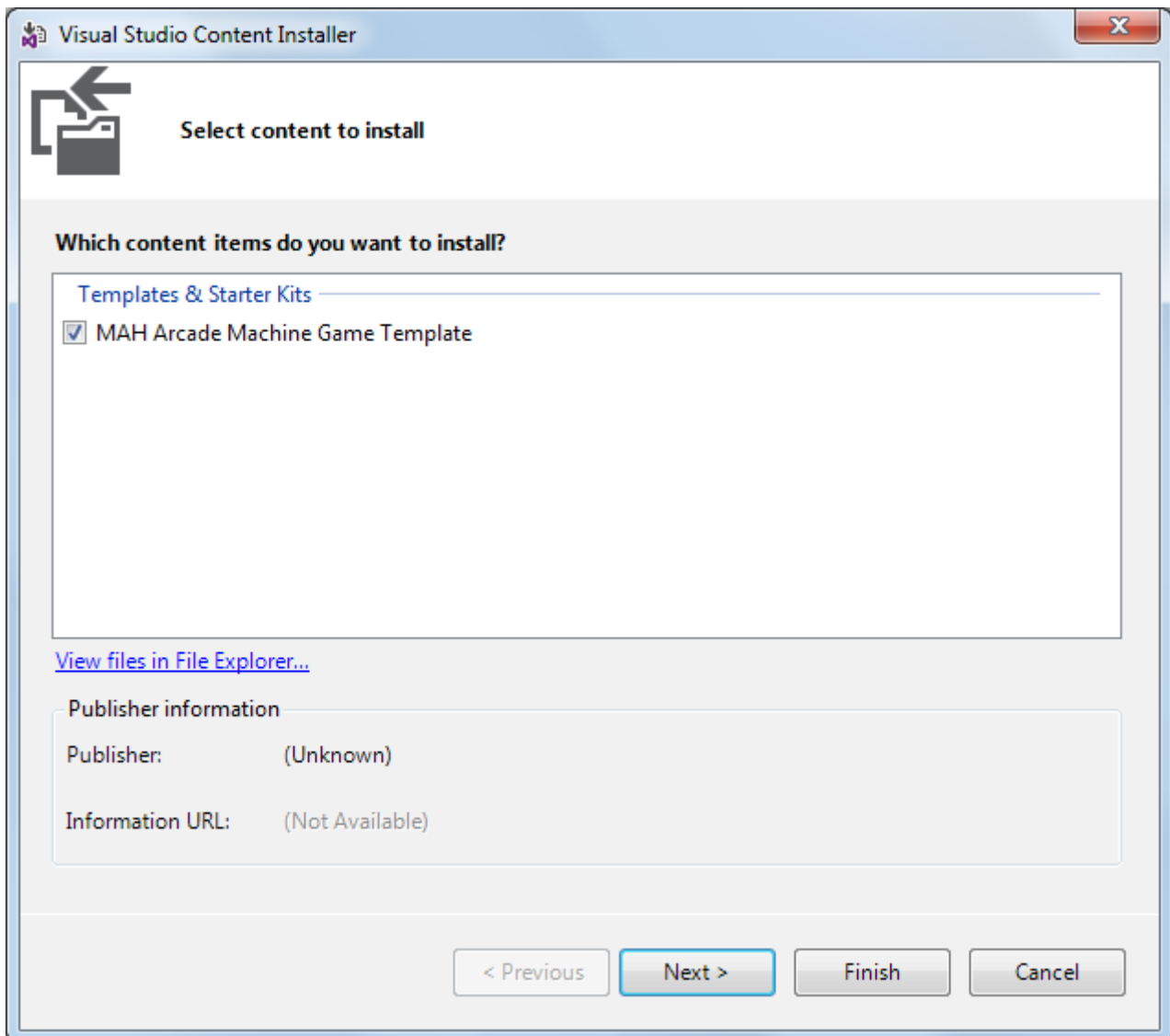
För mer information om XNA besök: <http://www.microsoft.com/en-us/download/details.aspx?id=23714>

För mer information om .NET Framework besök: <http://www.microsoft.com/en-us/download/details.aspx?id=17851>

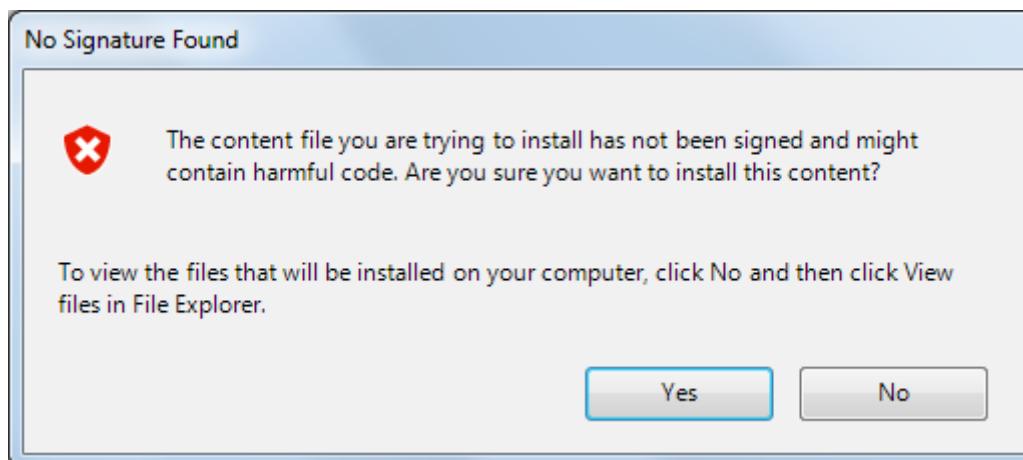
2. Installationsinstruktioner för MAH Arcade Game Template (Med hjälp av installer)

Dubbelklicka på MAH Arcade Game Template Installer.VSI för att starta installationen. Ett installationsfönster för Visual Studio Content kommer nu visas.

