

# GladeUnityAI – User Guide

## Table of Contents

1. Overview
2. Getting Started
3. Initial Setup
4. Using the AI Assistant
5. Configuration Options
6. Features
7. Troubleshooting
8. Tips for Best Results
9. Additional Resources

## Overview

GladeUnityAI is an agentic AI assistant that enables direct interaction with Unity through natural language. Describe what you want to build, and the AI will automatically create scripts, manipulate GameObjects, add components, and perform other requested Unity operations.

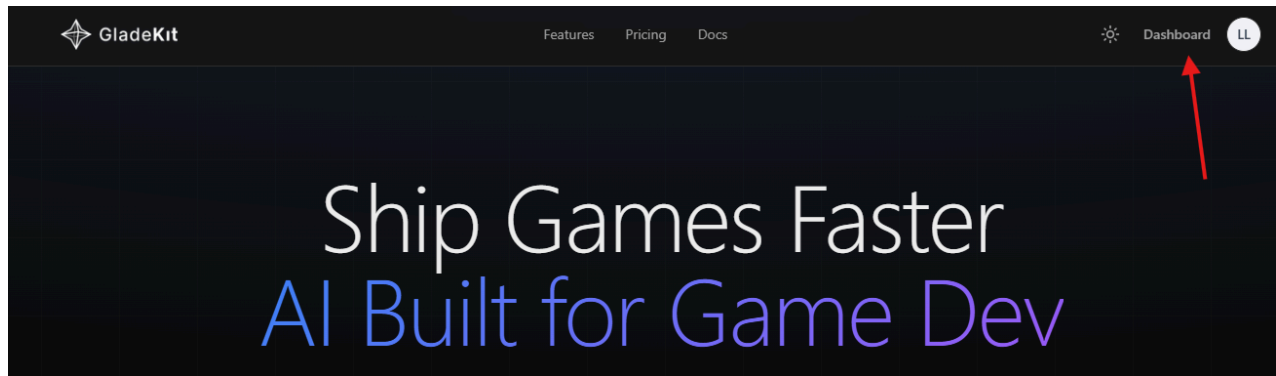
## Key Capabilities

- **Natural Language Interface** – Describe your needs in plain English
- **Automatic Code Generation** – Creates and modifies C# scripts automatically
- **GameObject Manipulation** – Creates, destroys, and modifies GameObjects and components, and in-engine assets for game development needs.
- **Real-Time Streaming** – See AI responses as they're generated
- **Multi-Model Support** – GPT-5 models, GPT-4 models, GPT-3.5 Turbo, Claude 4.5 (Sonnet / Opus / Haiku), Claude 3 (Sonnet / Opus / Haiku)
- **Context-Aware** – Understands your project structure and current scene.

