
Cities: Skylines Modding Documentation

Release 0.0.1

DarkArcana

August 27, 2017

1	Documentation	3
1.1	General	3
1.2	Modding	4
1.3	Assets	15
1.4	Maps	16
1.5	Resources	16
1.6	About The Docs	113
1.7	Links	117
2	Indices and tables	119

Welcome to the Cities: Skylines modding documentation! This is an unofficial wiki created by fans for the fans.

Warning: The documentation here is still under construction. As such, not everything is complete or accurate. Learn how to contribute [here](#).

Documentation

General

Folder Structure

The main directory you will need for modding is located in local app appdata. This is where you create all your modifications to the game.

- **On Windows** %LOCALAPPDATA%\Colossal Order\Cities_Skylines
- **On Mac** /Users/<username>/Library/Application Support/Colossal Order/Cities_Skylines
- **On Linux** /home/<username>/.local/share/Colossal Order/Cities_Skylines

The other directory you might need is located in your steamapps. This directory contains all the [DLLs](#) for modding, [output log](#) and more.

- **On Windows** [SteamLibrary]\SteamApps\common\Cities_Skylines
- **On Mac** [SteamLibrary]/SteamApps/common/Cities_Skylines
- **On Linux** [SteamLibrary]/SteamApps/common/Cities_Skylines

Structure AppData

- Root
- Addons
 - **Assets** - Assets saved from the Asset Editor are stored here
 - **ColorCorrections** - Custom color corrections look-up tables can be placed in this folder
 - **Import** - All the assets (textures, models) to import custom assets in the Asset Editor are placed here
 - **MapEditor**
 - * **Brushes** - Custom made brush textures can be placed here
 - * **Heightmaps** - Heightmaps to import are placed in this folder, exported heightmaps from the Map Editor also will appear in this folder
 - **Mods** - Base folder for user made code modifications

- **Maps** - Maps saved from the `Map Editor` are stored here
- **Saves** - Save games are stored here
- **Snapshots*** - Snapshots taken from the `Map Editor` and the `Asset Editor` are stored in subfolders named after their unique id
- **WorkshopStagingArea**** - Steam Workshop content is shown in subfolders here before being uploaded

* - *These folders are never automatically cleaned up, that means the more snapshots a user take, the more images and subfolders there will be. It is convenient when working for a long time with multiple maps to preserve older snapshots to be reused. The folder can be deleted manually with no side effects for the game if needs arise.* ** - *These folders exist only during the time they are needed.*

Additionally, the game may use an additional “Temp” folder if needed. The folder is created under `Addons` and does not require any human interactions.

Structure SteamApps

Only the directories you need are listed

- **Root**
- **Cities Skylines Soundtrack** - If you have bought the Deluxe version of the game this is where you can find the bonus soundtracks.
- **Cites_Data** - **The `output_log.txt` is stored here which might contain errors from your mod**
 - **Managed** - All the DLL's you need are listed here.
- **Locale** - The locale files for all supported languages.
- **Files** - **All the external files the game uses including some default mods and locale files.**
 - **Mods** - Three default mods (`HardMode`, `UnlimitedMoney`, `UnlockAll`) and a template mod (`NewMod`) as reference.

This folder also contains the Digital Artbook and Monumental Buildings PDF.

Files

Cities: Skylines uses an in-house Colossal Raw Asset Package (.crp) file format to store various data. Those packages are containers and can encapsulate any data type, so a .crp file can be a save, a map, a color correction or an asset. The game will output packages in their respective designated place so it is safe to assume a Savegame produced by the game will always be written to the `Saves` folder. For assets importing, standard image formats such as `png`, `jpg`, `bmp`, `tga`, `dds`, `raw`, `r8`, `r16`, and `tiff` are supported, but depending on the tool you are using those with, only a subset may be available. Please refer to the tool documentation for more details. For geometry/models/meshes, the `FBX` file format is the only one officially supported.

Modding

Warning: Read this list of things that that can break your mod and/or people their game!

Getting Started

To create a mod you will have to write down `c#` source code. The game is made in [Unity 5](#) and this allows you to do almost anything you can do in Unity. The source code you write for your mod will be compiled in to a DLL. This DLL is your mod and this will also be placed on the workshop.

Compiling

The game comes with a compiler to compile the dll for your mod. However, this compiler only compiles your mod when you use the `ICities` API. The API is currently very limited and you can't do very much with it. Now the good news is that we also have access to all other classes. This means you can access almost everything in the game but the only downside is that you can't use the in game compiler. We have set up some guides to make it a lot easier to compile your mod within Visual Studio or MonoDevelop.

Setting Up Visual Studio

Install Visual Studio 2013 If you don't own a copy of Visual Studio 2013, you can download [Visual Studio Community 2013 for free](#). Installing Visual Studio should be straightforward.

Create a new solution and project

1. Start Visual Studio
2. Click *File* → *New* → *Project...* or press `Ctrl+Shift+N`
3. Select the *Class Library* template for the *Visual C#* language (other .NET languages should work, but I haven't tried that)
4. Enter a name for your project
5. Choose any location you'd like for your project, but don't use the mod directory
6. Click *OK* to create the solution

Add references

1. Right-click your project in the Solution Explorer and choose *Add* → *Reference...*
2. Click the *Browse...* button at the bottom of the dialog
3. Navigate to `[SteamLibrary]\SteamApps\common\Cities_Skylines\Cities_Data\Managed`, where `[SteamLibrary]` is the path where your Steam games are.
4. Select the [assembly DLL\(s\)](#) you want to use.

Code Create your base class implementing `ICities.IUserMod` as usual. Add as many classes in as many source files as you like. They will all be compiled to a single DLL.

Compile Press `F6` to compile your mod. The Output Type in your project's Properties should be set to Class Library. The compiled DLL will be placed in `obj\Debug` or `obj\Release` under the project folder, depending on the active configuration. You can switch configurations with the combo box in the toolbar.

Test Create a directory for your mod in `%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\` and copy the compiled DLL to it. It should now be available in the Content Manager of Cities: Skylines.

Automate To automatically copy the DLL to the mod directory after each build, follow these steps:

1. Right-click your project in the *Solution Explorer* and choose *Properties*
2. Select *Build Events* on the left hand side of the property sheet
3. Paste the following in the *Post-build event command line*:

```
mkdir "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\$(SolutionName) "
del "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\$(SolutionName)\$(TargetFileName) "
xcopy /y "$(TargetPath)" "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\$(SolutionName)\$(TargetFileName) "
```

This assumes that your mod directory has the same name as your solution. If it doesn't you can change `$(SolutionName)` to the directory of your mod.

Optionally, you can automate the launching of the game (to save those precious seconds of clicking in steam):

```
"%STEAMDIRECTORY%\Steam.exe" -applaunch 255710
```

4. To make the game reload your mod while running, change the last two lines in `AssemblyInfo.cs` (under *Properties* in the *Solution Explorer*) to read:

```
[assembly: AssemblyVersion("1.0.*")]
//[assembly: AssemblyFileVersion("1.0.0.0")]
```

Kudos to [reimarvin](#) for this post on reddit.

Kudos to [walrus_pug](#) for the auto updating with the `AssemblyVersion`.

Setting Up MonoDevelop

Install MonoDevelop If you don't own a copy of MonoDevelop, you can download [MonoDevelop for free](#). Installing MonoDevelop should be straightforward.

Setup a new project/solution

1. Start MonoDevelop and create a new "solution".
2. Put the source wherever you want except for the actual modding-directory the Modding API wiki suggests. You don't want the game to compile the source files.
3. Make it a C# Library, we don't need it to be executable.

Let's assume your project is called `FooBar`. You should get a solution with a `MyClass.cs` file with a tiny bit of code (namespace `FooBar` and a public class).

Set references

1. In Solution, right click `References - Edit References`. (If you don't see a "Solution" panel on the left side, go to `View - Pads - Solution`.)
2. I don't know if you can use the default System library, but just in case I remove it from "Selected references".

