# GameButlerWindowsForms.Net6

Using Visual Studio Community 2022

Version 17.7.6

Download from:

https://learn.microsoft.com/en-us/visualstudio/releases/2022/release-notes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WHILE INSTALLING SELECT:

ASP.NET and WEB development

.NET multi-platform app UI development

.NET Desktop Development

Universal Windows Platform development (may not actually be needed)

You have to install "Class Designer" under individual components, in order to view and change the designer files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* How to Run

Download the project click on Solution (.sln) file and run. It will compile the code automatically.

We are planning on creating a class library written in C# to interface with the API’s. We will be coding in C# and using the framework .NET 6, a cross-platform coding standard that allows for multiple types of architectures and frameworks. With one of the Web-based frameworks, ASP.NET, Multiple architectural styles have been Included, like Model View Controller (MVC), which is actually an MVVM architecture, Server-only processing, and Client-Server processing. These are all renditions layer-based and component-driven architectures.

The two Architectural styles we will choose will both utilize the exact same class library for the API calls, which is layered and component-based, but also utilizes the architecture of the internet with cookies, HTTP GET, and POST commands via REST API style architecture. The Inputs of this layer will be Login/ Password information (encrypted) and method calls, and the output of this will be essentially a shared memory architecture, and Boolean variables indicating success.

The difference in the two different implementations of the program is that they will completely utilize different front-end implementation frameworks.

One will use a component-based layered architectural style done in Windows Forms, a native windows application, but with a side-loaded HTML viewer component called WebView2, which is a .net framework plugin based on Microsoft Edge. The main layers at the low-level implementation will be: a presentation layer, interface layer, a possible business layer (not currently utilized, but may be needed to meet uncommitted goals), and a link to the data-access layer. This final DAL layer will be in the Class library used in both architectures.

In this implementation there will be multiple presentation layers: one for authentication, one for login, and one for presentation of the game information itself. Behind each presentation layer item there will be an interface layer for the action routing of that specific Windows Forms based view. The uncommitted, but planned part of this application will include a chat feature with friends, however this implementation may need access to a payment only, C++ based, Partner level API, which I have recently applied for, but have not yet received the acceptance letter for access. This API is generally used in actual video games for speaking directly with steam servers to route messages and invites from friends in the Steam Network.

The other will utilize a complex web-based Client-Server, and Server-Server Architecture, with either REST-based MVC style architecture or purely REST-Based Controller architecture. In the Uncommitted example of this architecture, there will need to be Client processing written in C# but presented to the client as JavaScript code in order to hash and properly secure passwords for transmission across the internet to the Server, and then to the REST based API. But, because this code will be only transmitting locally to the server, this is an extended goal for hopes of actual deployment to a real web service.

This will have a similarly based GUI architecture, but follow a REST-like structure to interact with the other layers. This part of the implementation

We will refrain from using external packages, and not use any third-party frameworks or libraries to accomplish these goals. But, Because the C# language heavily relies on and strongly suggests coding standards which require the use of class structures, we will be coding the layers as classes but just as a nominal and organizational relation, not as a functional requirement. Some low-level objects that absolutely require class structure, will also be object-oriented, to organize them for data binding to UI elements.

We chose an event-based gui because it is perfect for GUI driven applications, and we chose to try out a client server based architecture because that architecture is perfect for web applications, which this is technically a combination of both, so we wanted to see which architecture would be better.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONFIGURATION AS COMPILED:

Microsoft Visual Studio Community 2022

Version 17.7.6

VisualStudio.17.Release/17.7.6+34221.43

Microsoft .NET Framework

Version 4.8.09032

Installed Version: Community

Visual C++ 2022 00482-90000-00000-AA678

Microsoft Visual C++ 2022

ASP.NET and Web Tools 17.7.274.37181

ASP.NET and Web Tools

Azure App Service Tools v3.0.0 17.7.274.37181

Azure App Service Tools v3.0.0

Azure Functions and Web Jobs Tools 17.7.274.37181

Azure Functions and Web Jobs Tools

C# Tools 4.7.0-3.23517.17+9d4cc0304792762c34f41cbbec67a1fcd2dc764f

C# components used in the IDE. Depending on your project type and settings, a different version of the compiler may be used.