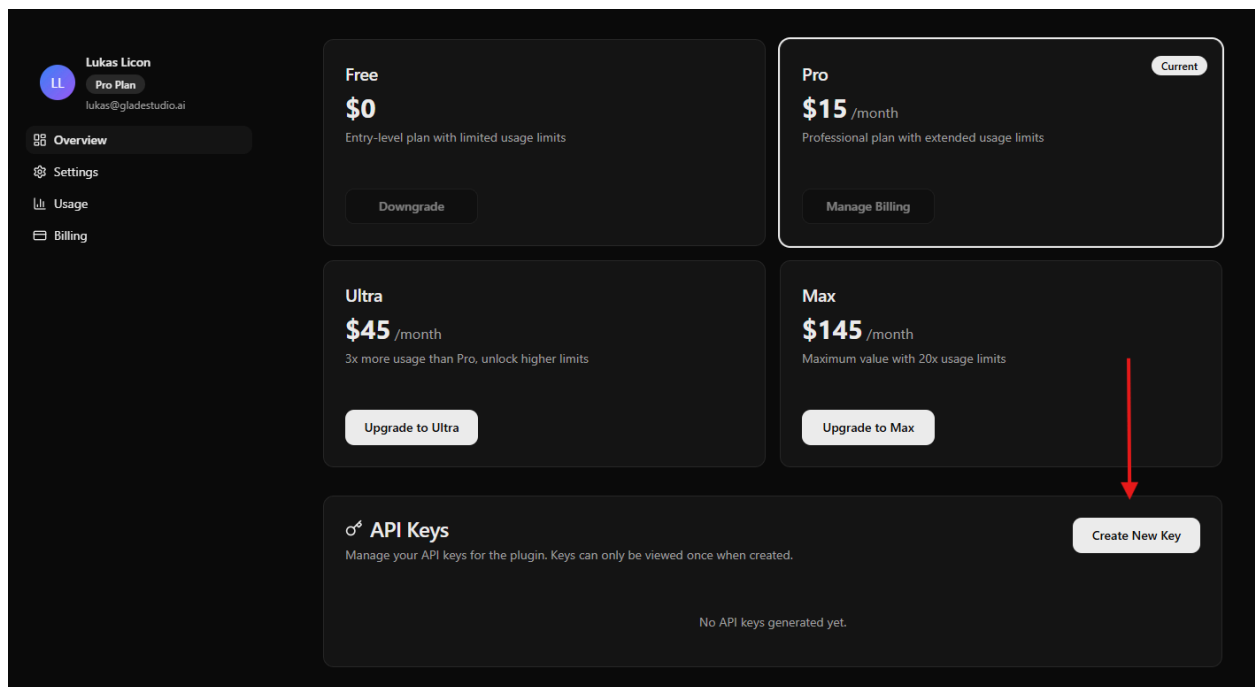
## Before Starting

**Register for a GladeKit account.**

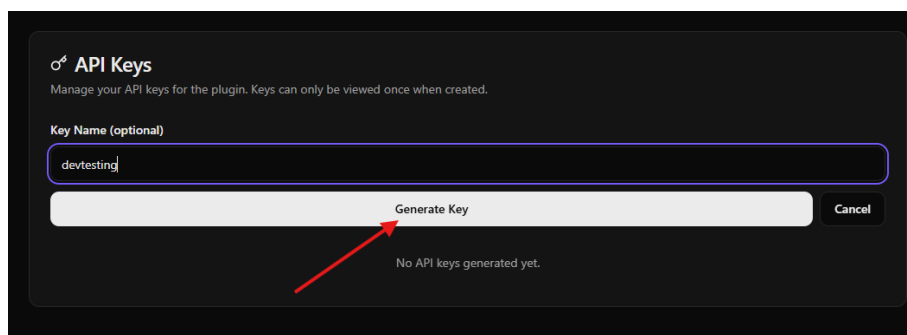
1. Go to [GladeKit.com](https://gladekit.com), click sign up at the top right of the page. If you have an account click sign in and sign in normally.
2. After signing in click dashboard at the top right on the navigation bar.



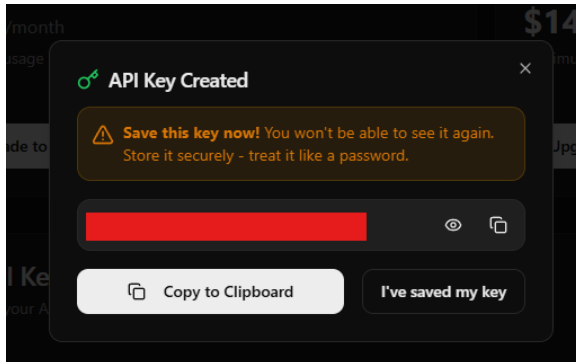
3. Here you can access your plan in overview. At the bottom where API Keys is click **Create New Key** to generate a key to use inside unity.



4. Name the key then click Generate



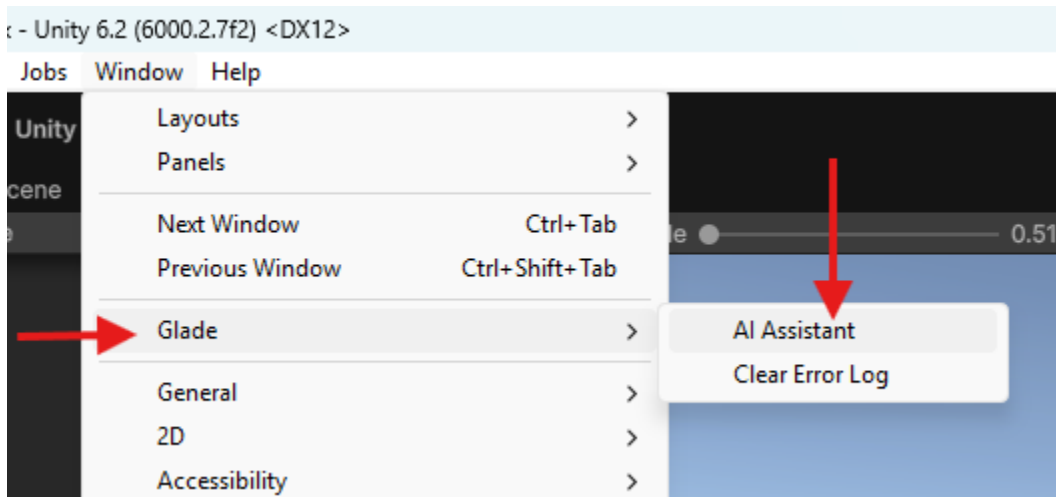
5. Save or copy the key as you won't be able to view it again. But can always generate a new key.