3. In the tab .NET Assembly, navigate to your Steam library folder, and then to SteamLibrary\SteamApps\common\Cities_Skylines\Cities_Data\Managed
4. Select the assembly DLL(s) you want to use.

Test autocompletion You should now be able to write a simple class like ...

```
public class myLoader : LoadingExtensionBase {
}
```

... and have the program autocomplete the LoadingExtensionBase keyword. Inside, you can then hit Alt+Insert to generate stubs for the interface. And inside those methods, you should be able to access the classes from ColossalManaged.dll and Assembly-CSharp.dll. For example, you should be able to use DebugOutputPanel.AddMessage() (which writes to the ingame Debug Console).

Compiling

1. You can hit F7 to build your solution (or go to Build - Build FooBar). This should generate warning/error messages if you make a mistake, etc.
2. By default, the project should be set to the “Debug” configuration (see Project - Active Configuration). Thus, F7 creates a directory bin\Debug\ (relative to your project file) and fills it with the necessary .dll-files and your Mod’s .dll.
3. Now you need to copy that .dll to the appropriate folder (see wiki) in its own sub-folder (e.g. “..\Addons\Mods\FooBar\FooBar.dll”).

Testing the Mod

1. Start the game
2. Hit F7 to open the Debug Console. It should tell you that it didn’t find the source for FooBar, but it should still be selectable in Content Manager - Mods. (Ignore the error it’s harmless)

Automate This will copy dll to Mods folder and then delete it before the next build so the new dll can be copied.

1. Go to Project - FooBar Options
2. Go to Build - Custom Commands, select Debug as configuration
3. From the drop down, add “**After Build**“
 - Command:

```
xcopy /Y "bin\${ProjectConfigName}\${SolutionName}.dll" "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\${SolutionName}\${SolutionName}.dll"
```

- Working Directory: \${ProjectDir}
- Check “Run on external console” (you can check Pause, too, to debug)

4. From the drop down, add “**Before Build**“
 - Command:

```
cmd /c "IF EXIST "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\${SolutionName}\${SolutionName}.dll" del "%LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\${SolutionName}\${SolutionName}.dll"
```

- Working Directory: \${ProjectDir}
- check “Run on external console” (you can check Pause, too, to debug)

It should look similar to [this](#). Now whenever you build this project, MonoDevelop deletes your old .dll (IF it exists), then after the build is complete the new .dll is copied via xcopy.

Kudos to [permutation](#) for this post on reddit.

Without an editor

If you are fine with just using the API you don't have to use an editor. You will still need a text editor like notepad or sublime text to write down your c# classes.

- Go to the mods folder in local appdata. (See [Folder Structure](#)) %LOCALAPPDATA%\Colossal Order\Cities_Skylines\Addons\Mods\
 - Inside that folder create a sub folder for your mod with the name of your mod.
 - Create a sub folder inside your mod folder with the name 'Source'.
 - Inside the source folder create a c# file. (The name can be anything but it's recommended to use the name of your mod) C# files have a extension of .cs!

Now you can start creating your mod the same way as you do with the other methods described above.

Starting a mod

Namespace

First of all you have to define a [namespace](#) for your mod. All the classes you create should be wrapped inside this namespace. The namespace should be the name of your mod or at least something very similar. You can see the namespace as a container for all your code to organize things.

```
1 namespace UnlockAllMod {
2     //Inside here you define your class or multiple classes.
3 }
```

When using code from the API, Unity, other mods, the game or anything else you will have to let the compiler know about this. For example to use the API you will be using the ICities namespace. On the top of your classes above where you define your own namespace you specify all the namespaces you will be using.

```
1 using System;
2 using ICities;
3 using UnityEngine;
4 //etc...
```

Mod information

If you look in the Content Manager you will see all mods have a name followed with a description. You will have to define this in your mod else your mod wont be recognized by the game. To set the name and the description you have to inherit the IUserMod interface from the ICities namespace.

```
1 public class MyModInfo : IUserMod {
2     public string Name {
3         get { return "My mod name"; }
4     }
5
6     public string Description {
7         get { return "My mod description"; }
8     }
9 }
```

```
8     }  
9 }
```

Multiple classes

To keep things organized you should work with multiple classes/files. Most people prefer just having one class per file and this is the most common way to work. This does not mean that your mod won't work if you put all your code in one class/file. You can even create more folders inside your Source folders to keep things even more organized.

Example

With the information given above you should now have something like this. When you compile this the mod should be in the Content Manager. However, we haven't really coded anything so the mod will just load in and do nothing.

MyModInfo.cs

```
1 using System;  
2 using ICities;  
3  
4 namespace MyMod {  
5     public class MyModInfo : IUserMod {  
6         public string Name {  
7             get { return "My mod name"; }  
8         }  
9  
10        public string Description {  
11            get { return "My mod description"; }  
12        }  
13    }  
14 }
```

Next step

Now that you can create a mod and compile it it's time to do something in the mod. Visit the pages below to learn more what to do next and make your mod actually do something.

- [Workflow](#) - Information how to improve your workflow like debugging etc.
- [Development](#) - This is where you can learn how to make content for your mods.

Workflow

Best Practices

This is a list of best practises. Some of these are very important as they can break people their game! Check this page fruequently for new things.

Consistent Assembly Versioning

The game keeps strict version references, so if you update the assembly version of your mod, old references will break, corrupting the savegame. So, **always use the same assembly version!** See [this reddit post](#) for more information about this issue.

Spaceless Assembly Names

Assembly names should not contain spaces. Assemblies with a space in the name do not serialize properly, thus removing the mod will break serialized objects.

If you have anything else please [contribute](#) to this page or post it on the [forums](#)!

Reverse Engineering

[Reverse engineering](#) is taking apart an object to see how it works in order to duplicate or enhance the object. So in our case we will be taking apart the game source to create mods for it. Here you can read how to do this properly and efficient. There are of course other methods than the ones described here but this method will work fine for most. If you only want to work with the API you don't really need any of these things.

Other mods

The method described below also applies to other mods. This means you can read the source code of other mods and learn from it. It can also be used to check for malicious code.

The game source

It is possible to browse through the source code of the game. This means you can read how the game is build, what all methods do, what classes there are and so on. We have to thanks Padox/CO for not obfuscating the code else this wouldn't be as easy as it is now. Mods can still obfuscate their code but it's really not recommended to do so. People will most likely report your mods as it can't really be trusted. Even with obfuscation it's possible to read the code but it makes it a lot harder and a lot easier to put in malicious code.

Decompiling

To decompile the assembly and browse through it, you will need a .NET compiler. There are many programs that can do this but for this guide we will be using ILSpy. Some other good alternatives are [DotPeek](#), [JustDecompile](#) or built in options in your IDE.

- [Download ILSpy here](#) (Download binaries)
- Extract the ZIP somewhere on your hard drive (It doesn't have an installer)
- Open The program. (ILSpy.exe for Windows)
- Load in the [needed assemblies](#) (.dll) *File* → *Open*

You can now browse through all the classes in the assembly. And when you click on any of the classes they will open up and you can see all the code. Remember that you can **search**!

Using the ModTools mod

Thanks to [nlight](#) we now have a mod that can help with reverse engineering. This mod has a scene explorer which allows you to examine all objects in the scene. It doesn't just list the objects it also shows all the components, fields, properties and methods. This will help you a lot with understanding how the game works and find the parts of the code you need. It also has some other useful features like logging exceptions in the console so make sure to check it out! [ModTools by nlight](#).

More things to reverse engineer

There are several other things in the game which can be useful or required for your mod. For example the locale files, crp files and so on. Check out the following pages to learn more about this.

Locale Files Information not available yet. For now check out [this page](#).

.crp Files Information not available yet. For now check out [this page](#).

Debugging

The game has a debug panel which you can add messages to.

