Thank you for purchasing this asset. I hope you enjoy using it.

Contact me here, for bug reports, suggestions, or feedback.

Don't forget to leave a review if you enjoy using Text generation toolkit, this helps me a lot

Table of Content

Overview

Want to put a language AI in your game for your NPC, or just want to ask ChatGPT some questions inside the unity Editor? Then this plugin is for you!

This plugin allows you to:

- Use OpenAI's GPT-3 and ChatGPT Models, AKA GPT, into your game with both runtime, and editor usage!
- Azure Text-To-Speech, and Speech-To-Text to have verbal conversations with the AI.
- Use AI to generate your voice lines, and save them as audio clips!
- Use AI to determine if the player is being kind, or hatfull!
- Use GPT to reword what you want your NPC's to say!
- Use GPT with memory for better conversations!
- Use our Memory optimizer so you can get better and cheaper conversations!
- Use GPT to help you with your code!
- Use GPT as a spell checker!
- Security for your open AI key when you publish your game!

How To Set Up Azure Key Vault

Note: This is for deployment usage only. You will not need to follow these steps until you are ready to create a runtime build. The plugin will automatically look for the Azure Key Vault call once a program is compiled.

Since this asset has runtime support, it also needs a safe and secure way to store your OpenAI Key. I have created an easy way to access Azure Key Vault inside of Unity and integrated it in this plugin so that you can feel secure using Azure Key Vault. Please note, you are responsible for the safety of your API keys wherever you choose to store them. I prefer to use Azure Key Vault.

1. Open notepad then copy, and paste the following onto it:

   Azure Key Vault Name:
   Azure Secret name:
   Azure Client ID:
   Azure Client Secret Value:
   Azure Tenant ID:

2. Go to Microsoft Azure, and sign in, or make a new free account
   a. If you believe your project will really take off, sign up Microsoft's Founders Hub, and get thousands of dollars in free credits.

3. Click Create resources, which is under Azure services.

4. In the search bar search for "Key Vault".

5. Click the one called "Key Vault" made by Microsoft.

6. Either make a new resource group or use an old one.

7. Name the Key Vault whatever you want, and choose a location near you.

8. Then at the bottom click “Preview + create”, and copy and paste the name of the resource onto the notepad.

9. When it is done deploying, go to the top left of the page and click the three horizontal lines, and then click “Azure Active Directory”.

10. Copy “Tenant ID” and paste it onto notepad.

11. Then on the left click “App registrations”, then click “New registration”.

12. Name it whatever you want but remember what you named it, then click Register.

13. Now copy the “Application (client) ID” and paste it onto notepad.

14. Now on the left click “Certificates & secrets”, and make sure you are on “Clients secrets”.

15. Then click “new client secret”, then choose how long you want it to last. IMPORTANT: once you make the client secret you can not change it, and if you publish a game then do not update it with the new client secret, the Azure key vault will not work.

16. Click “Add”, and copy the “Value”, and paste it onto your notepad.

17. Go back to your Key Vault, and click “Secrets” on the left side.

18. Click “Generate/Import” and name it whatever you want.

19. Now paste your Open AI key onto the “Secret value”.

20. Click “Create”, then click “Access policies” On the left side.

21. Click “Create”, then under “Secret permissions” select “Get”, and “Set”.

22. Click “Next”, Then in the search bar type in the name of the access policy you made.

23. Click “Next” 2 times, then click “Create”. Then go back to unity.

24. Now go to the “Keys” scriptable object located at “Plugins/GTP3/Editor”

25. Then paste all the variables you copied onto the notepad onto the corresponding variable. NOTE: this scriptable object will not be compiled in the build.

26. Then if you need to update the OpenAIKey in the Azure Vault you can put the new OpenAIKey in the scriptable object then click “Update Vault”.

27. Then make a game object, with the “Key Vault Manager”, and paste all the variables you copied onto the notepad onto the corresponding variable. Again.

28. Then make sure that the game object, with the “Key Vault Manager” is loaded before anything is sent to the AI.

How to Use Azure Keyvault Core

1. Make a new C# script and attach a new Gameobject

2. In your C# script you will need to add the following Using Tag

```
using TheAshBot.Assets.TextAndVoiceGenerationToolkit.AzureKeyVault.Core;
```

3. Then In the start function add this code

```
KeyVault keyVault = new KeyVault();
```

4. Then you will need to asinine the values you put into the scriptable object into the "**keyVault**" Variable

```
keyVault.vaultName = "<YOUR AZURE CLIENT SECRET VALUE>";
keyVault.secretName = "<YOUR AZURE TENANT ID>";
keyVault.clientId = "<YOUR AZURE KEY VAULT NAME>";
keyVault.clientSecret = "<AZURE SECRET NAME>";
keyVault.tenantId = "<AZURE CLIENT ID>";
```

5. Then when you need the Key just do the following call

```
keyVault.GetSecret();
```

6. If you need to set the key to be a new value do the following call.

```
keyVault.SetSecret("<YOUR KEY>");
```