OpenAI Integration for Unity

User guide

Setup

1. Register for OpenAI: openai.com/api
2. Create a new secret key: beta.openai.com/account/api-keys
3. Set your authentication type on the ***DefaultAuthArgsV1*** scriptable object located in ***OpenAi/Runtime/Config/***

   Authentication options:

   a. **Local File -** This authentication method will attempt to find the private key at `***~/.openai/auth.json***` (Linux/Mac) or `***%USERPROFILE%/.openai/auth.json***` (Windows). If this file does not exist or the key is not present, api calls will fail. The JSON file should look like:

   ```
   {
       "private_api_key":"<YOUR API KEY>"
   }
   ```

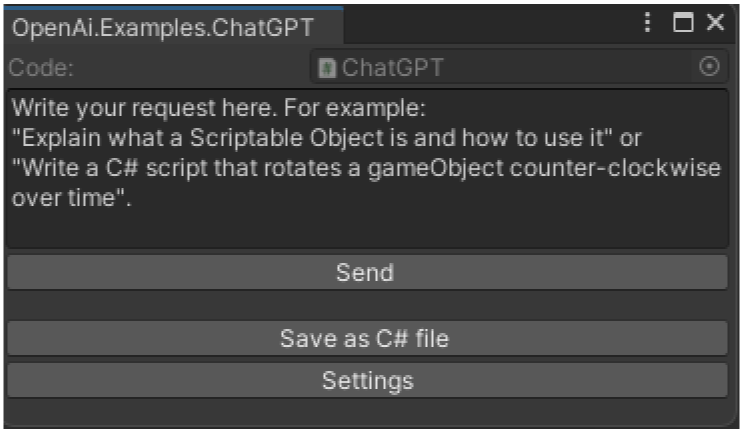
   b. **String -** This option allows you to enter your API key directly, but this method isn't safe if your code is exposed publicly (like on GitHub). **Use this option with caution.**

Recommended: Set usage limits in your OpenAI account to prevent accidentally exceeding your budget.
beta.openai.com/account/billing/limits
For more information on OpenAI pricing: openai.com/api/pricing/

Unity Editor Tools

ChatGTP

This tool functions similarly to OpenAI's own ChatGPT, except you can customize the settings to fine tune the responses you receive. By default, this tool is configured to act as a Senior Unity Game Developer, but you can change this in the settings.



Code: ChatGPT

Write your request here. For example:
"Explain what a Scriptable Object is and how to use it" or
"Write a C# script that rotates a gameObject counter-clockwise over time".

*You can access this tool in Unity by going to **Tools > OpenAI > ChatGPT***

How To Use

Simply enter your request in the text box and then press the send button. If you've chosen for the AI to only return code, you can use the **Save as C# file** button to create a file in your project with the name of the scripts class that was generated in the code.

Examples:

- *How can I create a custom Unity Editor window that has a settings button, and when clicked it reveals 5 more settings?*
- *Write me a Unity C# script that can be used as a loot table, rewarding the player with different items based on an assigned weight.*
- *Help me come up with a backstory for my game. The main characters name should be Cameron and he travels the world collecting elemental powers.*

Settings

- **Model** - The model to use for completion. See beta.openai.com/docs/models for a list of available models.

- **Max tokens** - The maximum number of tokens to generate. Requests can use up to 4000 tokens shared between prompt and completion. (One token is roughly 4 characters for normal English text)

- **Temperature** - Controls randomness: Lowering results in less random completions. As the temperature approaches zero, the model will become deterministic and repetitive.

- **Top P** - Controls diversity via nucleus sampling: 0.5 means half of all likelihood-weighted options are considered.

- **Stop** - Where the API will stop generating further tokens. The returned text will not contain the stop sequence.

- **Frequency penalty** - How much to penalize new tokens based on their existing frequency in the text so far. Decreases the model's likelihood to repeat the same line verbatim.

- **Presence penalty** - How much to penalize new tokens based on whether they appear in the text so far. Increases the model's likelihood to talk about new topics.