Using the panel

To open the panel press **F7** in game. You can resize the panel and drag it around. The panel also has a clear button, this will clear all the messages.

Reference the DLL

To be able to use this method of logging you will have to add a reference to the **Assembly-CSharp DLL**. If you're not sure how to do this see this guide for [Visual Studio](#) or for [MonoDevelop](#). And you'll also have to add this line at the top of your class to be able to use it.

```
using ColossalFramework.Plugins;
```

Logging

To send a message to the debug panel you use this code.

```
DebugOutputPanel.AddMessage(PluginManager.MessageType.Message, "Message");
```

you can easily create some static methods for logging. It's also recommended to prefix you log message so that users know where the message came from. For example: "[TreeBrush] Some debug message.."

Logging an objects properties

If you want to print out all the properties of an object, you can use the method below.

```
using System;
using System.ComponentModel;

public void dumpObject(object myObject)
{
    string myObjectDetails = "";
    foreach (PropertyDescriptor descriptor in TypeDescriptor.GetProperties(myObject))
    {
        string name = descriptor.Name;
        object value = descriptor.GetValue(myObject);
        myObjectDetails+= name + ": " + value + "\n";
    }
}
```

```
}  
    DebugOutputPanel.AddMessage(PluginManager.MessageType.Message, myObjectDetails);  
}
```

Other logging methods

There are a few other logging methods that other people made. Here is a list to some of them.

- [ChirpLogger](#)
- [Debugging Deluxe](#)
- *More will be added soon...*

Output-Log

Information about the output_log.txt will be here...

Assemblies

Below are the assemblies you might need when making your mod. There might be some that aren't listed below but these are the ones for Cities: Skylines and Unity.

You can find these assemblies in your steamapps. (See [folder structure](#))
[SteamLibrary]\SteamApps\common\Cities_Skylines

Check out the [reverse engineering page](#) for more information how these can be usefull. You will also need to reference these when setup up your project to be able to use them.

ICities.dll

This is the official modding API.

ColossalManaged.dll

The custom Colossal UI and a lot other framework classes which are used in the game. Things like: DataBinding, Globalization, HTTP, Importers, IO, Math, Packaging, Plugins, Security, Steamworks and Threading.

Assembly-CSharp.dll

This contains almost all of the gameplay related classes. From the road system to citizens and from buildings to the chirper. You will find most things in this assembly.

UnityEngine.dll

The [Unity game engine](#) which the game was built in. Check out the [scripting reference](#) on unity for more information. Remember that this doesn't contain UnityEditor only everything in the UnityEngine namespace.

UnityEngine.UI.dll

The built in **UI** from **Unity**. You can also use **ColossalManaged** to use the UI which the game uses. It completely depends on your preference, in fact you can even use the **old GUI** from Unity but this is not recommended.

Development

How to use ColossalFramework.UI

The **ColossalFramework.UI** is the UI that is used in the game and you can create your own UI components.

The UIView

The **UIView** is a **MonoBehaviour** which is like the main class for the UI. It's like the canvas component in the **UnityUI** this **UIView** has all UI elements in it. So if you're adding a new UI element it will have to be added to the **UIView**.

```
1 UIView v = UIView.GetAView ();
```

Adding a UIComponent

UI components have to be added to the **UIView**. A **UIComponent** is a holder and base class for a UI element like a panel, button, text etc. All these components extend from **UIComponent**. (See list of components below).

```
1 UIComponent uic = v.AddUIComponent (typeof(ExamplePanel));
```

ExamplePanel is the class of the custom panel we create. This class would extend from a **UIComponent** like **UIPanel**.

```
1 public class ExamplePanel : UIPanel {
2 }
```

Setting up a component

When you have created your 'custom' component you can set it up in the **Start()** method. This is a default **Monobehaviour** method people who have used **Unity** should be familiar with this.

```
1 public override void Start () {
2     this.backgroundSprite = "GenericPanel";
3     this.color = new Color32(255,0,0,100);
4     this.width = 100;
5     this.height = 200;
6 }
```

It's quite simple to understand what this does. It just creates a panel 100 wide and 200 high. And then it sets the background sprite of the panel to "GenericPanel" and the color to red half transparent.

Now each component has different properties and you can pretty much customize everything. See the list of components below for a bit of information about how to set up each component.

Adding more stuff (children)

Now that you have a panel you probably want to add some stuff inside of it. It's really easy to add more components inside the component. You don't even have to create a new class for every component you create. So inside the Start() method where you set up the component you can also add more like this:

```
1 UILabel l = this.AddUIComponent<UILabel> ();
2 l.text = "I am a label";
3 l.eventClick += FooBarClickHandler;
```

Again quite straight forward how this works and it shouldn't be that hard to fill in your panel with more components. The only thing that might confuse you is the eventClick += FooBarClickHandler. See the next paragraph for more info about that.

(Input) Events

There are many components that have some sort of user input. For example a button you can click on it and it has a hover state and active state and so on. The behavior of this can easily be modified by appending to the event delegate. As you saw in the example above we used l.eventClick += FooBarClickHandler; The FooBarClickHandler is just a function you can add in your ExamplePanel class. Each method has a different set of arguments for example a slider value change event has the UIComponent as first arg and a float with the value as second arg. If you're using VisualStudio1 or MonoDevelop you can easily generate the method and it will fill in all arguments needed.

List of components

There are more components then these but these are the most commonly used ones. information about components will be added soon!

UIPanel A panel is just a window with a background image. You can add components to this like buttons and so on to fill in your panel.

Properties backgroundSprite (A string that represents a sprite) [TODO need a list of sprites] * flip (?) * atlas (?) * autoLayout (This can be set to true to automatically layout components in the panel) * autoLayoutDirection (A LayoutDirection for the autoLayout option) * padding (A RectOffset to specify the space between the border of the panel and the content) * autoLayoutPadding (, but then for auto layout) * autoFitChildrenHorizontally (This can be set to make all components fit horizontal) * autoFitChildrenVertically (, vectual) * wrapLayout (This will force components in the panel to only render inside the panel) * autoLayoutStart (?) * useCenter (?)

UIScrollablePanel Coming soon...

UIButton Coming soon...

UISprite The UISprite is used for displaying images/sprites.

Properties

- spriteName (A string that represents a sprite) [TODO need a list of sprites]
- spriteInfo (?)

- color (The color for the sprite)
- pixelsToUnits (The amount of pixels there are in one unit (Unity units))
- size (A Vector2 with the width,height of the sprite)
- flip (?)
- invertFill (?)
- fillDirection (?)
- fillAmount (?)
- offset (A vector3 to set the offset vector for the sprite)
- baseIndex (?)

UITextComponent Coming soon...

UILabel Coming soon...

UISlider Coming soon...

List of sprites These are not all the sprites but these have been tested and used.

Getting a list of all sprites TODO: Method to get a list of all sprites.

Sprite list

- name (what it does)

Kudos to [permutation](#) for this post on [reddit](#).

Assets

Importing

Stub section needs to be filled with import methods for stuff like Blender.

Tools

Stub section needs to be filled with information about external tools for making assets as well as in game asset editor tools.

Inspiration

Stub section needs to be filled with inspirational stuff for asset makers.

Tutorials

Stub section for pointing people to asset making tutorials.

Maps

Importing

Stub section needs information for importing terrain maps and finished maps into the game for editing and play respectively.

Tools

Stub section needs information for creating terrain maps and editing them outside and inside of the game.