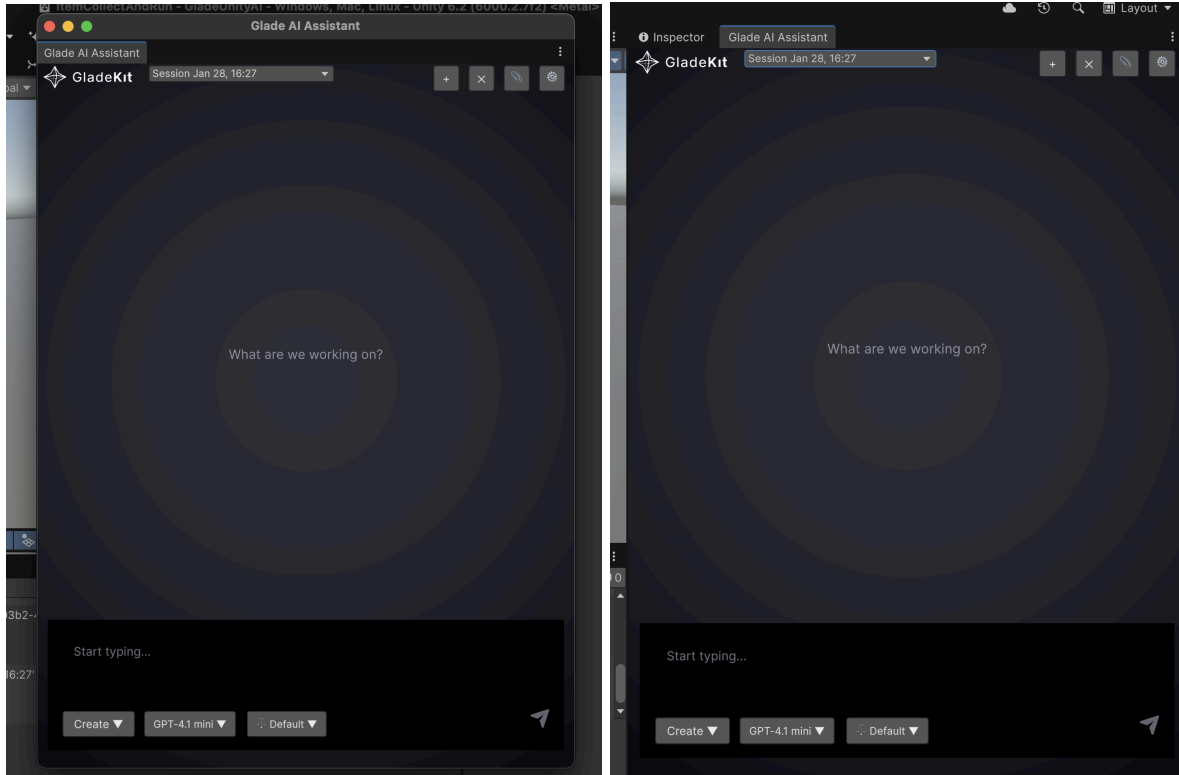
## Getting Started

### Opening the AI Assistant

1. In the Unity Editor, navigate to the menu bar
2. Click **Window** → **Glade** → **AI Assistant**



The AI Assistant window will open inside the Unity Editor and can be docked in the layout.

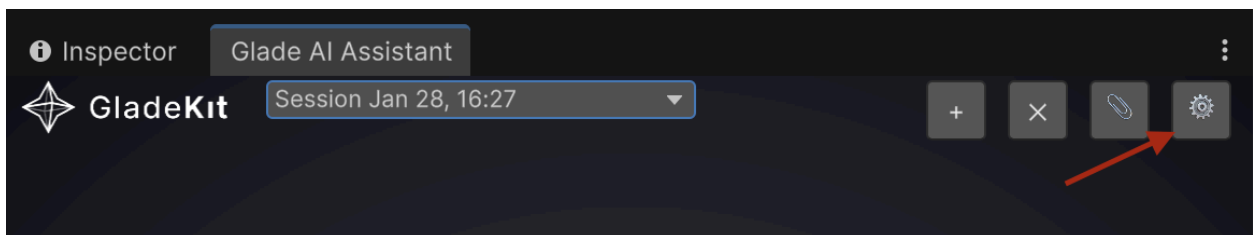


## Initial Setup

### Step 1: Configure Your API Key

Before using the AI Assistant, you must configure your GladeKit API key.