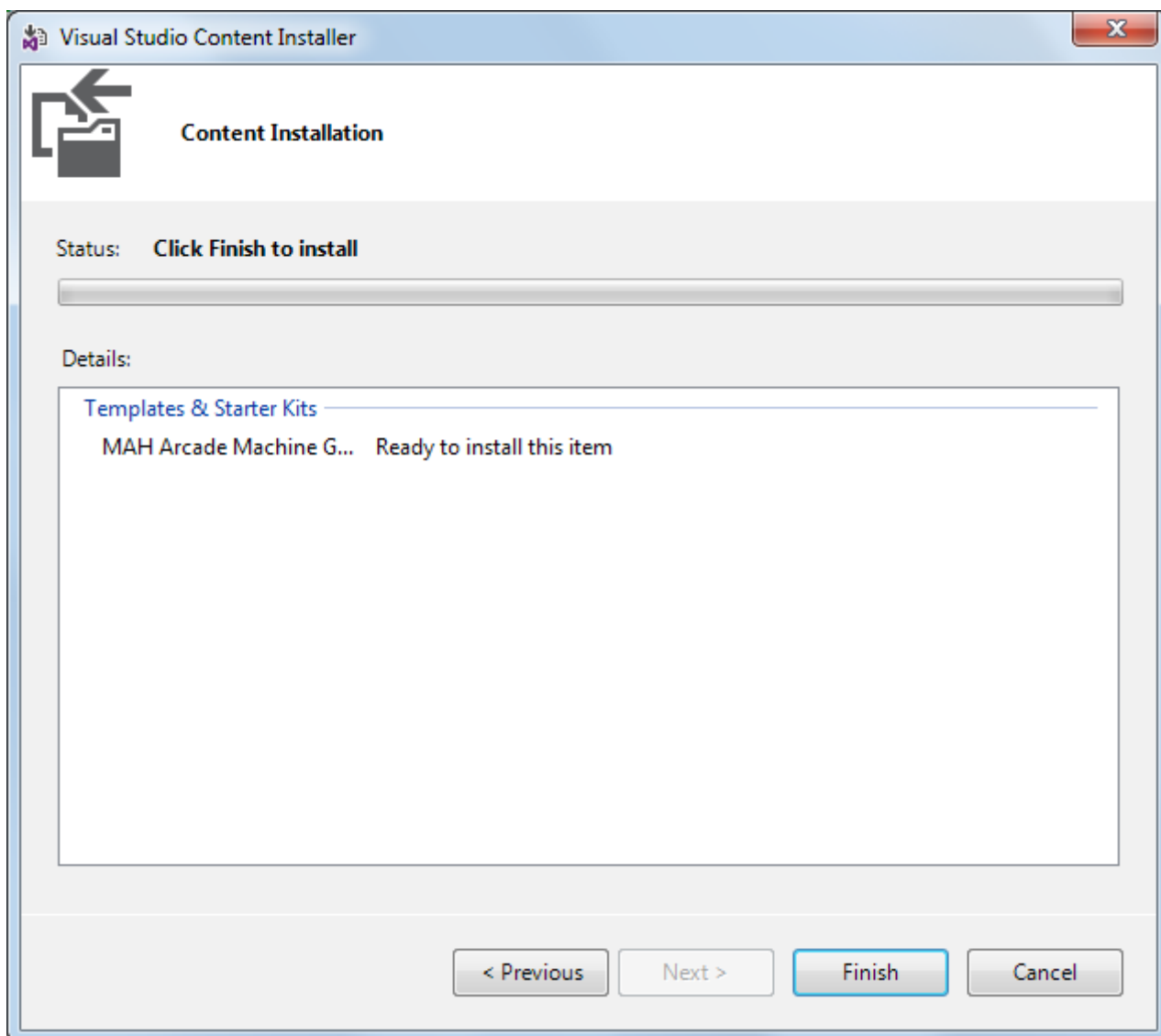
Säkerställ att MAH Arcade Machine Game Template är icheckad i menyn *Templates & Starter Kits*. Efter att du säkerställt detta, klicka på *Next*.