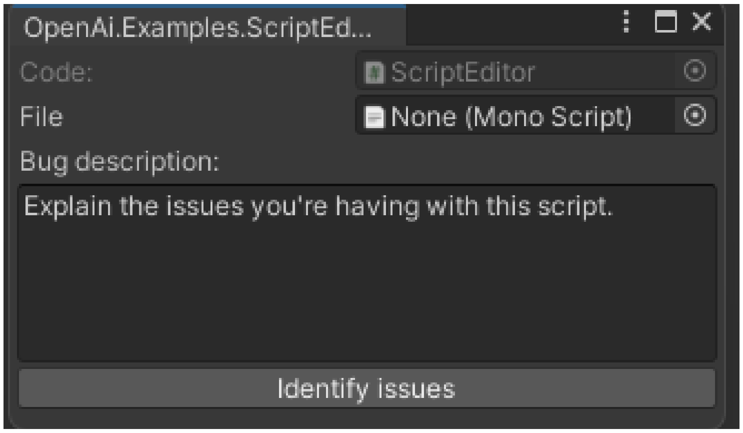
- **Output code only** - Prompts the AI to only respond with code. Use this if you plan on saving the output to a file.
  Note: Selecting this option automatically switches the model to *code-davinci-002* which is currently free in beta and more specialized for writing code.
  - **Add comments to code** - Prompts the AI to add comments to the code it writes to clarify what it's doing.

- **Instruction prompt** - Any text here is added to the beginning of your message to the AI. By default, there is a prompt entered that tells the AI to act as a Senior Unity Game Developer, but you can replace this with anything you'd like.

Tip: Use the ***Prompt Generator*** (found in ***Tools > OpenAI > Prompt Generator***) to create high quality prompts to try.

Script Editor

This tool allows you to analyze any script in your project using AI, which will take into account whatever message you send it and respond back with suggestions. **You can use this to debug scripts, enhance them, or make general changes.** If you agree with the AI recommendations, you can choose to have it automatically update the file using the “**Apply Recommendations**” button.



Code: ScriptEditor

File: None (Mono Script)

Bug description:

Explain the issues you're having with this script.

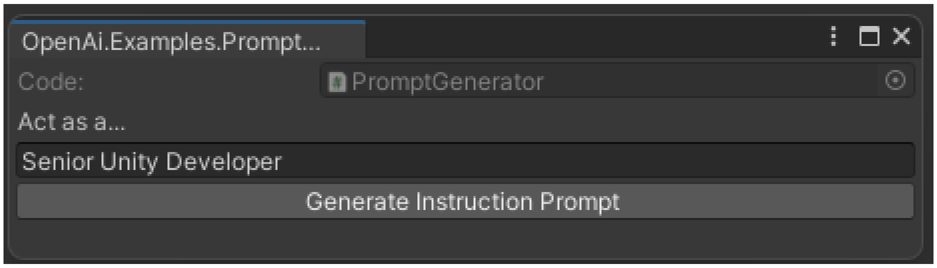
You can access this tool in Unity by going to ***Tools > OpenAI > Script Editor***

How To Use

Simply enter your request in the text box and then press the “**Identify issues**” button. If you’d like to apply the recommendations to the file, press the “**Apply Recommendations**” button to overwrite it.

Prompt Generator

This tool allows you to analyze any script in your project using AI, which will take into account whatever message you send it and respond back with suggestions. **You can use this to debug scripts, enhance them, or make general changes.** If you agree with the AI recommendations, you can choose to have it automatically update the file using the “**Apply Recommendations**” button.



Code: PromptGenerator

Act as a...