1. Click the **Settings** (⚙️) icon in the top-right corner of the AI Assistant window



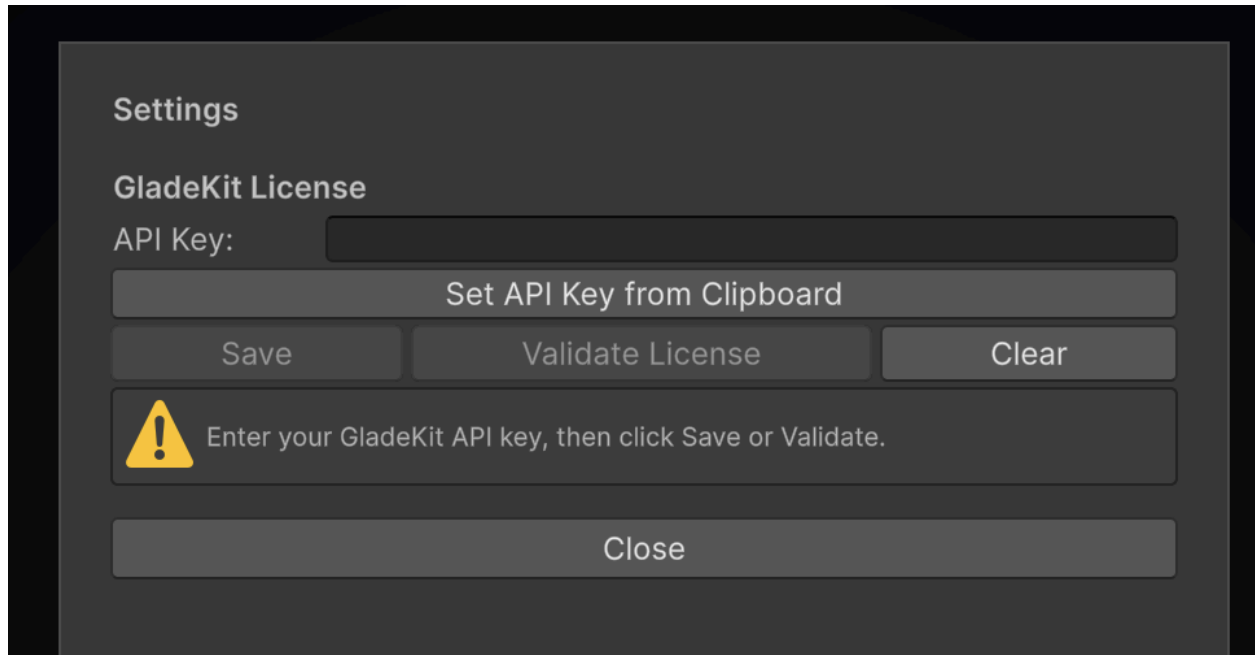
2. The **Settings** popup appears with the **GladeKit License** section
3. Enter your API key using one of the following methods:

#### Option A – Manual Entry

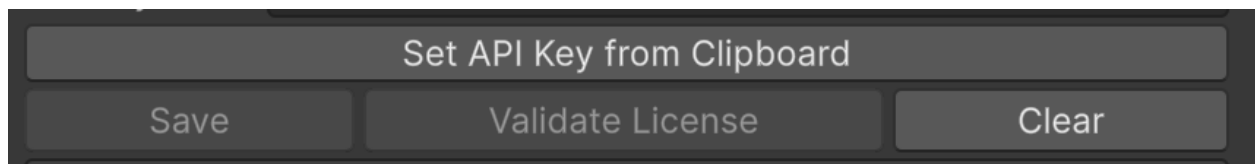
- Click the **API Key** field (password-protected)
- Paste or type your key

## Option B – Clipboard Import

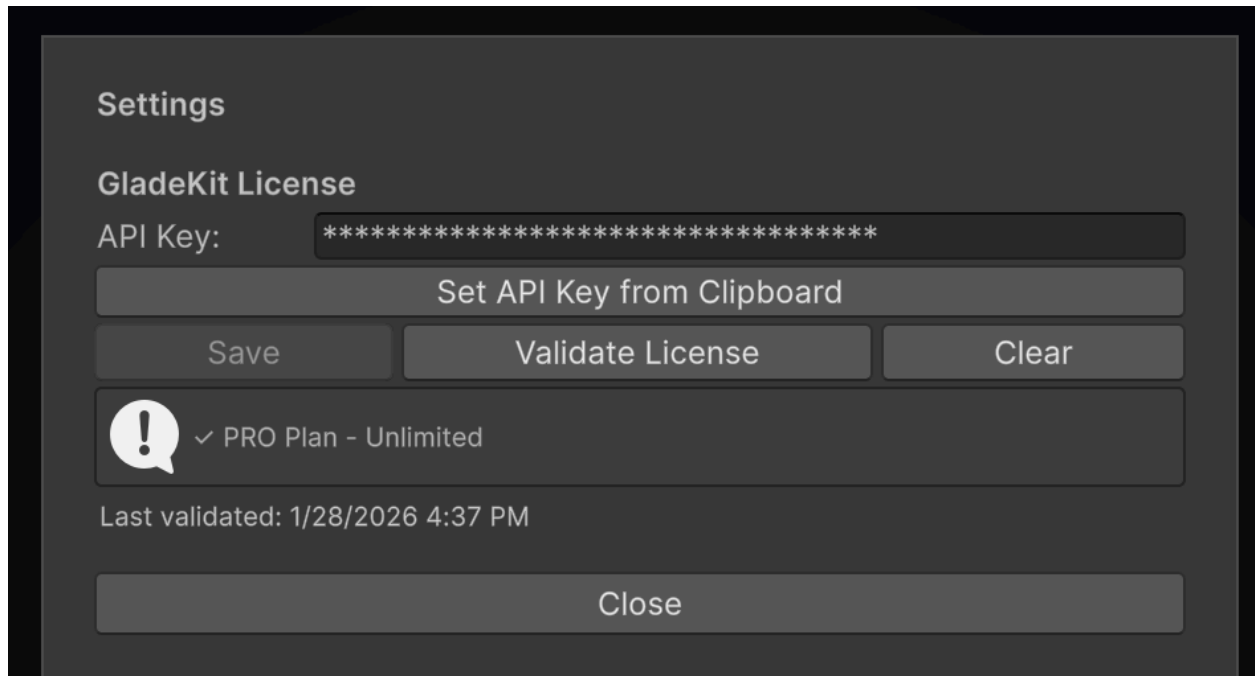
- Copy your API key
- Click **Set API Key from Clipboard**