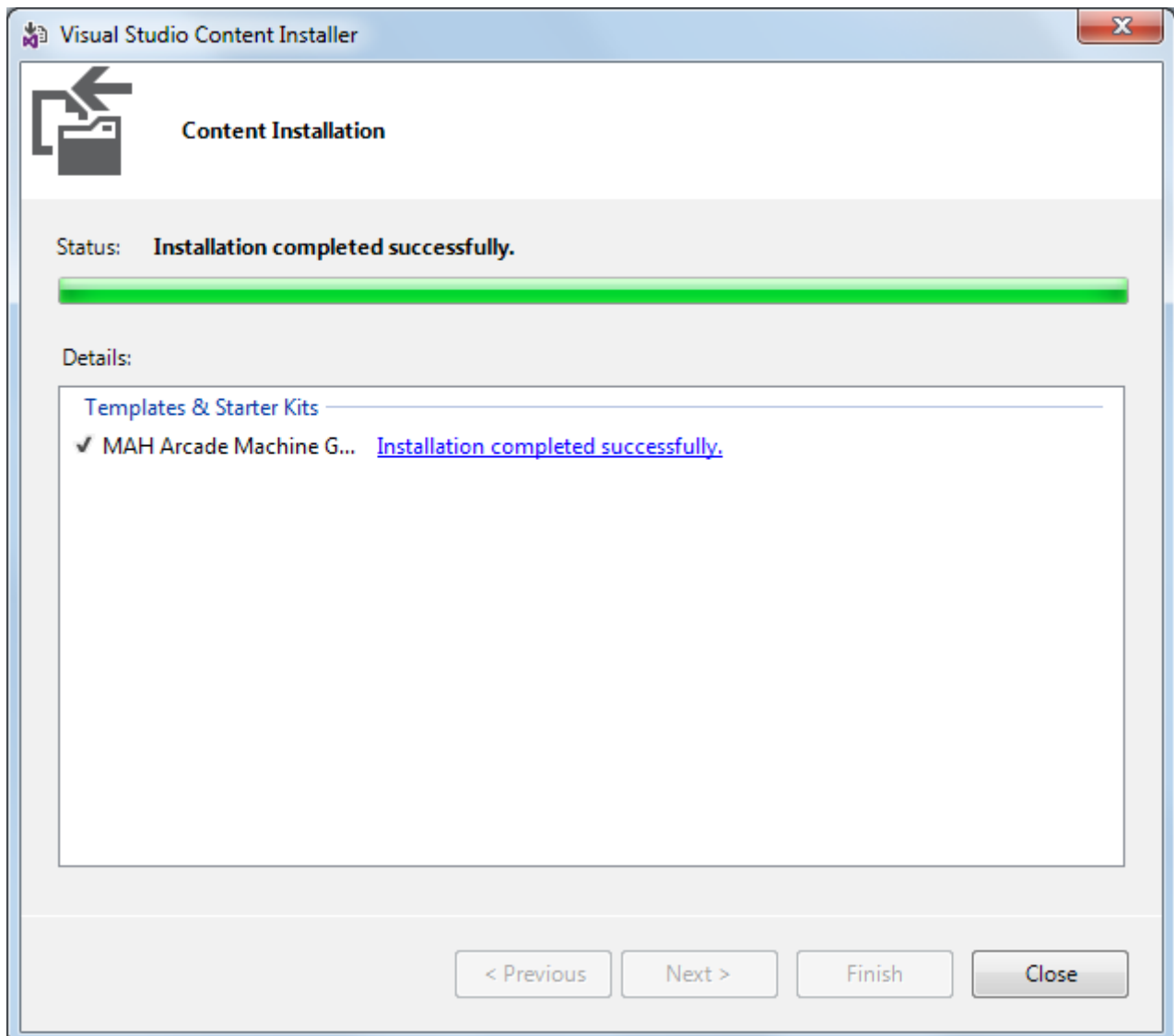
Ett pop-up fönster med titeln *No Signature Found* kommer nu visas. Fönstret informerar om att templatens innehåll kan innehålla skadlig kod. Det här är normalt. Koden är harmlös för ditt system, men eftersom MAH inte är publicatören av templatens har inte templatens blivit signerad.



Installationsfönstret för Visual Studio Content kommer nu uppdateras och fråga dig om du vill slutföra installationen. Klicka på *"Click Finish to install"*.



När installationen är slutförd kommer installationsfönstret för Visual Studio Content uppdateras igen och visa *"Installation completed successfully"*. Du kan nu klicka på *Close* för att slutföra det sista steget i installationen.