Resources

UI Sprites 1/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on



AchievementCheckedFalse



AchievementCheckedTrue



AdvisorBubble

AllowWeapons

ArrowLeft

ArrowLeftDisabled



ArrowLeftFocused

ArrowLeftHovered



ArrowLeftPressed



ArrowRight



ArrowRightDisabled

ArrowRightFocused

ArrowRightHovered

ArrowRightPressed

AssetEditorItemBackground

AssetEditorItemBackgroundDisabled

AssetEditorItemBackgroundFocused

AssetEditorItemBackgroundHovered

AssetEditorItemBackgroundPressed

BudgetBarBackground

BudgetSlider

BuildingIcon

Bulldozerbar

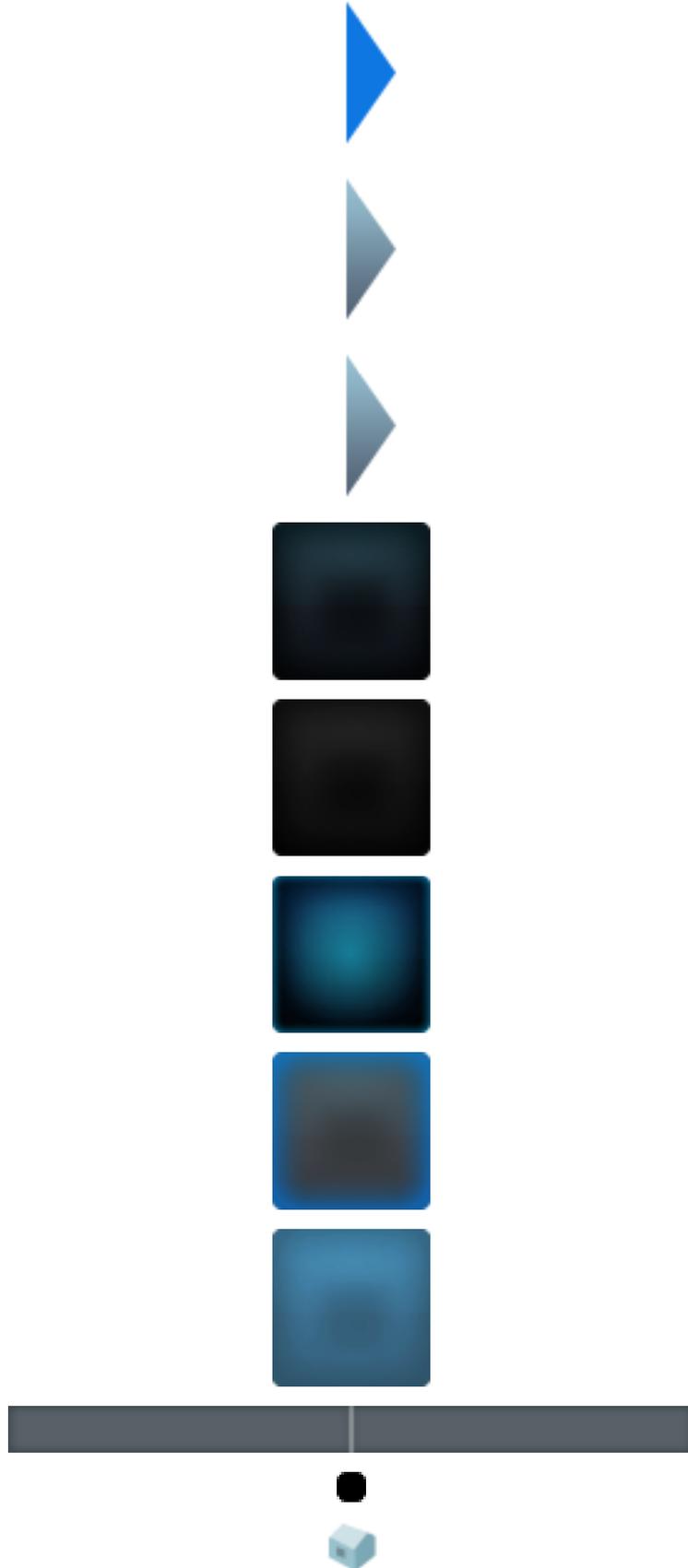
buttonclose

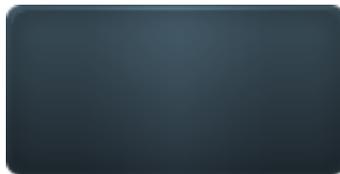
buttonclosehover

buttonclosepressed

ButtonMenu

ButtonMenuDisabled





ButtonMenuFocused



ButtonMenuHovered



ButtonMenuMain



ButtonMenuPressed

ButtonPause

ButtonPauseFocused

ButtonPauseHovered

ButtonPausePressed

ButtonPlay

ButtonPlayFocused

ButtonPlayHovered

ButtonPlayPressed

buttonresize

ButtonTimeLeft

ButtonTimeLeftDisabled

ButtonTimeLeftFocused

ButtonTimeLeftHovered

ButtonTimeLeftPressed

ButtonTimeRight

ButtonTimeRightDisabled

ButtonTimeRightFocused

ButtonTimeRightHovered



ButtonTimeRightPressed



check-checked



check-unchecked



ChirperBubble

ChirperCounter

ChirperDisabled

ChirperFocused

ChirperHovered

ChirperIcon

ChirperPressed

ChirpScrollbarThumb

ChirpScrollbarTrack

CitizenBackground

CityInfo

CityInfoDisabled

CityInfoFocused

CityInfoHovered

CityInfoPressed

ColorPickerColor

ColorPickerIndicator

ColorPickerOutline

ColorPickerOutlineHovered

CursorInfoBack

DemandBack

DistrictOptionBrushLarge

DistrictOptionBrushLargeDisabled

DistrictOptionBrushLargeFocused







DistrictOptionBrushLargeHovered
DistrictOptionBrushLargePressed
DistrictOptionBrushMedium
DistrictOptionBrushMediumDisabled
DistrictOptionBrushMediumFocused
DistrictOptionBrushMediumHovered
DistrictOptionBrushMediumPressed
DistrictOptionBrushSmall
DistrictOptionBrushSmallDisabled
DistrictOptionBrushSmallFocused
DistrictOptionBrushSmallHovered
DistrictOptionBrushSmallPressed
EconomyMoreInfo
EconomyMoreInfoHovered
EducationBoost
EmptyIcon
EmptySprite
esc
FeatureDistricts
FeatureLoans
FeatureMonumentLevel2
FeatureMonumentLevel3
FeatureMonumentLevel4
FeatureMonumentLevel5







About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 2/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on