4. Save or validate your key:
  - **Save** – Stores the key
  - **Validate License** – Stores and verifies in one step



5. Verify license status:
  - **Valid License** – Plan and usage info displayed
  - **Invalid License** – Error message shown



6. Close Settings to return to the main window

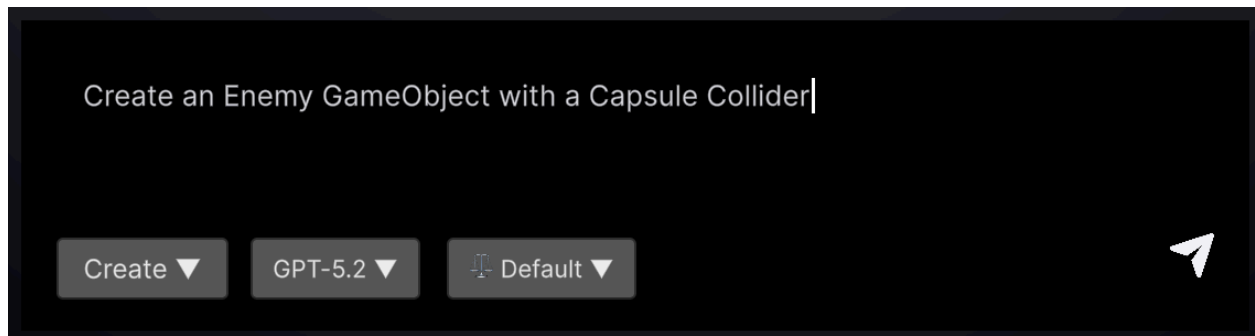
## Using the AI Assistant

### Basic Usage

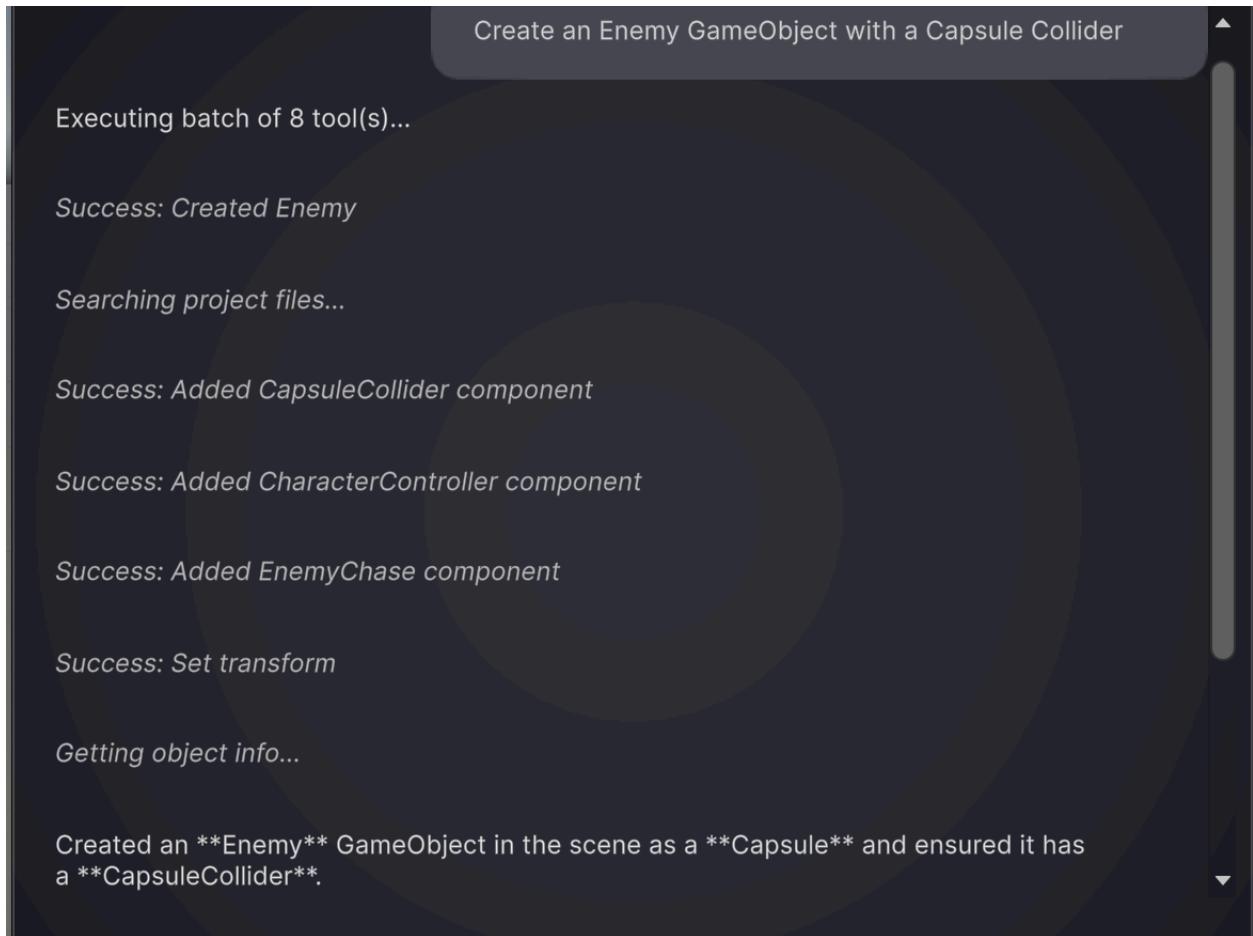
