🏗️ Jenga Build System

Modern Multi-Platform C/C++ Build System with Unified Python DSL

License Proprietary

Python 3.7+

Platforms Windows | Linux | macOS | Android | iOS | WebAssembly

📋 Table of Contents

✨ Features

🎯 Core Capabilities

- **Unified Python DSL** - Clean, readable configuration syntax
- **Multi-Platform Support** - Windows, Linux, macOS, Android, iOS, WebAssembly
- **Intelligent Cache** - 20x faster incremental builds
- **Integrated Testing** - Built-in Unitest framework
- **Zero Dependencies** - Pure Python 3, no external tools required

🔧 Build System

- **C/C++ Toolchains** - GCC, Clang, MSVC support
- **Cross-Compilation** - Android NDK, Emscripten
- **Parallel Builds** - Multi-core optimization
- **Dependency Graph** - Automatic build ordering
- **Smart File Tracking** - Changed files detection

📦 Packaging & Distribution

- **APK/AAB Generation** - Android packages
- **IPA Creation** - iOS App Store packages
- **Desktop Packages** - ZIP, DMG, AppImage
- **Code Signing** - Android, iOS, Windows, macOS

- **CI/CD Ready** - Automated pipeline integration

🧪 Testing Framework

- **Auto Test Discovery** - No manual test registration
- **Parallel Test Execution** - Multi-threaded testing
- **Elegant Reporting** - Colorful console output
- **Performance Benchmarks** - Integrated profiling
- **Code Coverage** - Built-in coverage tools

🚀 Quick Start

Hello World in 60 Seconds

1. **Create project structure:**

```
mkdir hello-world
cd hello-world
```

2. **Create `main.cpp`:**

```
#include <iostream>

int main() {
    std::cout << "Hello, Jenga!" << std::endl;
    return 0;
}
```

3. **Create `hello.jenga`:**

```
with workspace("HelloWorld"):
    configurations(["Debug", "Release"])

    with project("Hello"):
        consoleapp()
        language("C++")
        files(["main.cpp"])
        targetdir("Build/Bin/%{cfg.buildcfg}")
```

4. **Build and run:**

```
jenga build
jenga run
# Output: Hello, Jenga!
```

Installation

Method 1: From PyPI (Recommended)

```
pip install jenga-build-system
```

Method 2: From Source

```
# Clone repository
git clone https://github.com/RihenUniverse/Jenga.git
cd Jenga

# Install in development mode
pip install -e .

# Or install globally
pip install .
```

Method 3: Manual Installation

1. Download the latest release
2. Extract to your preferred location
3. Add to PATH:
    - **Windows**: Add folder to system PATH
    - **Linux/macOS**: `export PATH="$PATH:/path/to/jenga"`

Script Availability

After installation, both `.sh` and `.bat` scripts are available:

- **Linux/macOS**: `jenga` command globally available
- **Windows**: `jenga.exe` or `jenga` command available
- **All platforms**: `python -m Jenga.jenga` also works

Basic Usage

Project Configuration

```
with workspace("MyApplication"):
    # Global settings
    configurations(["Debug", "Release", "Dist"])
    platforms(["Windows", "Linux", "Android"])
    startproject("MainApp")
    # Compiler toolchain
    with toolchain("gcc", "g++"):
        cppcompiler("g++")
        cppdialect("C++20")

    # Library project
    with project("CoreLibrary"):
        staticlib()
        files(["src/core/**.cpp", "include/**.h"])
        includedirs(["include"])

    # Application project
    with project("MainApp"):
        consoleapp()
        files(["src/app/**.cpp"])
        dependson(["CoreLibrary"])

        # Unit tests
        with test("Unit"):
            testfiles(["tests/**.cpp"])
```

Common Commands

```
# Build default project
jenga build

# Build specific configuration
jenga build --config Release --platform Windows

# Run application
jenga run
jenga run --project MyApp

# Clean build artifacts
jenga clean
jenga clean --all

# Show project info
jenga info

# Generate project files (VS, Xcode, etc.)
jenga gen

# Package for distribution
jenga package --platform Android --config Release

# Code signing
jenga keygen --platform Android
jenga sign --platform Android
```

```
# Run tests
jenga run --project MyProject_Unit_Tests
```

🏗 Project Structure

```
jenga-project/
├── .jenga/                    # Build cache (auto-generated)
├── Build/                     # Build artifacts
│   ├── Bin/                  # Executables
│   ├── Lib/                  # Libraries
│   └── Obj/                  # Object files
├── src/                       # Source code
│   ├── core/                 # Core functionality
│   ├── platform/             # Platform-specific code
│   └── main.cpp
├── tests/                     # Test files
├── third_party/              # External dependencies
└── myproject.jenga           # Jenga configuration
```

Multi-Platform Structure

```
src/
├── core/                     # Platform-independent
│   ├── math.cpp
│   └── graphics.cpp
└── platform/
    ├── windows/             # Windows-specific
    │   └── window_win32.cpp
    ├── linux/               # Linux-specific
    │   └── window_x11.cpp
    └── android/             # Android-specific
        └── window_android.cpp
```