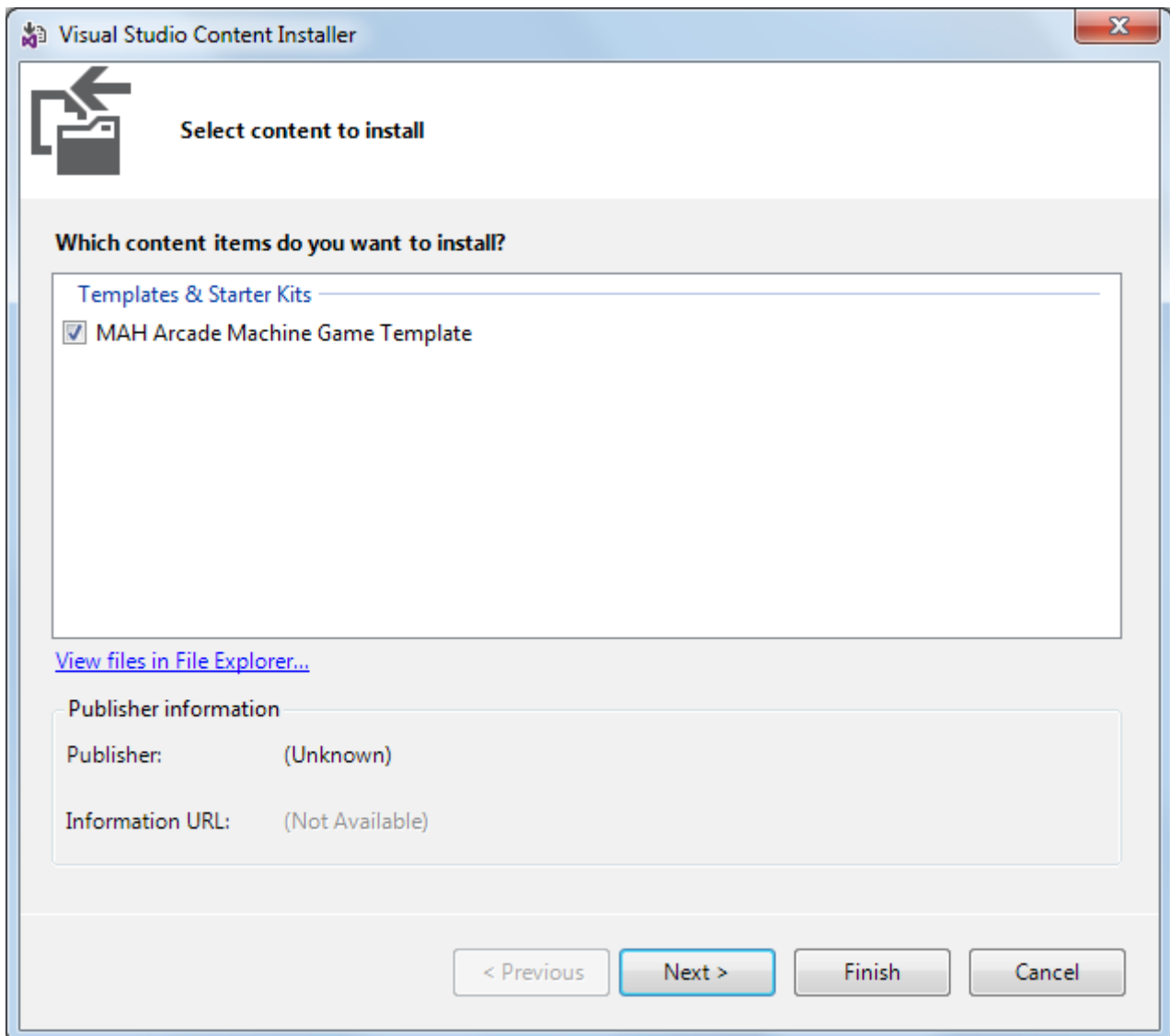


3. Installationsinstruktioner för manuell installation av MAH Arcade Game Template till Visual Studio 2013

Installer för templatén stödjer endast Visual Studios 2012. Därför måste installationen av templatén göras manuellt.

Dubbelklicka på MAH Arcade Game Template Installer.VSI för att starta installationen. Ett installationsfönster för Visual Studio Content kommer nu visas.

Klicka på texten *View files in File Explorer...*



Kopiera filen MAH Arcade Game Template.zip och klistra in filen i foldern : %HOMEPATH%\Documents\Visual Studio 2013\Templates\Project Templates\Visual C#\

Den manuella installationen av templatén är nu slutförd.

4. Tekniska likheter mellan *Microsoft.Xna.Framework.Game* och *MAH ArcadeSystem.BaseGame*

4.1. Fields och Properties

ContentManager Content { get; }	ContentManager är bestämd och får inte ändras. Skriv Content is set after the Constructor(); but before the Initialize();
---------------------------------	---

Utöver detta fungerar allt som vanligt i ContentManager. <http://msdn.microsoft.com/en-us/library/Microsoft.Xna.Framework.Content.ContentManager.aspx>.

4.2. Metoder

Initialize();	Du måste inte kalla på metoden base.Initialize() Initialize() kallas efter konstruktorkallet.
Update(GameTime gameTime);	Fungerar som metoden Update(GameTime gameTime); med undantaget att du inte behöver kalla på base.UpdateWorks. Uppdateringsfrekvensen är förutbestämd och kan inte ändras.
Draw(GameTime gameTime);	Fungerar som metoden Draw(GameTime gameTime); med undantaget att du inte behöver kalla på base.Draw(gameTime); samt att du utför alla Draw anrop genom den statiska klassen Renderer.

5. Tekniska skillnader mellan Microsoft.Xna.Framework.Game och MAH ArcadeSystem.BaseGame

5.1 Fields och Properties

GraphicsDeviceManager graphics;	N/A
SpriteBatch spriteBatch;	N/A
GameComponentCollection Components { get; }	N/A
GraphicsDevice GraphicsDevice { get; }	N/A
TimeSpan InactiveSleepTime { get; set; }	N/A
bool IsActive { get; }	N/A
bool IsFixedTimeStep { get; set; }	N/A
bool IsMouseVisible { get; set; }	N/A
LaunchParameters LaunchParameters { get; }	N/A
GameServiceContainer Services { get; }	N/A
TimeSpan TargetElapsedTime { get; set; }	N/A
GameWindow Window { get; }	N/A
event EventHandler<EventArgs> Activated;	N/A
event EventHandler<EventArgs> Deactivated;	N/A
event EventHandler<EventArgs> Disposed;	N/A
event EventHandler<EventArgs> Exiting;	N/A

5.2. Metoder:

BeginDraw();	N/A
BeginRun();	N/A
Dispose();	N/A
Dispose(bool disposing);	N/A
EndDraw();	N/A
EndRun();	N/A
Exit();	N/A
LoadContent();	N/A. Om du behöver metoden rekommenderar vi att du skriver en <i>private void LoadContent()</i> ; och kallar metoden i slutet av konstruktorn.
OnActivated(object sender, EventArgs args);	N/A
OnDeactivated(object sender, EventArgs args);	N/A
OnExiting(object sender, EventArgs args);	N/A
ResetElapsedTime(); Doesn't exist.	N/A
Run();	N/A
RunOneFrame();	N/A
ShowMissingRequirementMessage(Exception exception);	N/A
SuppressDraw();	N/A
Tick();	N/A
UnloadContent();	N/A

5.3.Drawing och Rendering

SpriteBatch är inte längre tillgänglig. Istället används den statiska klassen `Renderer` med metodtilläggen:

<code>Clear(Color color);</code>
<code>SetRenderTarget(RenderTarget2D renderTarget);</code>

Så länge du följer renderingsstrukturen för XNA `old draw call order from XNA` och ersätter alla instanser av `SpriteBatch` och `GraphicsDevice` med `Renderer` borde det inte orsaka några problem.

5.4.Input handling

Eftersom användaren ger input till systemet via en digital joystick och knappar har vi tagit fram en input handler. Du kan fortfarande använda den standardiserade XNA keyboard state metoden, men `InputHandler` ger ett enkelt sätt att kontrollera olika input-statements samt att den har alla knappar mappade till systemet.

För att få information om tillstånden för knapp och joystick har du tre metoder inom den statiska klassen `InputHandler` till ditt förfogande:

<code>public static InputState GetButtonState(PlayerInput button);</code>	
<code>public static bool IsKeyDown(PlayerInput button, bool checkLastInputState);</code>	Fungerar på ett liknande sätt som <code>KeyboardState.IsKeyDown()</code> ;
<code>public static bool IsKeyUp(PlayerInput button, bool checkLastInputState);</code>	Fungerar på ett liknande sätt som <code>KeyboardState.IsKeyUp()</code> ;

Du har också två enumtyper till ditt förfogande:

<code>PlayerInput</code>	Fungerar på samma sätt som <code>Keys</code> enum
<code>InputState</code>	Fungerar på ett liknande sätt som <code>KeyState</code> .

