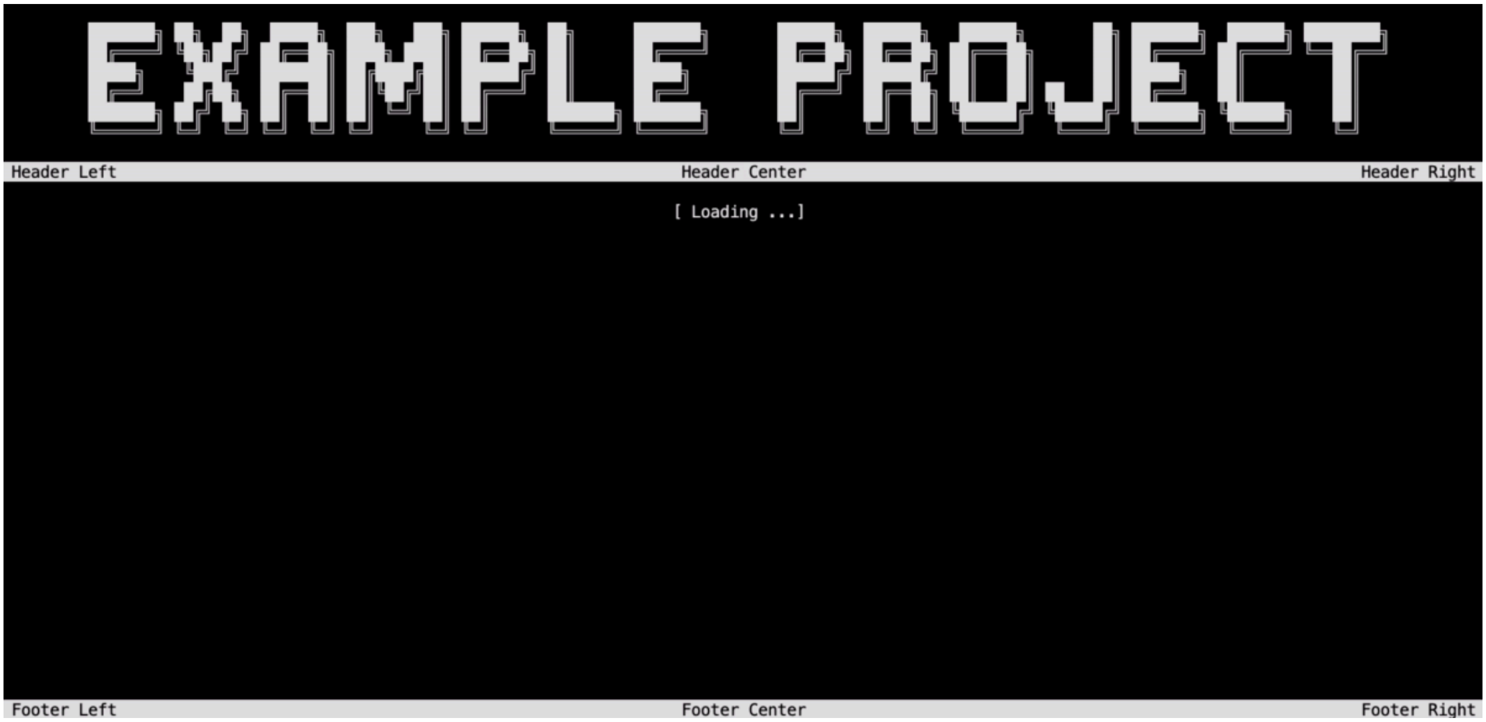
Welcome to the docs

version v3.6.0 downloads 11k stars 30 coverage 96% License GPL v2.0 Code lines 29k



EXAMPLE PROJECT

```
Header Left                                Header Center                                Header Right
                                            [ Loading ...]

Footer Left                                Footer Center                                Footer Right
```

Table of contents

Welcome to the documentation of the `ConsoleAppVisuals` library. This is a simple and easy-to-use library that allows you to create visual elements in the console. Here are all the resources available:

- Introduction: find the basic concepts and the first steps into the library to learn how to use it.
- Elements: find all the visual elements available in the library with their description.
- References: find all methods, properties and classes with their description and all arguments available.
- Examples: find some examples to learn how to use the library in concrete use cases.
- Articles: find some additional library-related articles.
- Legacy: find the outdated documentation of the library for the versions 2.x.x and below.

Roadmap

The library is still in active development. The next feature and bug resolutions are listed in the Project section of the GitHub repository.