1. Type your request in the input field

Example prompts:

- “Add a Rigidbody to the selected GameObject”
- “Create an Enemy GameObject with a Capsule Collider”



Click **Send** or press **Enter**



## Understanding the Interface

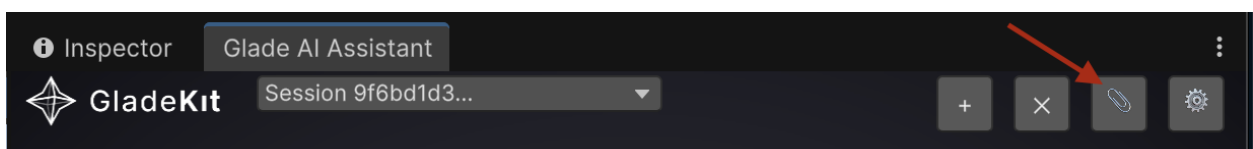
### Chat Area

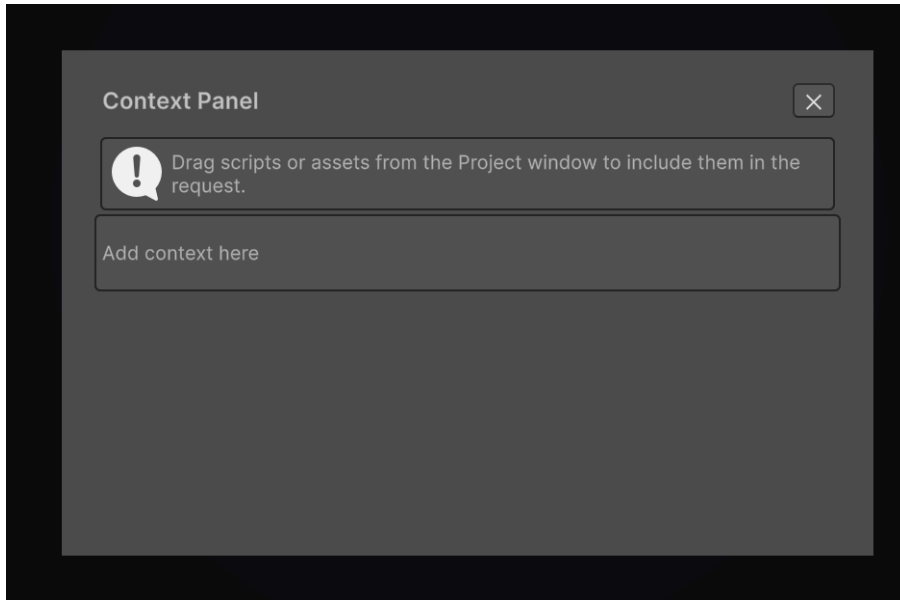
- User messages on the right
- AI messages on the left
- Syntax-highlighted code blocks
- Tool execution results inline

