OpenAI API Readme

Transform your Unity project into an intelligent, language-aware application with OpenAI Unity Integration. With just a few lines of code, you can integrate OpenAI's powerful text completion models directly into your Unity project, allowing you to generate natural language text in real time.

Sample Scene

The quickest way to get started is to view the sample scene here: Scene (UnityEngine.SceneAsset)

Project Structure

> OpenAI (UnityEngine.DefaultAsset) Root file location
> docs (UnityEngine.DefaultAsset) More documentation!
> Editor (UnityEngine.DefaultAsset)
> Prefabs (UnityEngine.DefaultAsset) Where we save prefabs used to support this editor window.
> EditorUtils (UnityEngine.MonoScript) Random useful tools used by other editor scripts.
> OpenAiApiExampleEditor (UnityEngine.MonoScript) Editor script for OpenAiApiExample
> OpenAiCredentialsWindow (UnityEngine.MonoScript) Editor window to help with credential setup.
> OpenAiImageReplaceEditor (UnityEngine.MonoScript) Editor script for OpenAiImageReplace
> OpenAiTextReplaceEditor (UnityEngine.MonoScript) Editor script for OpenAiTextReplace
> OpenAiWindow (UnityEngine.MonoScript) Edior window you're probably looking at right now!
> Images (UnityEngine.DefaultAsset) Default location for images generated and default initial save file location.
> Runtime (UnityEngine.DefaultAsset)
> CoroutineRunner (UnityEngine.MonoScript) Helper for retriving a monobehavior to run co-routines in
> OpenAiApi (UnityEngine.MonoScript) OpenAI API core interface. Formats and sends requests and parses response
> OpenAiApiExample (UnityEngine.MonoScript) **(MonoHebavior)** Example script for simple OpenAiApi usage
> OpenAiImageReplace (UnityEngine.MonoScript) **(MonoHebavior)** Simple replacement of text in a unity scene
> OpenAiTextReplace (UnityEngine.MonoScript) **(MonoHebavior)** Complex exmaple of image generation and manipulation
> Utils (UnityEngine.MonoScript) Helpful utils used by other scripts.
> Images (UnityEngine.DefaultAsset) Where we save temp images generated by OpenAI.

Also note, these Tiny PHX dependencies imported to support OpenAI API here:
• Readme (UnityEngine.DefaultAsset)
• Shared (UnityEngine.DefaultAsset)

Get Started

Requirements

• Unity 2021.3 or later
• OpenAI API from Unity Asset Store https://assetstore.unity.com/packages/slug/247238)

Setup

1. Create an OpenAI Account
https://platform.openai.com/signup

2. Get your Organization ID from the "Settings" page
https://beta.openai.com/docs/api-reference/authentication)

3. Create an *API Key* on the API Key page
https://platform.openai.com/account/api-keys

4. In Unity, open the OpenAI window
Window > OpenAI

5. Go to the "Credentials" tab and add fields your just retrieved.
6. This will create a file in your users forlder in this format:

```
{
"private_api_key":"YOUR-API-KEY",
"organization":"YOUR-ORG-ID"
}
```

7. Add one of the built-in components to your scene:
8. Add the `OpenAiImageReplace` or `OpenAiTextReplace` example components to any GameObject in your scene.
9. Add a prompt and click `Generate Image` or `Generate Text`

Out-of-the-Box Components

OpenAI Unity Asset includes three components for integrating OpenAI APIs into Unity games:

• **OpenAiApiExample** for both text completion and image generation
• **OpenAiImageReplace** for replacing sprites with AI-generated images
• **OpenAiTextReplace** for replacing text objects with AI-generated text.

Scripting Interface

Here's an example of how you can create a text completion request and image generation request in Unity using the OpenAI Unity Integration:

Generate Text

Simple text generation

```
using UnityEngine;
using OpenAi;

public class SampleScript : MonoBehaviour {
```

```
async void Start() {
var openai = new OpenAiApi();
var completion = await openai.CreateCompletion("Hello world");
Debug.Log("OpenAI Response: " + completion.Text);
}
}
```

Using a callback instead async/await

```
openai.CreateCompletion("Hello world", completion =>
{
Debug.Log("OpenAI Response: " + completion.Text);
});
```

Generate Images

Simple image generation

```
using UnityEngine;
using OpenAi;

public class SampleScript : MonoBehaviour {
async void Start() {
var openai = new OpenAiApi();
var image = await openai.CreateImage("Hello cat");
Texture2D texture = image.Texture;
}
}
```

Using a callback instead async/await

```
openai.CreateImage("Hello world", image =>
{
Texture2D texture = image.Texture;
});
```

Review

A reputable reviewer had this to say about the asset:

"*Overall, the code seems to be well-organized and follows good coding practices such as encapsulation and modularization.***"**

- ChatGPT

Documentation

For more information on how to use OpenAI's APIs, refer to the OpenAI documentation

- OpenAI documentation: https://beta.openai.com/docs