FeatureMonumentLevel6



FeaturePolicies



FeatureSecondLoan

FeatureTaxes

FeatureThirdLoan

FeatureWonders

Fillsprite

FillspriteDemand

FillspriteOfficeIndustry

frame

GenericPanel

GenericPanelLight

GenericPanelWhite

GenericProgressBar

GenericProgressBarFill

GenericTab



GenericTabDisabled

GenericTabFocused



GenericTabHovered



GenericTabPressed

Heiro-OpenSans-Regular-16px

IconAnimal

IconAssetBuilding

IconAssetCitizen

IconAssetIntersection

IconAssetParks

IconAssetProp

IconAssetTree

IconAssetVehicle

IconCargoShip

IconCitizenVehicle

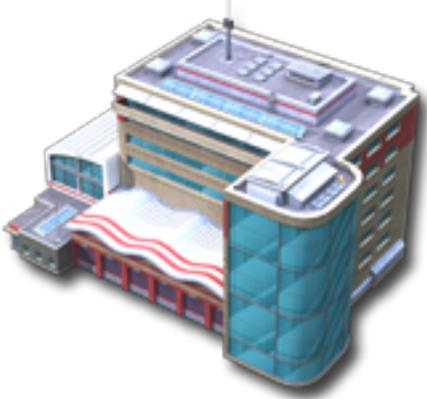
IconDownArrow

IconDownArrowDisabled

IconDownArrowFocused

IconDownArrowHovered

IconDownArrowPressed







IconError

IconLeftArrow



IconMessage



IconPolicyAlligatorBan



IconPolicyAllowWeapons

IconPolicyBaseCircle

IconPolicyBaseCircleDisabled

IconPolicyBaseCircleFocused

IconPolicyBaseCircleHovered

IconPolicyBaseCirclePressed

IconPolicyBaseRect

IconPolicyBaseRectDisabled

IconPolicyBaseRectFocused

IconPolicyBaseRectHovered

IconPolicyBaseRectPressed

IconPolicyBigBusiness

IconPolicyEducationBoost

IconPolicyFarming

IconPolicyFarmingDisabled

IconPolicyFarmingFocused

IconPolicyFarmingHovered





IconPolicyFarmingPressed

IconPolicyForest

IconPolicyForestDisabled

IconPolicyForestFocused

IconPolicyForestHovered

IconPolicyForestPressed

IconPolicyFreePublicTransport

IconPolicyFreeTransport

IconPolicyHeavyTrafficBan

IconPolicyHighriseBan

IconPolicyHighTechHousing

IconPolicyIndustrySpace

IconPolicyNoHomo

IconPolicyNoHomoDisabled

IconPolicyNoHomoFocused

IconPolicyNoHomoHovered

IconPolicyNoHomoPressed

IconPolicyNone

IconPolicyOil

IconPolicyOilDisabled

IconPolicyOilFocused

IconPolicyOilHovered

IconPolicyOilPressed

IconPolicyOre

IconPolicyOreDisabled

IconPolicyOreFocused









IconPolicyOreHovered



IconPolicyOrePressed



IconPolicyParksandRecreation

IconPolicyPetBan

IconPolicyPetBanDisabled

IconPolicyPetBanFocused

IconPolicyPetBanHovered

IconPolicyPetBanPressed

IconPolicyPowerSaving

IconPolicyPowerSavingDisabled

IconPolicyPowerSavingFocused

IconPolicyPowerSavingHovered

IconPolicyPowerSavingPressed

IconPolicyRecreationalUse

IconPolicyRecreationBoost

IconPolicyRecycling

IconPolicyRecyclingDisabled

IconPolicyRecyclingFocused

About this page

This wiki page was created in game with the [SpriteDumper mod](#) . To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.





UI Sprites 3/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on



IconPolicyRecyclingHovered
IconPolicyRecyclingPressed
IconPolicySmallBusiness
IconPolicySmallBusinessEnthusiast
IconPolicySmokeDetectors
IconPolicySmokeDetectorsDisabled
IconPolicySmokeDetectorsFocused
IconPolicySmokeDetectorsHovered
IconPolicySmokeDetectorsPressed
IconPolicySmokingBan
IconPolicySmokingBanDisabled
IconPolicySmokingBanFocused
IconPolicySmokingBanHovered
IconPolicySmokingBanPressed
IconPolicyTaxLowerComHigh
IconPolicyTaxLowerComLow
IconPolicyTaxLowerOffice
IconPolicyTaxLowerResHigh
IconPolicyTaxLowerResLow
IconPolicyTaxRaiseComHigh
IconPolicyTaxRaiseComLow
IconPolicyTaxRaiseIndustrial
IconPolicyTaxRaiseOffice





IconPolicyTaxRaiseResHigh



IconPolicyTaxRaiseResLow



IconPolicyWaterSaving



IconPolicyWaterSavingDisabled

IconPolicyWaterSavingFocused

IconPolicyWaterSavingHovered

IconPolicyWaterSavingPressed

IconRightArrow

IconServiceVehicle

IconSpeed1Hover

IconSpeed1Normal

IconSpeed2Hover

IconSpeed2Normal

IconSpeed3Hover

IconSpeed3Normal

IconTourist

IconTouristVehicle

IconUpArrow

IconUpArrowDisabled

IconUpArrowFocused

IconUpArrowHovered

IconUpArrowPressed

IconWarning

InfoBubble

InfoBubbleService

InfoBubbleVehicle





InfoDisplay



InfoDisplayFocused



InfoDisplayHover



InfoDisplayPressed

InfoIconAge

InfoIconAgeDisabled

InfoIconAgeFocused

InfoIconAgeHovered

InfoIconAgePressed

InfoIconBaseDisabled

InfoIconBaseFocused

InfoIconBaseHovered

InfoIconBaseNormal

InfoIconBasePressed

InfoIconCrime

InfoIconCrimeDisabled

InfoIconCrimeFocused

InfoIconCrimeHovered

InfoIconCrimePressed

InfoIconDistricts

InfoIconDistrictsDisabled

InfoIconDistrictsFocused

InfoIconDistrictsHovered

InfoIconDistrictsPressed

InfoIconEducation

InfoIconEducationDisabled





InfoIconEducationFocused



InfoIconEducationHovered



InfoIconEducationPressed



InfoIconElectricity

InfoIconElectricityDisabled

InfoIconElectricityFocused

InfoIconElectricityHovered

InfoIconElectricityPressed

InfoIconEntertainment

InfoIconEntertainmentDisabled

InfoIconEntertainmentFocused

InfoIconEntertainmentHovered

InfoIconEntertainmentPressed

InfoIconFireSafety

InfoIconFireSafetyDisabled

InfoIconFireSafetyFocused

InfoIconFireSafetyHovered

InfoIconFireSafetyPressed

InfoIconGarbage

InfoIconGarbageDisabled

InfoIconGarbageFocused

InfoIconGarbageHovered

InfoIconGarbagePressed

InfoIconHappiness

InfoIconHappinessDisabled

InfoIconHappinessFocused





About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 4/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on