Knappar och joystick kan ha fyra möjliga tillstånd

Pressed	<i>Pressed</i> innebär att knappen / joysticken inte var nertryckt i den föregående framen, men är nertryckt nu.
Released	<i>Released</i> innebär att knappen / joysticken var nertryckt i föregående framen, men är släppt nu.
Up	<i>Up</i> innebär att knappen / joysticken var nertryckt i både föregående frame och nuvarande frame.
Down	<i>Down</i> innebär att knappen / joysticken var släppt i både föregående frame och nuvarande frame.

User instructions for MAH Arcade Game Template - English version

1.Prerequisites

For the template to work you need to have XNA 4.0 or later and .NET Framework 4.0 or later installed.

You also need Visual Studios 2010 or later installed.

It is also recommended that you install the latest version of DirectX 11.

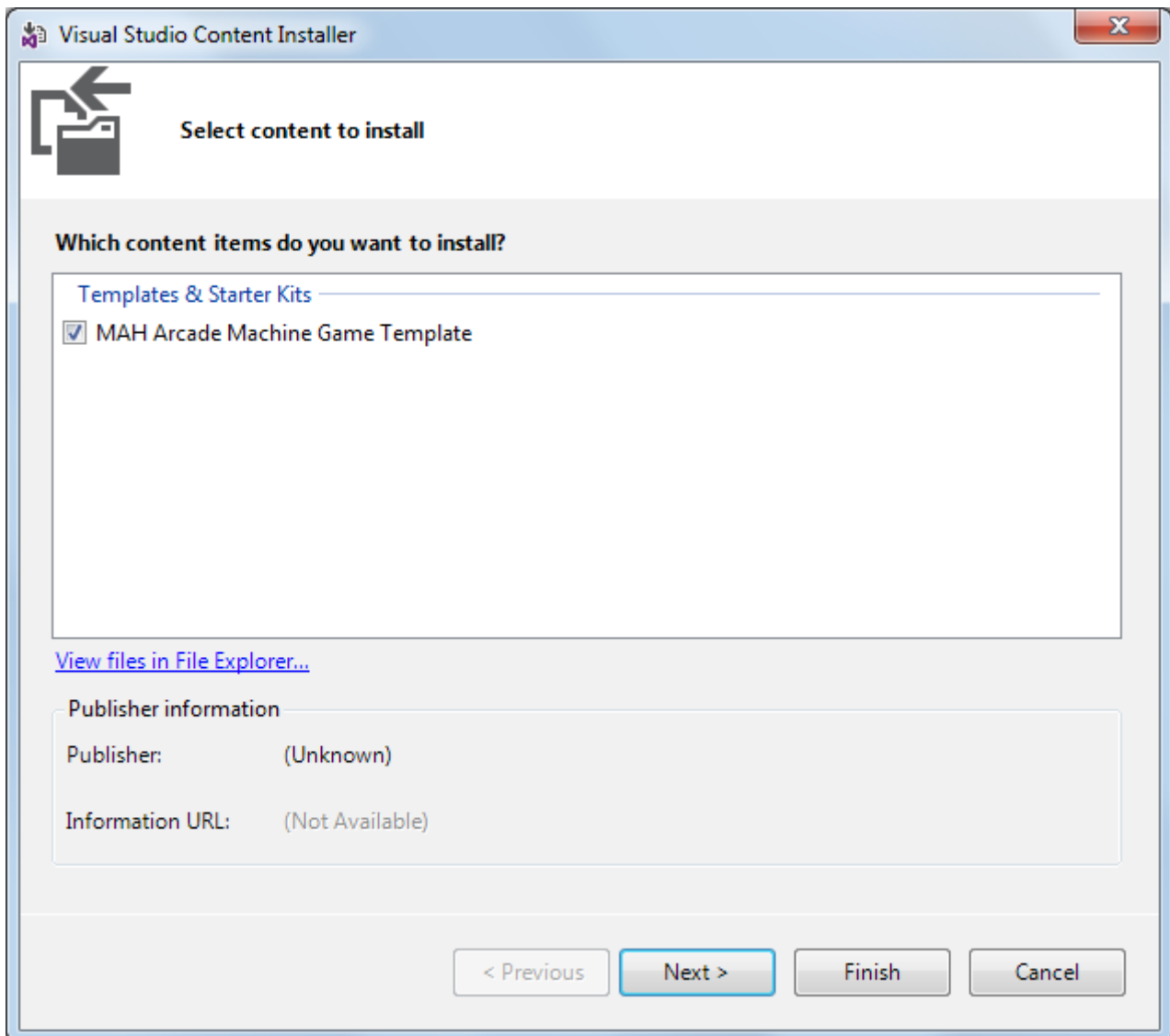
For more info on XNA: <http://www.microsoft.com/en-us/download/details.aspx?id=23714>

.NET Framework: <http://www.microsoft.com/en-us/download/details.aspx?id=17851>

2.Install instructions of MAH Arcade Game Template (Using the Installer)

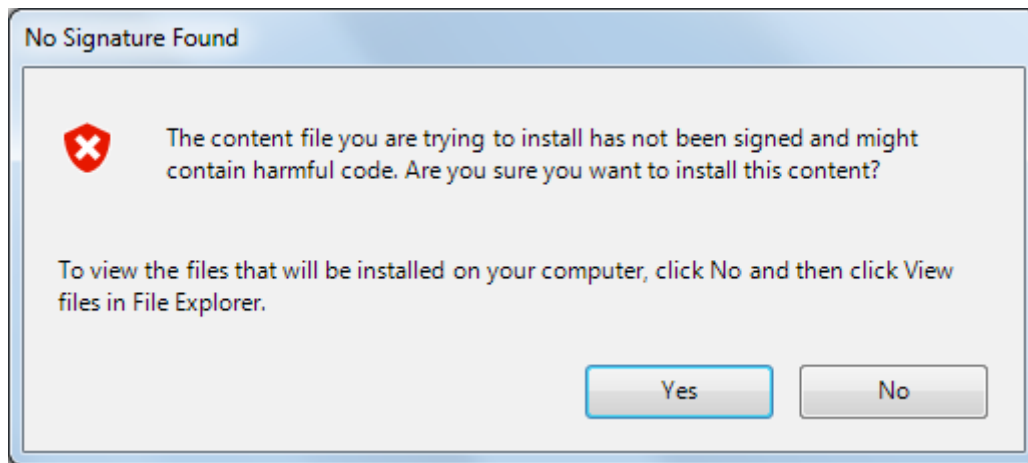
Double click the MAH Arcade Game Template Installer.VSI to start the installer. A Visual Studio Content Installer should now appear.