Supported .NET versions

Version	Supported
net9.x	✅
net8.x	✅
net7.x	✅
net6.x	✅
< net6.x	❌

Security Policy

Consider reading our SECURITY policy to know more about how we handle security issues and how to report them. You will also find the stable versions of the project.

Acknowledgments

Consider reading the ACKNOWLEDGMENTS file. It's a testament to the collaborative effort that has gone into improving and refining our library. We're deeply grateful to all our contributors for their invaluable input and the significant difference they've made to the project.

It also lists the open source projects that have been used to build this library until now.

Contributing

Contributions are what make the open source community such an amazing place to learn, inspire, and create. Any contributions you make are **greatly appreciated**. To do so, follow the steps described in the CONTRIBUTING file.

We are always open for feedback and discussions. If you are using our library and want to share your use case, or if you have any suggestions for improvement, please feel free to open an issue or open a discussion on our GitHub repository. Your input helps us understand possible use cases and make necessary improvements.

Do not hesitate to **star** and **share** the project if you like it!

Table of content

In this section, you will find the list of all the elements available in the library. They will be accompanied by a brief description and a link to their respective documentation page.

They are divided into three categories:

- Passive: the default elements, which do not require any specific action from the user.
- Interactive: like a menu or prompt, these elements require user input and interaction.
- Animated: elements that are animated, like a progress bar.

> **NOTE**
>
> An article is available to help you create your own elements. You can find it here.

Have a question, give a feedback or found a bug? Feel free to open an issue or start a discussion on the GitHub repository.

Getting started

This section describes all references of the library. You will find all arguments, method signatures, classes, and enums that are available for the user.

Access to the namespaces classes and functions at the left side of this page.

> **WARNING**
>
> Adding `using ConsoleAppVisuals;` at the beginning of your C# file is necessary but not sufficient to use the full potential of the library. Refer to the descriptions below to discover which namespaces to add to your project and the Introduction section to see how they are used.

Namespace descriptions

Namespaces are used to organize the classes and interfaces of the library. They are used to avoid naming conflicts and to group the classes that are related to each other. Here are the different namespaces of the library:

`ConsoleAppVisuals`

```
using ConsoleAppVisuals.Elements;
```

This is the main namespace of the library. It contains the `Core` and `Window` classes. The `Core` class is the core of the library interaction with the console. It contains the methods to interact with the console on a low level basis. The `Window` class is used to manage visual elements. You may use it to add, remove, update and display elements on the console.

`PassiveElements`

```
using ConsoleAppVisuals.PassiveElements;
```

This namespace contains all the passive elements of the library. You may find the elements that do not provide interaction with the user and that you may display several from the same type at the same time. They share common characteristics and methods defined in the `models/Element.cs` class.

`InteractiveElements`

```
using ConsoleAppVisuals.InteractiveElements;
```

This namespace contains all the interactive elements of the library. You may find the elements that provide interaction with the user and that you may display only one at a time. They share common characteristics and methods defined in the `models/InteractiveElement.cs` class.

`Enums`

```
using ConsoleAppVisuals.Enums;
```

This namespace contains all the enumerations used in the library. They are used to define the behaviors, position, response of the elements in the console.

`Models`

```
using ConsoleAppVisuals.Models;
```

This namespace contains all the models of the library. They are used to define the characteristics of the elements and the interactions. You may find the `Element`, `InteractiveElement` classes for example.

`Attributes`

```
using ConsoleAppVisuals.Attributes;
```

This namespace contains the `VisualAttribute` class. This attribute is used to ignore the element when calculating coverage since untestable.

`Errors`

```
using ConsoleAppVisuals.Errors;
```

This namespace contains all the custom exceptions of the library. They are used to handle specific errors that may occur during the execution of the library.