Common Azure Tools 1.10

Provides common services for use by Azure Mobile Services and Microsoft Azure Tools.

Cookiecutter 17.0.23189.3

Provides tools for finding, instantiating and customizing templates in cookiecutter format.

Extensibility Message Bus 1.4.34 (main@d5ab18b)

Provides common messaging-based MEF services for loosely coupled Visual Studio extension components communication and integration.

JsonViewer 1.0

JSON viewer

Linux Core Dump Debugging 1.0.9.34221

Enables debugging of Linux core dumps.

Mono Debugging for Visual Studio 17.7.27 (547ea6f)

Support for debugging Mono processes with Visual Studio.

NuGet Package Manager 6.7.0

NuGet Package Manager in Visual Studio. For more information about NuGet, visit https://docs.nuget.org/

Razor (ASP.NET Core) 17.7.3.2333001+0ab18affdf2a37647768d0e25f5f021bee6257a1

Provides languages services for ASP.NET Core Razor.

SQL Server Data Tools 17.7.12.0

Microsoft SQL Server Data Tools

Test Adapter for Boost.Test 1.0

Enables Visual Studio's testing tools with unit tests written for Boost.Test. The use terms and Third Party Notices are available in the extension installation directory.

Test Adapter for Google Test 1.0

Enables Visual Studio's testing tools with unit tests written for Google Test. The use terms and Third Party Notices are available in the extension installation directory.

TypeScript Tools 17.0.20829.2001

TypeScript Tools for Microsoft Visual Studio

Visual Basic Tools 4.7.0-3.23517.17+9d4cc0304792762c34f41cbbec67a1fcd2dc764f

Visual Basic components used in the IDE. Depending on your project type and settings, a different version of the compiler may be used.

Visual C++ for Cross Platform Mobile Development (Android) 17.0.33906.96

Visual C++ for Cross Platform Mobile Development (Android)

Visual C++ for Linux Development 1.0.9.34221

Visual C++ for Linux Development

Visual Studio IntelliCode 2.2

AI-assisted development for Visual Studio.

VisualStudio.DeviceLog 1.0

Information about my package

VisualStudio.Mac 1.0

Mac Extension for Visual Studio

VSPackage Extension 1.0

VSPackage Visual Studio Extension Detailed Info

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOT NEEDED BELOW BUT WAS INSTALLED ON MY COMPUTER \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Microsoft JVM Debugger 1.0

Provides support for connecting the Visual Studio debugger to JDWP compatible Java Virtual Machines

Python - Profiling support 17.0.23189.3

Profiling support for Python projects.

Python with Pylance 17.0.23189.3

Provides IntelliSense, projects, templates, debugging, interactive windows, and other support for Python developers.

Visual Studio Tools for Unity 17.7.0.0

Visual Studio Tools for Unity

Visual F# Tools 17.7.0-beta.23314.10+e612cf93b989503c89e3a5830090062b7ab5e143

Microsoft Visual F# Tools

Xamarin 17.7.0.223 (d17-7@c374b3e)

Visual Studio extension to enable development for Xamarin.iOS and Xamarin.Android.

Xamarin Designer 17.7.3.9 (remotes/origin/d17-7@796d191def)

Visual Studio extension to enable Xamarin Designer tools in Visual Studio.

Xamarin Templates 17.7.21 (150ec9f)

Templates for building iOS, Android, and Windows apps with Xamarin and Xamarin.Forms.

Xamarin.Android SDK 13.2.1.2 (d17-5/a8a26c7)

Xamarin.Android Reference Assemblies and MSBuild support.

Mono: d9a6e87

Java.Interop: xamarin/java.interop/d17-5@149d70fe

SQLite: xamarin/sqlite/3.40.1@68c69d8

Xamarin.Android Tools: xamarin/xamarin-android-tools/d17-5@ca1552d

Xamarin.iOS and Xamarin.Mac SDK 16.4.0.18 (9d266025e)

Xamarin.iOS and Xamarin.Mac Reference Assemblies and MSBuild support.