Make sure that the MAH Arcade Machine Game Template is checked, under Templates & Starter Kits. Then click next.

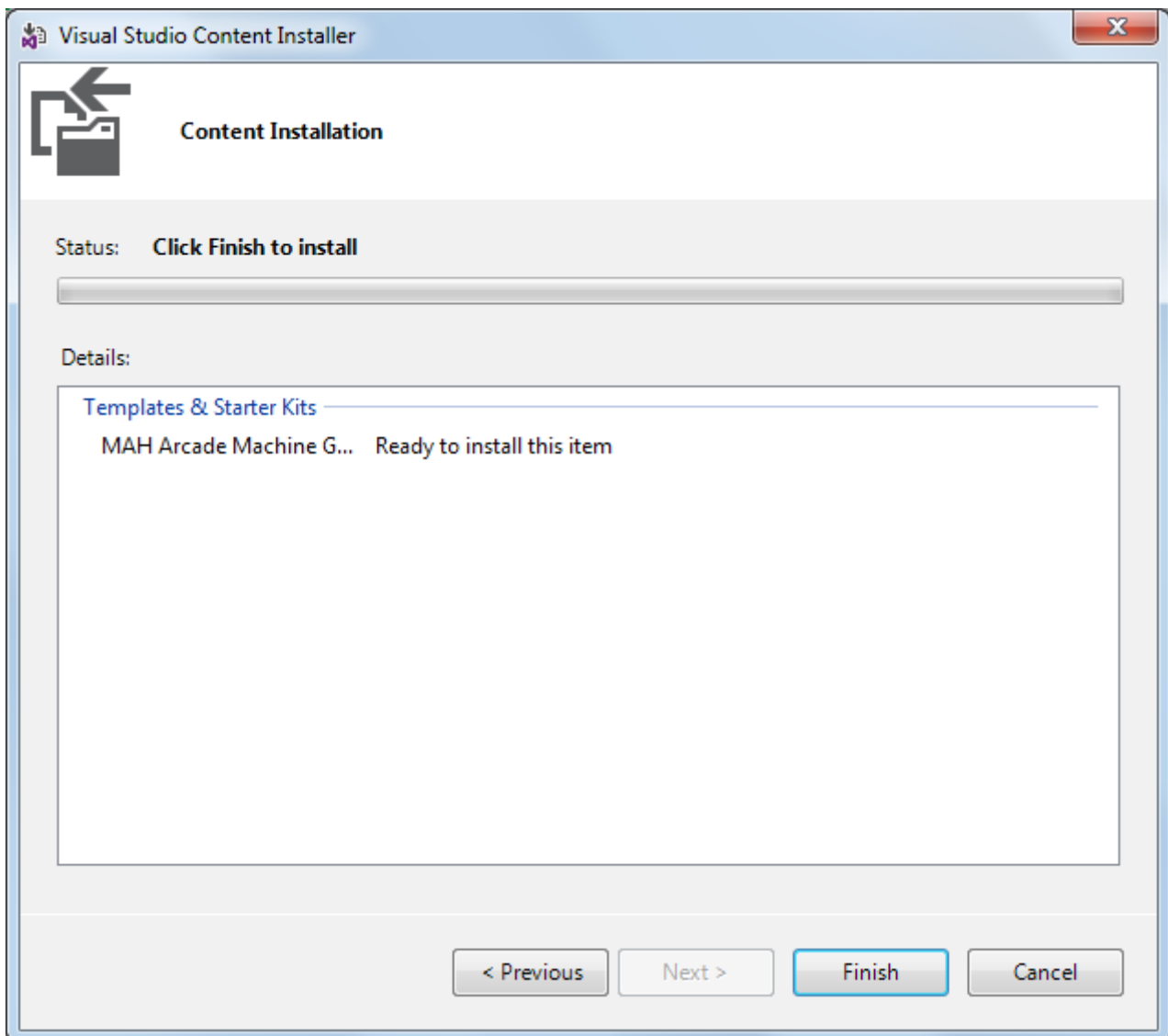


A pop up window with the title *No Signature Found* will appear, informing you that this content might contain harmful code.