📚 Documentation

Complete Documentation

All documentation is included in the `Docs/` directory:

Document	Description
📖 BOOK_PART_1.md	Introduction & Installation
📖 BOOK_PART_2.md	Core Concepts
📖 BOOK_PART_3.md	Advanced Features

Document	Description
🔧 QUICKSTART.md	Quick Start Guide
📖 API_REFERENCE.md	Complete API Reference
🤖 ANDROID_EMSCRIPTEN_GUIDE.md	Android & WebAssembly
🪟 MSVC_GUIDE.md	Windows/Visual Studio Guide
🧪 TESTING_GUIDE.md	Testing Framework
📦 PACKAGING_SIGNING_GUIDE.md	Packaging & Signing
🔄 MIGRATION_GUIDE.md	Migration from CMake/Make
🔍 TROUBLESHOOTING.md	Troubleshooting Guide
📋 CHANGELOG.md	Version History

Online Resources

- **GitHub Repository**: https://github.com/RihenUniverse/Jenga
- **Issue Tracker**: https://github.com/RihenUniverse/Jenga/issues
- **Discussion Forum**: GitHub Discussions

🔧 Advanced Features

Multi-Platform Configuration

```python
with workspace("CrossPlatformGame"):
    platforms(["Windows", "Linux", "Android", "iOS"])

    with project("GameEngine"):
        staticlib()

        # Common code
        files(["src/engine/**.cpp"])

        # Platform-specific
        with filter("system:Windows"):
            links(["d3d11", "dxgi"])

        with filter("system:Android"):
            androidminsdk(21)
            links(["log", "android", "EGL"])

        with filter("system:iOS"):
            framework("UIKit")
            framework("OpenGLES")
```

External Project Inclusion

```
with workspace("MyProject"):
    # Include external libraries
    include("external/mathlib/math.jenga")
    include("third_party/logger.jenga", ["Logger"])

    with project("App"):
        consoleapp()
        dependson(["MathLib", "Logger"])
```

Android APK Generation

```
with workspace("AndroidApp"):
    platforms(["Android"])

    with project("App"):
        sharedlib()
        androidapplicationid("com.company.app")
        androidversioncode(1)
        androidversionname("1.0.0")

        # Assets and resources
        dependfiles([
            "assets/**",
            "res/**",
            "AndroidManifest.xml"
        ])
```

iOS IPA Generation

```
with workspace("iOSApp"):
    platforms(["iOS"])

    with project("App"):
        consoleapp()
        # iOS-specific settings
        # TODO: Add iOS configuration API
```

📁 Project Examples

Example 1: Simple Library

```
simple-lib/
├── mathlib.jenga
├── include/
│   └── math.h
└── src/
    └── math.cpp
```

mathlib.jenga:

```
with workspace("MathLib"):
    with project("Math"):
        staticlib()
        files(["src/**.cpp"])
        includedirs(["include"])
        targetdir("Build/Lib/%{cfg.buildcfg}")
```

Example 2: Game Engine

```
game-engine/
├── engine.jenga
├── Core/          # Math, Physics, etc.
├── Rendering/     # OpenGL, Vulkan
├── Audio/         # Sound system
└── Game/          # Game logic
```

Example 3: Mobile App

```
mobile-app/
├── app.jenga
├── native/        # C++ core
├── android/       # Android-specific
├── ios/          # iOS-specific
└── shared/        # Common assets
```

🤝 Contributing

We welcome contributions! Here's how you can help:

Reporting Issues

1. Check existing issues in GitHub
2. Use the issue template
3. Include system info and reproduction steps

Feature Requests

1. Describe the use case
2. Show example syntax
3. Discuss implementation

Code Contributions

1. Fork the repository
2. Create a feature branch
3. Write tests for new features
4. Submit a pull request

Development Setup

```
# Clone and setup
git clone https://github.com/RihenUniverse/Jenga.git
cd Jenga
pip install -e .[dev]

# Run tests
pytest

# Format code
black .

# Check code quality
flake8 Jenga/
mypy Jenga/
```

📄 License

Proprietary License - Rihen

Copyright © 2026 Rihen. All rights reserved.

Permissions

☑ **Free to Use** - No cost for personal or commercial use
☑ **Modification Rights** - You may modify the source code
☑ **Distribution** - You may distribute modified versions
☑ **Integration** - Can be used in proprietary projects

Conditions

1. **Attribution Required** - Must include this license in distributions
2. **Copyright Notice** - Must preserve Rihen copyright
3. **No Removal** - Cannot remove license headers from source files

4. **No Sublicensing** - Cannot grant additional rights to others

Restrictions

❌ **No Resale** - Cannot sell Jenga as a standalone product
❌ **No Warranty** - Provided "as is" without guarantees
❌ **Liability** - Rihen not liable for damages
❌ **Patent Claims** - No patent licenses granted