Senior Unity Developer

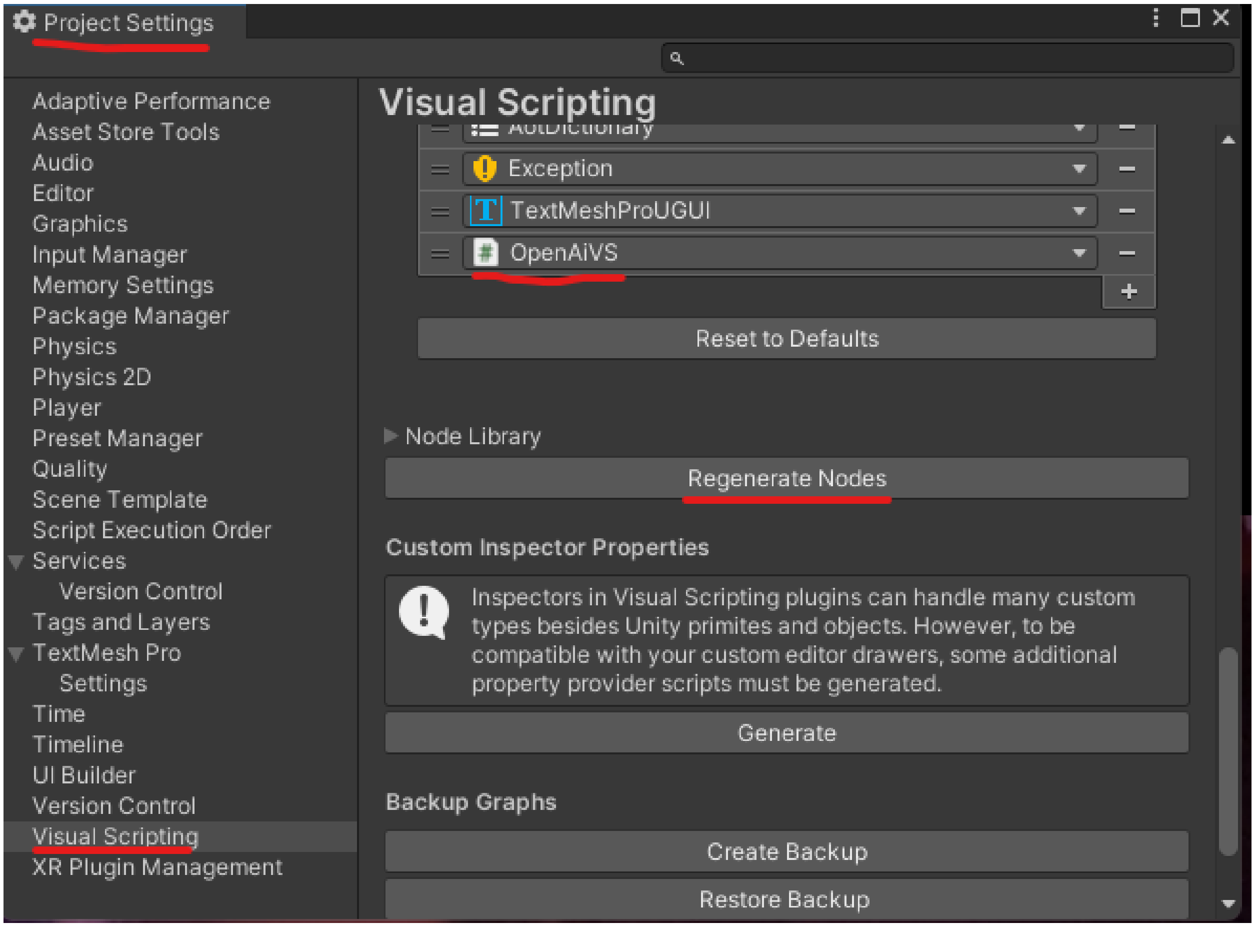
You can access this tool in Unity by going to ***Tools > OpenAI > Prompt Generator***

How To Use

Simply enter your request in the text box and then press the “**Identify issues**” button. If you’d like to apply the recommendations to the file, press the “**Apply Recommendations**” button to overwrite it.

Visual Scripting Demos

1. To use the visual scripting features, you'll first have to goto ***Project Settings > Visual Scripting*** and initialize if you haven't already.
2. **Add OpenAiVS to your node types** and then press the "**Regenerate Nodes**" button.



Inspectors in Visual Scripting plugins can handle many custom types besides Unity primites and objects. However, to be compatible with your custom editor drawers, some additional property provider scripts must be generated.

In-Game AI Chat

Scenes > Visual Scripting - Chat Demo

The **Controller gameObject** has the script graph containing all of the logic. It also contains the **OpenAiVS script** which is **required** on any gameObject using OpenAI with visual scripting.

Simply start play mode and enter in any message to the AI. You can control the settings of the AI from within the **OpenAiDemo script graph** that's attached to the controller.

In-Game Language Translation

Scenes > Visual Scripting - Translate Demo

The **Controller gameObject** has the script graph containing all of the logic. It also contains the **OpenAiVS script** which is **required** on any gameObject using OpenAI with visual scripting.

Simply start play mode and enter in the language you'd like to translate your messages to, then write a message to send. You can control the settings of the AI from within the **OpenAiDemo script graph** that's attached to the controller.

OpenAI API

If you're interested in creating your own OpenAI tools and integrations, you can read more documentation about how the code works in **OpenAI > Documentation**

The code for this asset is based on the work of:

- Hexthedev - hexthedev/OpenAi-Api-Unity: Integration for the OpenAi Api in Unity (github.com)
- OkGoDoIt - OkGoDoIt/OpenAI-API-dotnet: A C#/.NET SDK for accessing the OpenAI GPT-3 API (github.com)