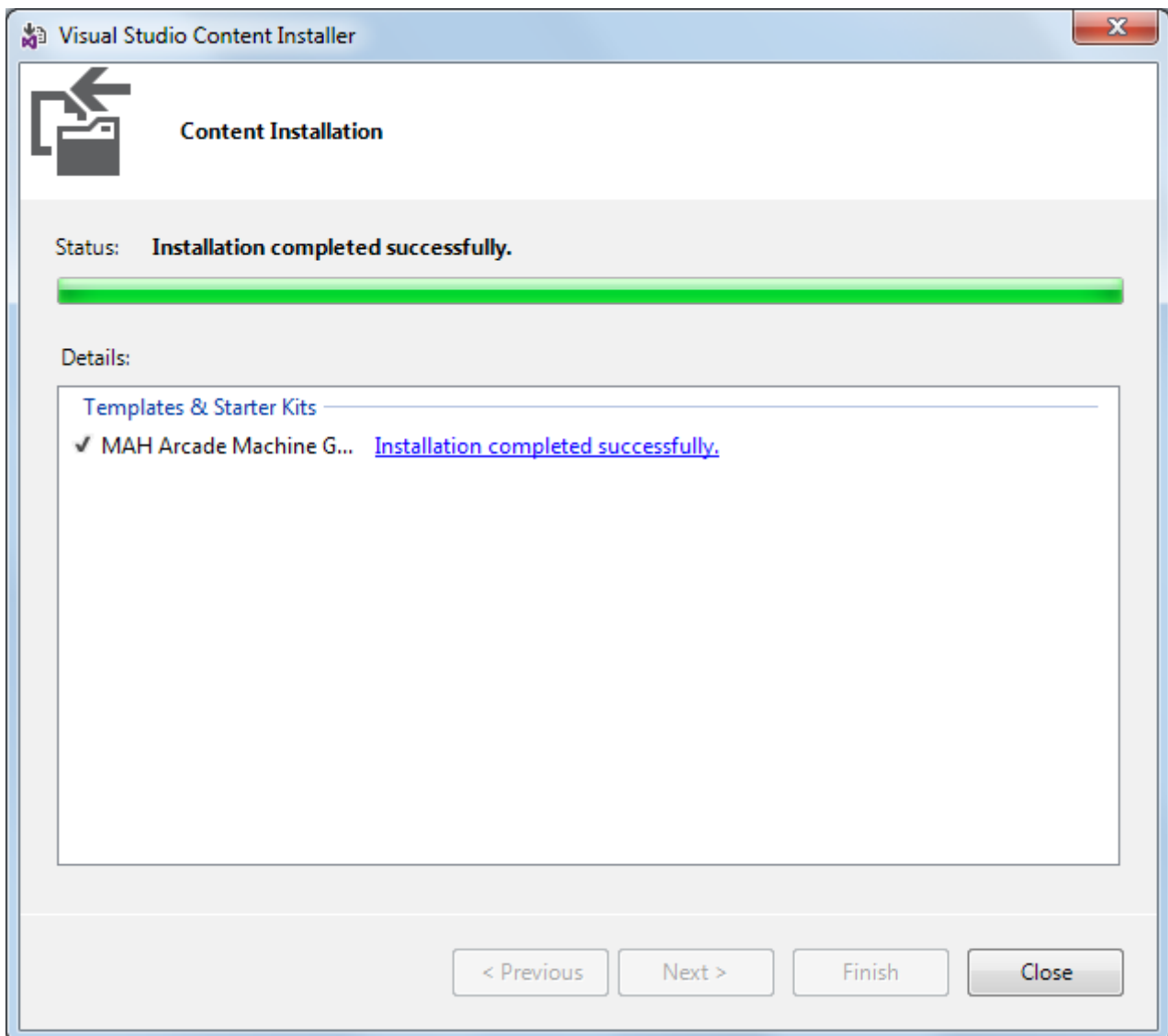
This is because MAH is not a publisher and therefore the content haven't been Signed. The code is harmless though. Press *yes* to continue.



The Visual Studio Content Installer window should update and give you a Status bar with the text: Click Finish to install. now click Finish to install.



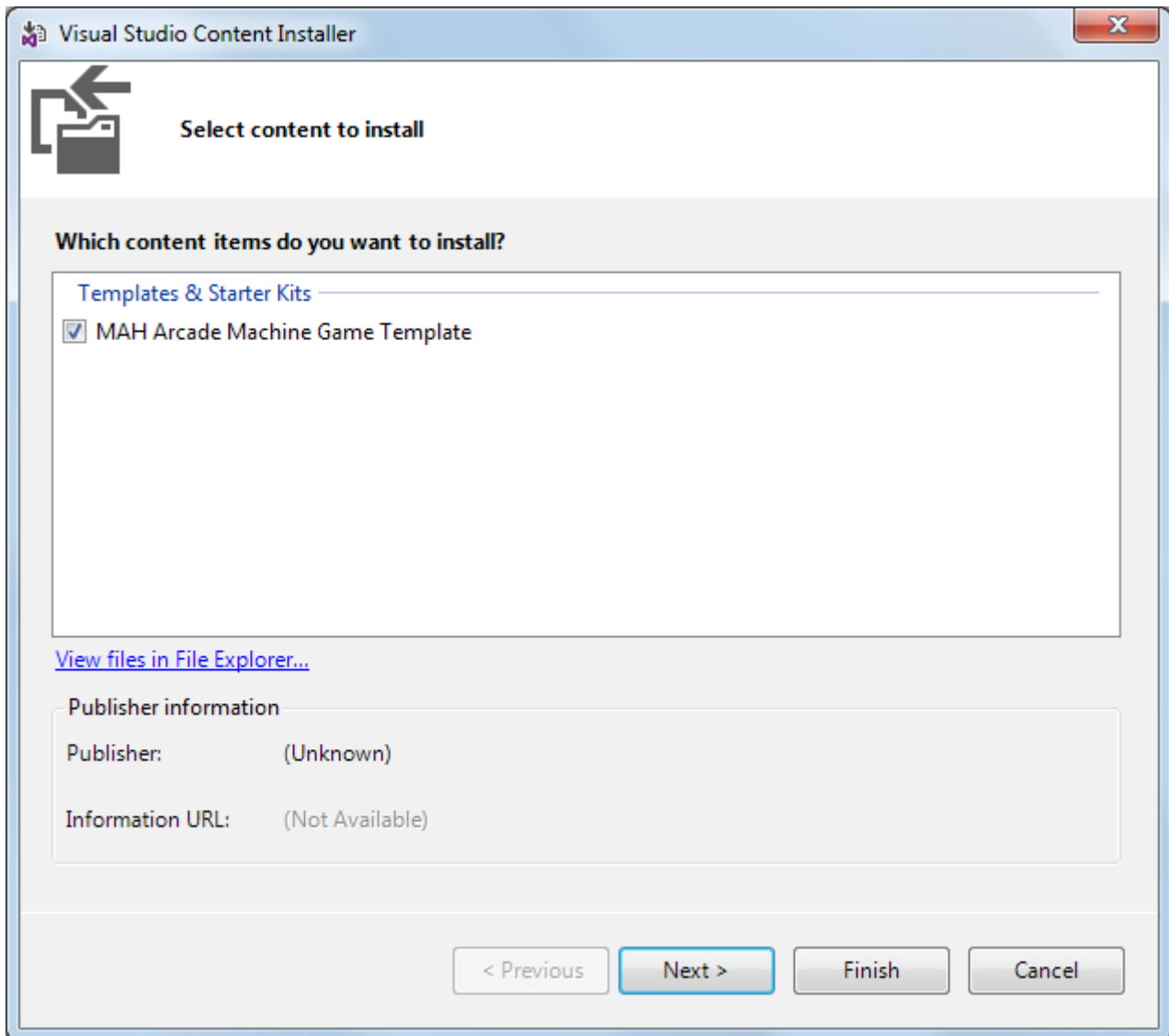
When the install is done Status bar text will change to: Installation completed successfully. You can now click close.