InfoIconHappinessHovered



InfoIconHappinessPressed



InfoIconHealth

InfoIconHealthDisabled

InfoIconHealthFocused

InfoIconHealthHovered

InfoIconHealthPressed

InfoIconLandValue

InfoIconLandValueDisabled

InfoIconLandValueFocused

InfoIconLandValueHovered

InfoIconLandValuePressed

InfoIconLevel

InfoIconLevelDisabled

InfoIconLevelFocused

InfoIconLevelHovered

InfoIconLevelPressed





InfoIconNoisePollution



InfoIconNoisePollutionDisabled



InfoIconNoisePollutionFocused

InfoIconNoisePollutionHovered

InfoIconNoisePollutionPressed

InfoIconOutsideConnections

InfoIconOutsideConnectionsDisabled

InfoIconOutsideConnectionsFocused

InfoIconOutsideConnectionsHovered

InfoIconOutsideConnectionsPressed

InfoIconPolicies

InfoIconPoliciesDisabled

InfoIconPoliciesFocused

InfoIconPoliciesHovered

InfoIconPoliciesPressed

InfoIconPollution

InfoIconPollutionDisabled

InfoIconPollutionFocused

InfoIconPollutionHovered

InfoIconPollutionPressed

InfoIconPopulation

InfoIconPopulationDisabled

InfoIconPopulationFocused

InfoIconPopulationHovered

InfoIconPopulationPressed

InfoIconPublicTransport





InfoIconPublicTransportDisabled

InfoIconPublicTransportFocused



InfoIconPublicTransportHovered



InfoIconPublicTransportPressed



InfoIconResources

InfoIconResourcesDisabled

InfoIconResourcesFocused

InfoIconResourcesHovered

InfoIconResourcesPressed

InfoIconTrafficCongestion

InfoIconTrafficCongestionDisabled

InfoIconTrafficCongestionFocused

InfoIconTrafficCongestionHovered

InfoIconTrafficCongestionPressed

InfoIconWater

InfoIconWaterDisabled

InfoIconWaterFocused

InfoIconWaterHovered

InfoIconWaterPressed

InfoIconWindSpeed

InfoIconWindSpeedDisabled

InfoIconWindSpeedFocused

InfoIconWindSpeedHovered

InfoIconWindSpeedPressed

InfoPanel

InfoPanelBack





InfoPanelIconCurrency

InfoPanelIconFreecamera



InfoPanelIconGenericTab



InfoPanelIconHappiness



InfoPanelIconInfo

InfoPanelIconPopulation

InfoPanelRCIOIndicator

InfoviewPanel

LevelBarBackground

LevelBarForeground

ListItemHighlight

ListItemHover

loading_icon

LocationMarkerActiveDisabled

LocationMarkerActiveFocused

LocationMarkerActiveHovered

LocationMarkerActiveNormal

LocationMarkerActivePressed

LocationMarkerDisabled

LocationMarkerFocused

LocationMarkerHovered

LocationMarkerNormal

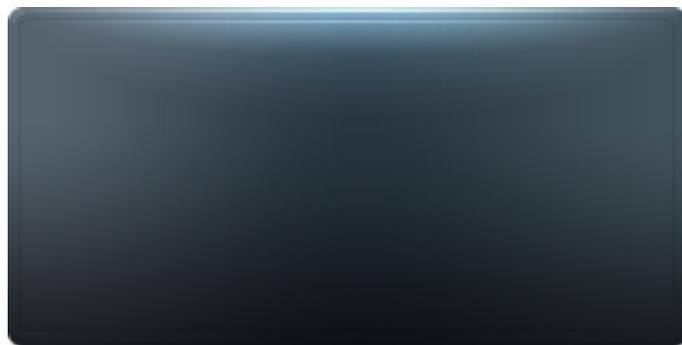
LocationMarkerPressed

LockIcon

Menubar

MenuContainer





MenuPanel



MenuPanel2

MenuPanel3

MenuPanel3Deluxe

MenuPanelInfo

MeterBackground

About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 5/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on

MeterIndicator

MoneyThumb

Niet

NotificationIconDead

NotificationIconExcellentHealth

NotificationIconExtremelyHappy

NotificationIconHappy

NotificationIconHealthy

NotificationIconNotHappy

NotificationIconPoorHealth

NotificationIconUnhappy

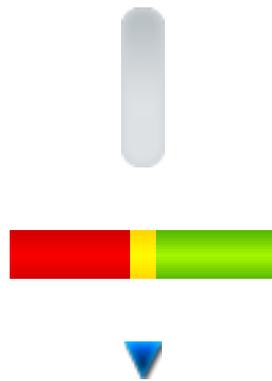
NotificationIconVeryHappy

NotificationIconVeryHealthy

NotificationIconVeryPoorHealth

NotificationIconVerySick









NotificationIcon VeryUnhappy

OptionBase

OptionBaseDisabled

OptionBaseFocused

OptionBaseHovered

OptionBasePressed

Options

OptionsDisabled

OptionsFocused

OptionsHovered

OptionsPressed

paradox-store-logo

PauseOutline

pdxLogo

PieChartBg

PieChartWhiteBg

PieChartWhiteFg

PoliciesBubble

PolicyBarBack

PolicyBarBackActive

PolicyTagBackground

ProgressBarFill

ratingBG

ratingFG

RelocateIcon

ResourceFertility

ResourceFertilityDisabled

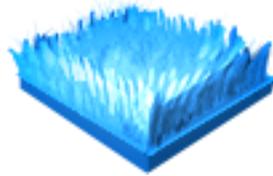
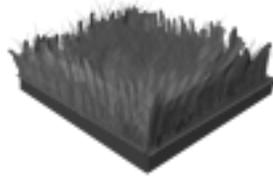




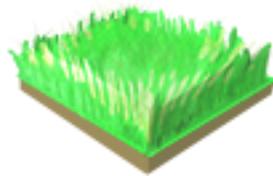
BETA







ResourceFertilityFocused



ResourceFertilityHovered

ResourceFertilityPressed

ResourceForestry

ResourceOil

ResourceOilDisabled

ResourceOilFocused

ResourceOilHovered

ResourceOilPressed

ResourceOre

ResourceOreDisabled

ResourceOreFocused

ResourceOreHovered

ResourceOrePressed

ResourceSand

ResourceSandDisabled

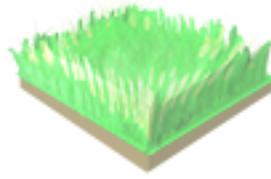
ResourceSandFocused

ResourceSandHovered

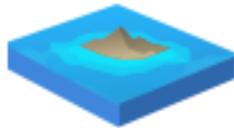
ResourceSandPressed

ResourceWater

RoadOptionCurved







RoadOptionCurvedDisabled



RoadOptionCurvedFocused



RoadOptionCurvedHovered



RoadOptionCurvedPressed

RoadOptionFreeform

RoadOptionFreeformDisabled

RoadOptionFreeformFocused

RoadOptionFreeformHovered

RoadOptionFreeformPressed

RoadOptionStraight

RoadOptionStraightDisabled

RoadOptionStraightFocused

RoadOptionStraightHovered

RoadOptionStraightPressed

RoadOptionUpgrade

RoadOptionUpgradeDisabled

RoadOptionUpgradeFocused

RoadOptionUpgradeHovered

RoadOptionUpgradePressed

RoundBackBig

RoundBackBigDisabled

RoundBackBigFocused

RoundBackBigHovered

RoundBackBigPressed

ScrollbarThumb

ScrollbarTrack





Servicebar

SliderBudget



SliderFill



Snapping



SnappingDisabled

SnappingFocused

SnappingHovered

SnappingPressed

SteamCloud

StopEmptyingIcon

SubBar

SubBarBeautificationParksnPlazas

About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 6/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on