Bonus: `GlobalUsings.cs`

This file contains the different usings of the library. It is used to import the different classes of the library and enable them globally in the library. I recommend you to do the same in your projects. See file

Structure

Here is the detailed file structure of the library:

```
ConsoleAppVisuals
├───elements
│   ├───animated
│   │   ├───FakeLoadingBar.cs
│   │   └───LoadingBar.cs
│   ├───interactive
│   │   ├───FloatSelector.cs
│   │   ├───IntSelector.cs
│   │   ├───Prompt.cs
│   │   ├───Dialog.cs
│   │   ├───ScrollingMenu.cs
│   │   └───TableSelector.cs
│   └───passive
│       ├───inspectors
│       │   ├───ElementsList.cs
│       │   └───ElementsDashboard.cs
│       ├───EmbedText.cs
│       ├───Text.cs
│       ├───Matrix.cs
│       ├───TableView.cs
│       ├───Banner.cs
│       ├───Header.cs
│       ├───Footer.cs
│       └───Title.cs
├───attributes
│   └───VisualAttribute.cs
├───enums
│   ├───BordersType.cs
│   ├───DialogOption.cs
│   ├───Direction.cs
│   ├───ElementType.cs
│   ├───Font.cs
│   ├───Placement.cs
│   ├───PromptInputStyle.cs
│   ├───Status.cs
│   └───TextAlignment.cs
├───errors
│   ├───EmptyFileException.cs
│   ├───LineOutOfConsoleException.cs
│   ├───DuplicateElementException.cs
│   ├───ElementNotFoundException.cs
│   └───NotSupportedCharException.cs
├───models
│   ├───Element.cs
│   ├───AnimatedElement.cs
│   ├───PassiveElement.cs
│   ├───Borders.cs
│   ├───InteractiveElement.cs
│   ├───InteractionEventArgs.cs
│   ├───Position.cs
│   ├───TextStyler.cs
│   └───FontYamlFile.cs
├───Core.cs
├───Window.cs
└───GlobalUsings.cs
```

Have a question, give a feedback or found a bug? Feel free to open an issue or start a discussion on the GitHub repository.

Table of content

Introduction

In this section, you will find internal and external example projects that use the ConsoleAppVisuals library. Feel free to explore them as you like and use them as a reference for your own projects.

Each one will guide you through the setup and execution of the project, as well as provide a brief explanation of the purpose of the project.

Contributing

If you use the ConsoleAppVisuals library in your project and would like to share it with the community, feel free to reach us at morgan@kodelab.fr.

Before you do, make sure to do the following:

- Read the contribution guidelines
- Use the latest or recent version of the library
- Ensure that your code is clean and understandable (documentation is not mandatory but appreciated)
- Capture a GIF or Snapshot demo of your project
- Add a license to your project
- [OPTIONAL]: Follow the template provided here.

Have a question, give a feedback or found a bug? Feel free to open an issue or start a discussion on the GitHub repository.

Table of content

On this section, you will find articles related to the use of the ConsoleAppVisuals library and general articles for C# projects.

Regarding the library,

- Create your own visual element
- Create and use font
- Debugging using the library

Regarding C# projects,

- Create your project documentation
- Publish a library
- C# for Visual Studio Code

Have a question, give a feedback or found a bug? Feel free to open an issue or start a discussion on the GitHub repository.

Getting started

CAUTION

This part of the documentation exists for legacy purposes. All 3.x.x versions and above are not compatible with the instructions on this section.

Outline

Welcome to the docs, here you will find all the information you need to use this library.

You will find the following topics about the library:

- Homescreen visuals
- Write on the console
- Specific methods

And finally, you will find the precise documentation in "References" section.

NOTE

Feel free to contribute to the project by forking it and making a pull request or open an issue if you encounter a bug.

Structure

The library is composed of 4 main classes:

```
ConsoleAppVisuals
├───models
│   ├───Position.cs
│   ├───Placement.cs
│   └───FontYamlFile.cs
├───Core.cs
├───Extensions.cs
├───Matrix.cs
├───Table.cs
├───TextStyler.cs
└───Usings.cs
```

Usings.cs

This file contains the different usings of the library. It is used to import the different classes of the library and enable them globally in the library.

Core.cs

This class is the core of the library. It contains the methods to display the different visuals and variables.

Extensions.cs

This class contains different extensions methods for strings and tuples.

With Position.cs and Placement.cs, it belongs to the tools classes.

TextStyler.cs

This class is used to style the text. It contains the methods to apply a specific style to a text. Often used for the title. It may be useful to create your own style.

Table.cs

This class is used to create a table. It may be useful to display data in a table on the screen.

Matrix.cs

This class is used to create a matrix. It may be useful to display data in a matrix on the screen.

Position.cs

This class is used to define any position defined by an X and Y coordinate. It may be used in cases like matrix selectors for example.

Placement.cs

This class is used to define the placement of a text in the console. It may be useful to indicate where to place a text in a console, or to define the position of a text in a larger string.

FontYamlFile.cs

This class is used to define a font from a yaml file. It may be useful to create your own font.