3. Install instructions of MAH Arcade Game Template for Visual Studio 2013

Double click the MAH Arcade Game Template Installer.VSI to start the installer. A Visual Studio Content Installer should now appear.

The installer can't install the template for Visual Studio 2013, so it have to be done manually. Click on the text View files in File Explorer...



Copy the file MAH Arcade Game Template.zip and paste it in the folder %HOMEPATH%\Documents\Visual Studio 2013\Templates\ProjectTemplates\Visual C#\

4. Similarities between *Microsoft.Xna.Framework.Game* and *MAH ArcadeSystem.BaseGame*

4.1. Fields and Properties

ContentManager Content { get; }	The ContentManager is preset and can't be changed. Content is set after the Constructor(); but before the Initialize();
---------------------------------	---

Everything else in the ContentManager works as normal <http://msdn.microsoft.com/en-us/library/Microsoft.Xna.Framework.Content.ContentManager.aspx>.

4.2. Methods

Initialize();	You don't need to call base.Initialize(); Initialize(); is called right after the Constructor();
Update(GameTime gameTime);	Works like the normal Update(GameTime gameTime); except you don't need to call base.Update(gameTime); Update rate is preset and can't be changed.
Draw(GameTime gameTime);	Works like the normal Draw(GameTime gameTime); except you don't need to call base.Draw(gameTime); and you make all draw calls to the static Renderer class.

5.Differences between Microsoft.Xna.Framework.Game and MAH ArcadeSystem.BaseGame

5.1.Fields and Properties:

GraphicsDeviceManager graphics;	N/A
SpriteBatch spriteBatch;	N/A
GameComponentCollection Components { get; }	N/A
GraphicsDevice GraphicsDevice { get; }	N/A
TimeSpan InactiveSleepTime { get; set; }	N/A
bool IsActive { get; }	N/A
bool IsFixedTimeStep { get; set; }	N/A
bool IsMouseVisible { get; set; }	N/A
LaunchParameters LaunchParameters { get; }	N/A
GameServiceContainer Services { get; }	N/A
TimeSpan TargetElapsedTime { get; set; }	N/A
GameWindow Window { get; }	N/A
event EventHandler<EventArgs> Activated;	N/A
event EventHandler<EventArgs> Deactivated;	N/A
event EventHandler<EventArgs> Disposed;	N/A
event EventHandler<EventArgs> Exiting;	N/A

5.2.Methods

BeginDraw();	N/A
BeginRun();	N/A
Dispose();	N/A
Dispose(bool disposing);	N/A
EndDraw();	N/A
EndRun();	N/A
Exit();	N/A
LoadContent();	N/A. If needed, recommended that you make a private void LoadContent(); and the call it at the end in the Constructor();
OnActivated(object sender, EventArgs args);	N/A
OnDeactivated(object sender, EventArgs args);	N/A
OnExiting(object sender, EventArgs args);	N/A
ResetElapsedTime(); Doesn't exist.	N/A
Run();	N/A
RunOneFrame();	N/A
ShowMissingRequirementMessage(Exception exception);	N/A
SuppressDraw();	N/A
Tick();	N/A
UnloadContent();	N/A

5.3.Drawing and Rendering

SpriteBatch is gone. You no longer call the draw methods through a SpriteBatch. Instead you use a static class called Renderer, with the Clear(Color color); and SetRenderTarget(RenderTarget2D renderTarget); method added to it.

As long as you follow the old draw call order from XNA and just replace all SpriteBatch and GraphicsDevice with Renderer you should be fine.

5.4.Input handling

Since the user is giving input through (digital) joysticks and buttons, a new input handler is provided. You can still use the standard XNA keyboard state method. But the new InputHandler provide an easy way to check the different input statement and have all the button pre mapped.

To get info about the state of a button or joystick direction, you have three methods within the static class InputHandler at your disposal:

public static InputState GetButtonState(PlayerInput button);	
public static bool IsKeyDown(PlayerInput button, bool checkLastInputState);	Works similar to KeyboardState.IsKeyDown();
public static bool IsKeyUp(PlayerInput button, bool checkLastInputState);	Works similar to KeyboardState.IsKeyUp();

You also have two enum types:

PlayerInput	Works the same way as the Keys enum
InputState	Works in a similar way to KeyState.

A button / joystick direction can be in four different states:

Pressed	Pressed means that the button / joystick direction was released in the previous frame but is now pressed.
Released	Released means that the button / joystick direction was pressed in the previous frame but is now released.now released.
Up	Up means that the button / joystick direction is pressed in both this frame and the previous.
Down	Down means that the button / joystick direction is released in both this frame and the previous.