SubBarBeautificationParksnPlazasDisabled

SubBarBeautificationParksnPlazasFocused

SubBarBeautificationParksnPlazasHovered

SubBarBeautificationParksnPlazasPressed

SubBarBeautificationPaths



SubBarBeautificationPathsDisabled

SubBarBeautificationPathsFocused



SubBarBeautificationPathsHovered



SubBarBeautificationPathsPressed



SubBarBeautificationProps

SubBarBeautificationPropsDisabled

SubBarBeautificationPropsFocused

SubBarBeautificationPropsHovered

SubBarBeautificationPropsPressed

SubBarButtonBase

SubBarButtonBaseDisabled

SubBarButtonBaseFocused

SubBarButtonBaseHovered

SubBarButtonBasePressed

SubBarMonumentCategory1

SubBarMonumentCategory1Disabled

SubBarMonumentCategory1Focused

SubBarMonumentCategory1Hovered

SubBarMonumentCategory1Pressed

SubBarMonumentCategory2

SubBarMonumentCategory2Disabled

SubBarMonumentCategory2Focused

SubBarMonumentCategory2Hovered

SubBarMonumentCategory2Pressed

SubBarMonumentCategory3

SubBarMonumentCategory3Disabled

SubBarMonumentCategory3Focused

SubBarMonumentCategory3Hovered

SubBarMonumentCategory3Pressed



III

III

SubBarMonumentCategory4

SubBarMonumentCategory4Disabled

SubBarMonumentCategory4Focused

SubBarMonumentCategory4Hovered

SubBarMonumentCategory4Pressed

SubBarMonumentCategory5

SubBarMonumentCategory5Disabled

SubBarMonumentCategory5Focused

SubBarMonumentCategory5Hovered

SubBarMonumentCategory5Pressed

SubBarMonumentCategory6

SubBarMonumentCategory6Disabled

SubBarMonumentCategory6Focused

SubBarMonumentCategory6Hovered

SubBarMonumentCategory6Pressed

SubBarMonumentLandmarks

SubBarMonumentLandmarksDisabled

SubBarMonumentLandmarksFocused

SubBarMonumentLandmarksHovered

SubBarMonumentLandmarksPressed

SubBarPublicTransportBus

SubBarPublicTransportBusDisabled

SubBarPublicTransportBusFocused

SubBarPublicTransportBusHovered

SubBarPublicTransportBusPressed

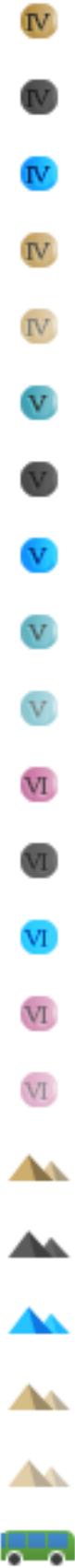
SubBarPublicTransportMetro

SubBarPublicTransportMetroDisabled

SubBarPublicTransportMetroFocused

SubBarPublicTransportMetroHovered

III





SubBarPublicTransportMetroPressed

SubBarPublicTransportPlane

SubBarPublicTransportPlaneDisabled

SubBarPublicTransportPlaneFocused

SubBarPublicTransportPlaneHovered

SubBarPublicTransportPlanePressed

SubBarPublicTransportShip

SubBarPublicTransportShipDisabled

SubBarPublicTransportShipFocused

SubBarPublicTransportShipHovered

SubBarPublicTransportShipPressed

SubBarPublicTransportTrain

SubBarPublicTransportTrainDisabled

SubBarPublicTransportTrainFocused

SubBarPublicTransportTrainHovered

SubBarPublicTransportTrainPressed

SubBarRoadsHighway

SubBarRoadsHighwayDisabled

SubBarRoadsHighwayFocused

SubBarRoadsHighwayHovered

SubBarRoadsHighwayPressed

SubBarRoadsIntersection

SubBarRoadsIntersectionDisabled

SubBarRoadsIntersectionFocused

SubBarRoadsIntersectionHovered

SubBarRoadsIntersectionPressed

SubBarRoadsLarge

SubBarRoadsLargeDisabled

SubBarRoadsLargeFocused







SubBarRoadsLargeHovered



SubBarRoadsLargePressed



SubBarRoadsMedium



SubBarRoadsMediumDisabled

SubBarRoadsMediumFocused

SubBarRoadsMediumHovered

SubBarRoadsMediumPressed

SubBarRoadsSmall

SubBarRoadsSmallDisabled

About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 7/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on

SubBarRoadsSmallFocused

SubBarRoadsSmallHovered

SubBarRoadsSmallPressed

SubcategoriesPanel

TextFieldPanel

TextFieldPanelHovered

TextFieldUnderline

ThumbnailJunctionsClover

ThumbnailJunctionsCloverDisabled





ThumbnailJunctionsCloverFocused

ThumbnailJunctionsCloverHovered

ThumbnailJunctionsCloverPressed

ThumbnailTrophy

ThumbStatistics

ToggleBase

ToggleBaseDisabled

ToggleBaseFocused

ToggleBaseHovered

ToggleBasePressed

Toolbar

ToolbarIconBaseDisabled

ToolbarIconBaseFocused

ToolbarIconBaseHovered

ToolbarIconBaseNormal

ToolbarIconBasePressed







ToolbarIconBeautification

ToolbarIconBeautificationDisabled

ToolbarIconBeautificationFocused

ToolbarIconBeautificationHovered

ToolbarIconBeautificationPressed

ToolbarIconBulldozer

ToolbarIconBulldozerDisabled

ToolbarIconBulldozerFocused

ToolbarIconBulldozerHovered

ToolbarIconBulldozerPipes

ToolbarIconBulldozerPipesDisabled

ToolbarIconBulldozerPipesFocused

ToolbarIconBulldozerPipesHovered

ToolbarIconBulldozerPipesPressed

ToolbarIconBulldozerPressed

ToolbarIconDistrict

ToolbarIconDistrictDisabled

ToolbarIconDistrictFocused

ToolbarIconDistrictHovered

ToolbarIconDistrictPressed

ToolbarIconEducation

ToolbarIconEducationDisabled

ToolbarIconEducationFocused

ToolbarIconEducationHovered

ToolbarIconEducationPressed









ToolbarIconElectricity



ToolbarIconElectricityDisabled

ToolbarIconElectricityFocused

ToolbarIconElectricityHovered

ToolbarIconElectricityPressed

ToolbarIconFireDepartment

ToolbarIconFireDepartmentDisabled

ToolbarIconFireDepartmentFocused

ToolbarIconFireDepartmentHovered

ToolbarIconFireDepartmentPressed

ToolbarIconGarbage

ToolbarIconGarbageDisabled

ToolbarIconGarbageFocused

ToolbarIconGarbageHovered

ToolbarIconGarbagePressed

ToolbarIconGovernment

ToolbarIconGovernmentDisabled

ToolbarIconGovernmentFocused

ToolbarIconGovernmentHovered

ToolbarIconGovernmentPressed

ToolbarIconGroup1Disabled

ToolbarIconGroup1Focused

ToolbarIconGroup1Hovered

ToolbarIconGroup1Normal

ToolbarIconGroup1Pressed

ToolbarIconGroup2Disabled





ToolbarIconGroup2Focused



ToolbarIconGroup2Hovered

ToolbarIconGroup2Normal



ToolbarIconGroup2Pressed

ToolbarIconGroup3Disabled

ToolbarIconGroup3Focused

ToolbarIconGroup3Hovered

ToolbarIconGroup3Normal

ToolbarIconGroup3Pressed

ToolbarIconGroup4Disabled

ToolbarIconGroup4Focused

ToolbarIconGroup4Hovered

ToolbarIconGroup4Layer 6

ToolbarIconGroup4Normal

ToolbarIconGroup4Pressed

ToolbarIconGroup5Disabled

ToolbarIconGroup5Focused

ToolbarIconGroup5Hovered

ToolbarIconGroup5Normal

ToolbarIconGroup5Pressed

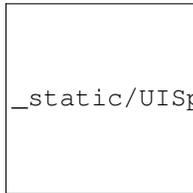
ToolbarIconGroup6Disabled

ToolbarIconGroup6Focused

ToolbarIconGroup6Hovered

ToolbarIconGroup6Normal

ToolbarIconGroup6Pressed



`_static/UISprites/ToolbarIconGroup4Layer6.png`





About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

UI Sprites 8/8

Here you can find a list of UI sprites. These are all the sprites available in the game. You can use these sprites in many of the UI components that have a sprite property for example a button, panel, slider and so on