## Adding File Context to Request

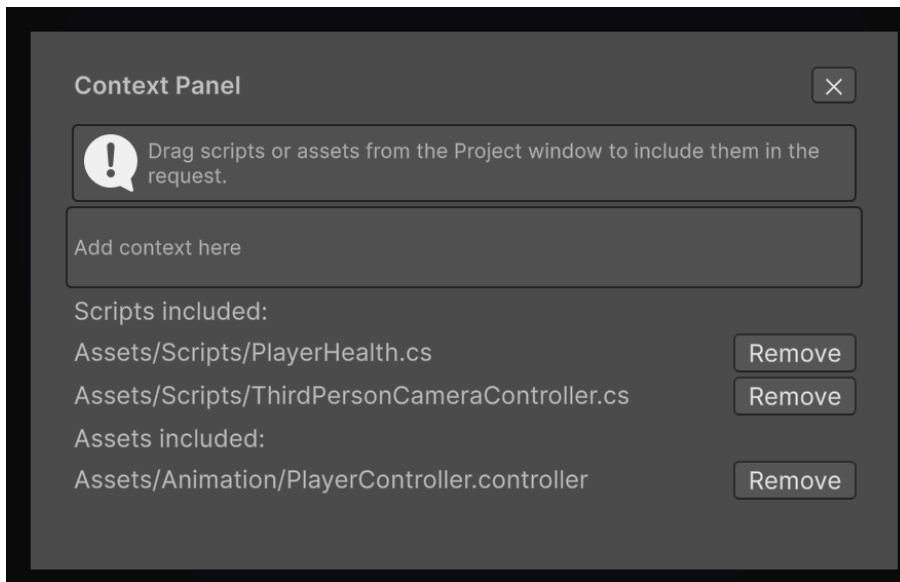
### Context Panel

Click the context panel button in the image below.





Drag files into the area that says “Add context here”



Once files are dragged into the box they will show which files are included in the next request.

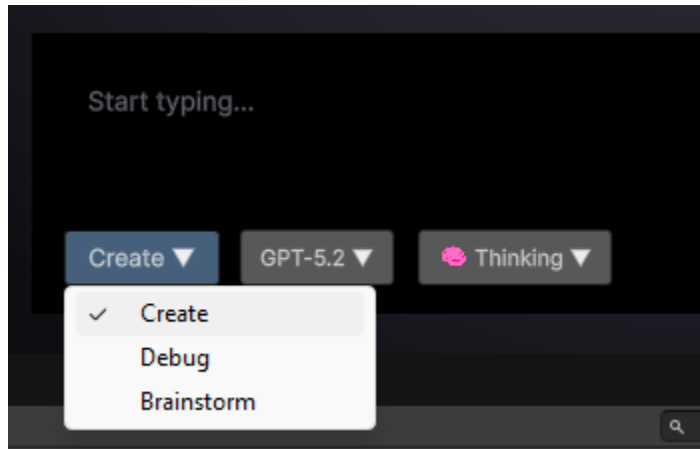
## Configuration Options

### Agent Modes

#### 3 Different Agent modes for GladeKit



1. Create, for adding/creating new objects in unity. Focuses on executing actions based on the user request, context, and message history.
2. Debug, for situations where debugging is needed. Focuses on identifying issues rather than executing actions.
3. Brainstorm, for figuring out new ideas, mechanics, designs, etc. Focuses on chat and creativity rather than executing actions.



## Model Selection

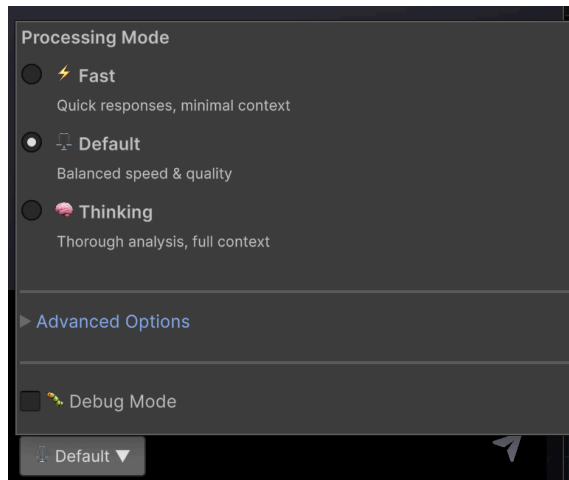


## Processing Modes

**3 Different modes for processing context.**

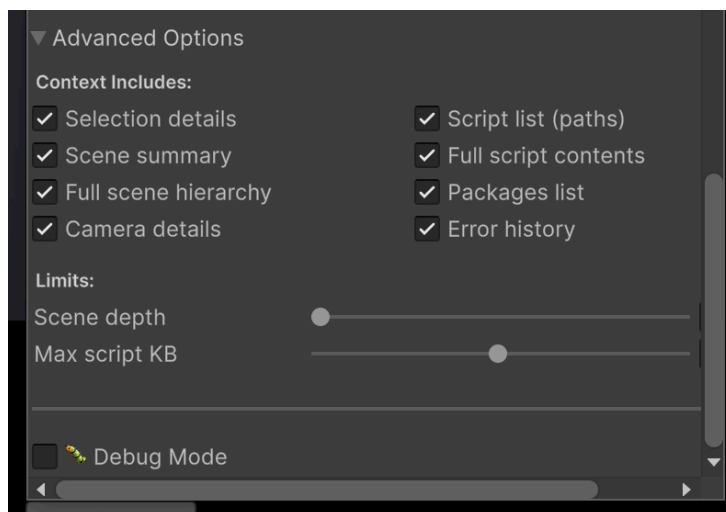
Fast, Default, Thinking, and Custom modes

1. Fast uses the least amount of context possible
2. Default uses a moderate amount of context in each request. Scene hierarchy not included.
3. Thinking uses the full amount of context available. Scene hierarchy included.