License Text

```
Jenga Build System
Copyright (c) 2026 Rihen

This software is provided under the Rihen Proprietary License.
You may use, modify, and distribute this software for any purpose,
provided that you include this license and copyright notice in all
copies or substantial portions of the software.

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND.
Rihen SHALL NOT BE LIABLE FOR ANY DAMAGES ARISING FROM
THE USE OF THIS SOFTWARE.

For complete terms, see the LICENSE file included with this distribution.
```

For Your Projects

Include this notice in your project's documentation:

```
This project uses Jenga Build System
Licensed under Rihen Proprietary License
Copyright © 2026 Rihen
```

⚖️ Disclaimer

Important Legal Notice

NO WARRANTY: Jenga Build System is provided "AS IS" without any warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

NO LIABILITY: In no event shall Rihen or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

USER RESPONSIBILITY: You are solely responsible for:

1. Testing the software in your environment
2. Ensuring compliance with applicable laws
3. Backing up your data before use
4. Verifying build results and outputs

THIRD-PARTY COMPONENTS: Jenga may interact with third-party tools (compilers, SDKs, etc.). Rihen is not responsible for:

- Availability or compatibility of third-party tools
- Licensing requirements of third-party components
- Issues arising from third-party tool changes

EXPORT COMPLIANCE: You are responsible for complying with all applicable export control laws and regulations.

INDUSTRY STANDARDS: While Jenga follows industry best practices, you should:

- Review generated build files before production use
- Perform security audits for sensitive applications
- Maintain your own quality assurance processes

By using Jenga Build System, you acknowledge that you have read this disclaimer, understand it, and agree to be bound by its terms.

📞 Support & Contact

- **GitHub Issues**: https://github.com/RihenUniverse/Jenga/issues
- **Email**: rihen.universe@gmail.com
- **Documentation**: https://github.com/RihenUniverse/Jenga/tree/main/Docs

---

Built with 💝 by Rihen

Jenga Build System - Making C++ builds simple across all platforms

---

📄 Fichier de licence (LICENSE) :

```
PROPRIETARY LICENSE AGREEMENT
For Jenga Build System
Copyright © 2026 Rihen

IMPORTANT: READ CAREFULLY

This Proprietary License Agreement ("Agreement") is a legal agreement between
you
(either an individual or a single entity) and Rihen for the Jenga Build
```

```
System software product ("Software").

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU AGREE TO BE BOUND
BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS
AGREEMENT, DO NOT INSTALL OR USE THE SOFTWARE.

1. GRANT OF LICENSE
Rihen grants you a non-exclusive, worldwide, royalty-free license to:
   a) Use the Software for any purpose, personal or commercial
   b) Modify the source code to create derivative works
   c) Distribute the Software or derivative works

   CONDITIONS:
   - You must include this license in all distributions
   - You must preserve all copyright notices
   - You cannot remove license headers from source files

2. RESTRICTIONS
You may not:
   a) Sell the Software as a standalone product
   b) Grant sublicenses to others
   c) Use the Rihen name for endorsement without permission
   d) Remove or alter any proprietary notices

3. COPYRIGHT
All title and copyrights in and to the Software are owned by Rihen.
This license does not transfer ownership of the copyright.

4. NO WARRANTY
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
Rihen OR ITS CONTRIBUTORS 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.

5. TERMINATION
This license is effective until terminated. Your rights under this license will
terminate automatically without notice from Rihen if you fail to comply
with any term(s) of this license.

6. GOVERNING LAW
This Agreement shall be governed by the laws of the jurisdiction where
Rihen is established, without regard to its conflict of law provisions.

7. ENTIRE AGREEMENT
This Agreement constitutes the entire agreement between the parties concerning
the Software and supersedes all prior or contemporaneous oral or written
agreements, proposals, representations, and communications.

8. ACKNOWLEDGMENT
YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE
```

```
TO BE BOUND BY ITS TERMS AND CONDITIONS.

For questions about this license, contact: rihen.universe@gmail.com

Version 1.0 - Effective Date: [Current Date]
```

Réponses à vos questions :

1. **Disponibilité des scripts après installation** :

☑ **Oui, les scripts seront disponibles** :

- Sur Windows : `jenga` ou `jenga.bat` fonctionnera
- Sur Linux/macOS : `jenga` ou `jenga.sh` fonctionnera
- Le `setup.py` configure correctement les entry points
- Alternative : `python -m Jenga.jenga` toujours disponible

2. **Licence propriétaire** :

La licence que j'ai rédigée :

- ☑ Permet l'utilisation gratuite (personnelle et commerciale)
- ☑ Permet les modifications du code source
- ☑ Exige la mention de Rihen
- ☑ Exige l'inclusion de la licence dans les distributions
- ☑ Se dédouane de toute responsabilité (clause "AS IS")
- ☑ Interdit la revente comme produit autonome
- ☑ Protège votre propriété intellectuelle

3. **Structure complète** :

Le README inclut :

- Toutes les fonctionnalités documentées
- Tous les guides de vos fichiers Docs/
- Des exemples pratiques
- Instructions d'installation détaillées
- Section contribution
- Licence complète
- Clause de non-responsabilité légale