ToolbarIconHealthcare



ToolbarIconHealthcareDisabled



ToolbarIconHealthcareFocused

ToolbarIconHealthcareHovered

ToolbarIconHealthcarePressed

ToolbarIconHelp

ToolbarIconMoney

ToolbarIconMoneyDisabled

ToolbarIconMoneyFocused

ToolbarIconMoneyHovered

ToolbarIconMoneyPressed

ToolbarIconMonuments

ToolbarIconMonumentsDisabled

ToolbarIconMonumentsFocused

ToolbarIconMonumentsHovered

ToolbarIconMonumentsPressed





ToolbarIconPolice

ToolbarIconPoliceDisabled



ToolbarIconPoliceFocused

ToolbarIconPoliceHovered

ToolbarIconPolicePressed

ToolbarIconPolicies

ToolbarIconPoliciesDisabled

ToolbarIconPoliciesFocused

ToolbarIconPoliciesHovered

ToolbarIconPoliciesPressed

ToolbarIconProps

ToolbarIconPropsDisabled

ToolbarIconPropsFocused

ToolbarIconPropsHovered

ToolbarIconPropsPressed

ToolbarIconPublicTransport

ToolbarIconPublicTransportDisabled

ToolbarIconPublicTransportFocused

ToolbarIconPublicTransportHovered

ToolbarIconPublicTransportPressed

ToolbarIconRoads

ToolbarIconRoadsDisabled

ToolbarIconRoadsFocused

ToolbarIconRoadsHovered

ToolbarIconRoadsPressed

ToolbarIconTemp





ToolbarIconUnlock



ToolbarIconUnlockRadial



ToolbarIconWaterAndSewage



ToolbarIconWaterAndSewageDisabled

ToolbarIconWaterAndSewageFocused

ToolbarIconWaterAndSewageHovered

ToolbarIconWaterAndSewagePressed

ToolbarIconWonders

ToolbarIconWondersDisabled

ToolbarIconWondersFocused

ToolbarIconWondersHovered

ToolbarIconWondersPressed

ToolbarIconZoning

ToolbarIconZoningDisabled

ToolbarIconZoningFocused

ToolbarIconZoningHovered

ToolbarIconZoningPressed

ToolbarIconZoomOutbase

ToolbarIconZoomOutbaseHovered

ToolbarIconZoomOutbasePressed

ToolbarIconZoomOutbasetDisabled

ToolbarIconZoomOutbasetFocused

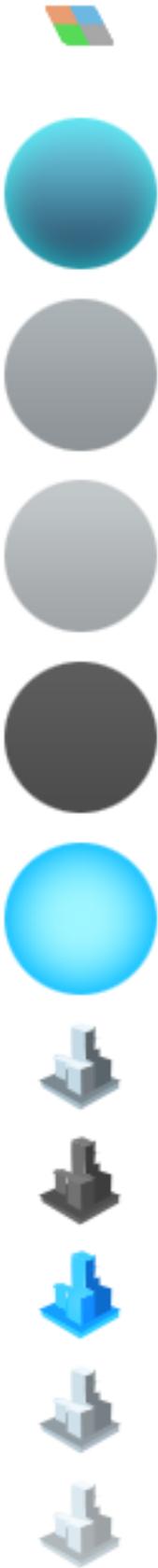
ToolbarIconZoomOutCity

ToolbarIconZoomOutCityDisabled

ToolbarIconZoomOutCityFocused

ToolbarIconZoomOutCityHovered





ToolbarIconZoomOutCityPressed



ToolbarIconZoomOutGlobe



ToolbarIconZoomOutGlobeDisabled



ToolbarIconZoomOutGlobeFocused

ToolbarIconZoomOutGlobeHovered

ToolbarIconZoomOutGlobePressed

TutorialBubble

TutorialBubbleFlipRight

TutorialBubbleLeft

TutorialBubbleRight

TutorialGlow

TutorialHighlite

UnlockingArrowLeft

UnlockingArrowLeftDisabled

UnlockingArrowLeftFocused

UnlockingArrowLeftHovered

UnlockingArrowLeftPressed

UnlockingArrowRight

UnlockingArrowRightDisabled

UnlockingArrowRightFocused

UnlockingArrowRightHovered

UnlockingArrowRightPressed

UnlockingBackground

UnlockingBar

UnlockingItemBackground

UnlockingItemBackgroundDisabled









UnlockingItemBackgroundFocused

UnlockingItemBackgroundHovered



UnlockingItemBackgroundPressed



UnlockingPanel

UnlockingPanel2

UnlockingProgressBar

UnlockingProgressBarFill

About this page

This wiki page was created in game with the [SpriteDumper mod](#). To modify the text in this document please create a PR on the mod on github. If there are sprites missing you can run the mod and create a PR on the docs repo with the new generated file.

Kudos to [Permutation](#) for sharing the method for dumping sprites.

About The Docs

Frequently Asked Questions

Authors and Contributors

So far, the only contributing member to this project is DarkArcana. If you'd like to contribute, check the FAQ (empty right now!) or [check this forum post](#).



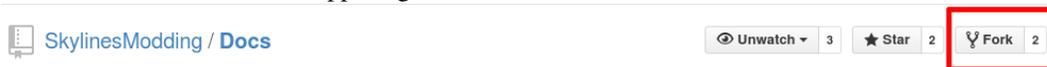
How to Contribute to the Docs

Warning: Pull Requests will **only** be accepted if source files have correct .rst formatting and Sphinx directives. Learn more at [sphinx-doc](#).

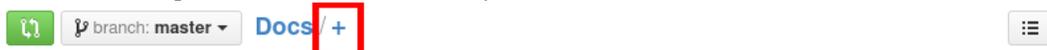
Web

This is the easiest way to contribute. You don't need to know exactly how git works, you just need to be able to interact with Github's web interface. Before you can do anything below, you must login to Github or create an account. Afterwards contributing is as simple as following these steps:

- Fork
 - You can't edit the repository directly because the master branch is protected. You have to make a fork where your edits are made that will later be merged back into the master branch if your edits are acceptable. Navigate to the [Documentation Repository](#) on Github and click the Fork button in the upper right corner:



- Add a file
 - Adding a file is also really simple. In each directory listing, there is a + symbol that you can press to create a new directory or file.



- Edit a file
 - Just choose a file you want to edit by clicking on it in the listing:



..		
static	Add favicon, adjust html short title	17 hours ago
assets	Restructure Assets indexing	15 hours ago
maps	Restructure Maps indexing	15 hours ago
modding	Fix several formatting issues in 'Setting Up and Using Visual Studio ...	3 hours ago
authors.rst	Add a small note to the authors file to be filled later when people m...	17 hours ago
conf.py	Killed the source links on all pages and the Copyright which is usele...	15 hours ago
contributing.rst	Add some content to the main index to get an idea how the whole thing...	2 days ago
faq.rst	Add some content to the main index to get an idea how the whole thing...	2 days ago
index.rst	Restructure Modding indexing	15 hours ago
license.rst	Add some content to the main index to get an idea how the whole thing...	2 days ago

– Then click on the edit pencil in the upper right corner of the file:



– Commit your changes with a good comment that explains what you did:

Commit changes

Fix formatting

Add an optional extended description...

Commit directly to the `master` branch

Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

- Make a Pull Request

– Compare and Review



– Double check that all of your committed changes are good and conform to RST formatting. Remember, bad formatting will **NOT** be accepted. If everything looks ok and you're sure your commits can be merged, go ahead and press the Create Pull Request Button:



Local

This method assumes you have some knowledge of the command line and git.

- First Time
 - Fork the repo like above.

```
cd /some/empty/directory
# Remember that you can get this link from the sidebar on GitHub
git clone https://github.com/<your name>/Docs.git
cd Docs
```

- All of the source files are in docs/source. Use your favorite editor to make changes. When you're done:

```
git add some/file/you/changed
git commit -m "Useful commit message explaining what you changed"
git push
```

- Back on GitHub, make a Pull Request.

Advanced

The best way to test your changes is to run them locally and see how they will work before they are merged. You can check any formatting errors in real time and reduce the work load upstream. This requires that you have the repository locally and have some up to date version of Python installed.

- Setting It Up
 - While in the Docs repo directory:

```
pip install virtualenv

# It has to be called venv because that's the name used in the .gitignore.
# If you use any other name, your future pull requests will not be accepted.

virtualenv venv

# Then you need to enter the virtual environment
# On Linux:
source venv/bin/activate

# On Windows:
venv/bin/activate

# use ``deactivate`` to get out of the virtual environment
# Once inside, you need to install all of the requirements for the system.
# Make sure you are inside the repository directory.
pip install -r requirements.txt
```

- Make Your Changes
- Test Your Changes
 - Inside the docs directory, just above source, you will see Makefile and make.bat. While inside the virtual environment, you can use these to issue commands to Sphinx to build the documentation.

```
# Before each run to make sure the build environment is clean.
make clean

# Then build the html docs, we don't need anything else.
make html
```

- Now go ahead and test your changes. You can open a file from the file browser or launcher a web browser from command line:

```
# Launch the index in Google Chrome
google-chrome source/index.rst
```

- Double check that everything looks good. If it does, make your commits and Pull Request as outlined above.

Live preview

To see the changes as you type you can use this [live preview editor](#). It does not format everything 100% correct but it formats things better than github. After you have wrote the page in the editor you can use the same methods described above. You will just have to paste in the code from the editor.

License

The MIT License (MIT)

Copyright (c) 2015 SkylinesModding

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Links

- [Official C:S Website](#)
- [Official C:S Wiki](#)
- [C:S on Steam](#)
- [C:S Steam Workshop](#)
- [CitiesSkylinesModding](#)
- [/r/CitiesSkylines](#)
- [/r/CitiesSkylinesModding](#)
- [/r/CSModAudits](#)

Indices and tables

- `genindex`
- `modindex`
- `search`