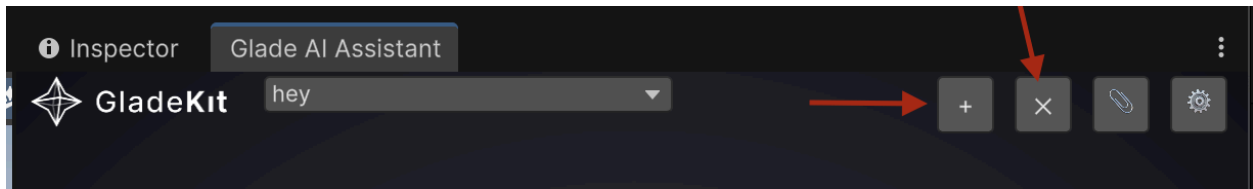
## Advanced Context Options

We recommend to stick to presets but allow users to edit advanced options as well.

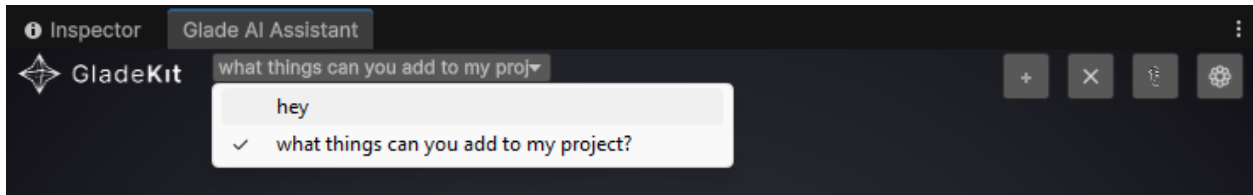


## Session Management

Multiple chat sessions supported through the + button to add new chats and x button to delete current chat.

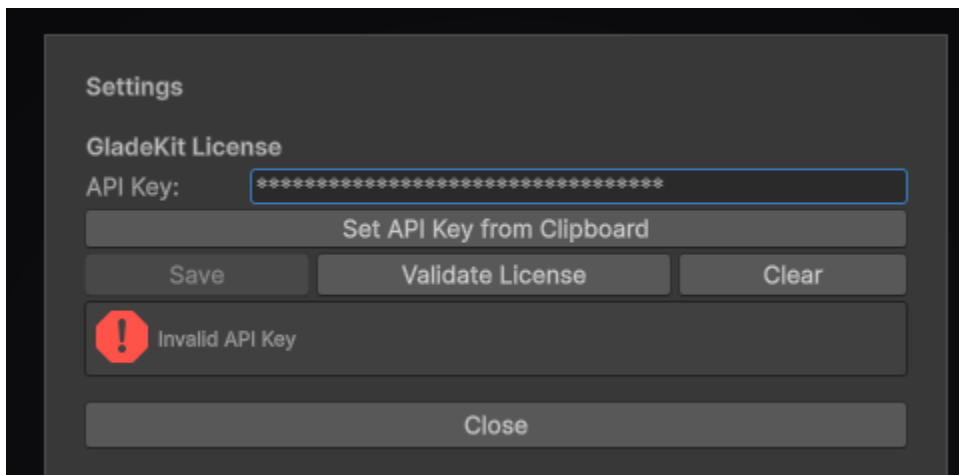


Click the chat dropdown to select which chat you switch between



## Troubleshooting

### API Key Issues



Make sure you have entered the correct API Key. If you continue to get this issue reach out to our team through our discord or email.

### License Limit Reached

Consult [gladekit.com](https://gladekit.com) for pricing and subscription tiers.

## Tips for Best Results

1. Be specific in your requests as details will help the model understand what you need best.

2. Add context to the context panel or reference scripts specifically by name to have the agent look for them.
  3. Minimize requests to optimize results
    - a. Working in one step at a time is better than asking for the entirety of the game at once.
    - b. Moving from system to system is more optimal for the AI Agent overall.
  4. Choose the right model for your development needs. Newer models are more thorough.
  5. Use the appropriate Agent mode to optimize results depending on the scenario.
  6. Use the appropriate processing mode to optimize results
    - a. Fast uses the least amount of context possible. Great for quick requests cheap in token cost that do not need heavy scene context.
    - b. Default uses a moderate amount of context in each request and does not include full scene hierarchy. Not good for debugging.
    - c. Thinking uses the full amount of context available. Great for debugging and all requests but consumes more tokens each message.
- 

## **Additional Resources**

For support, contact Glade directly.