Dracarys Interactive supplies Unity-based tools and templates for creating quality interactive 2D, 3D, and XR applications.

Welcome to *AI Studio*!

Introduction

AI Studio provides a flexible open-source framework for integrating Unity games and applications with AI systems that enable real-time, speech-driven dialogue and actions. The core framework may be integrated with your choice of speech and text recognition and generation models and APIs.

This framework is a byproduct of a case study exploring a step-by-step integration of OpenAI's GPT-3.5 model into Unity 3D for the purpose of enhancing non-playing characters (NPCs) in video games and interactive applications. By leveraging GPT-3.5's natural language processing capabilities, we aimed to create NPCs capable of dynamic real-time interactions with players and each other! Our methodology involved integrating GPT-3.5, implementing speech-driven dialogues, and incorporating humanoid avatars with lip-syncing and emotive animations.

Key features of *AI Studio* include:

- Core action-driven architecture implementing multiple character dialogues which feeds asynchronous events into a central queue for processing by the main Unity thread
- Basic *Scriptable Object* framework for representing characters and character dialogues and providing AI model prompt engineering inputs
- Abstracted interfaces to key implementation boundaries including speech recognition and synthesis, natural language dialogue generation, and dialogue character animation and movement
- Feature extensions including streaming and non-streaming audio, continuous and discrete player voice recognition, conditional dialogue triggering, player synthetic voice option, programmatic chat injection, and basic non-player character movement
- Example scenes demonstrating integration with Microsoft Cognitive Services Speech SDK, OpenAI's chat completion models (e.g. GPT-3.5), Unity Multipurpose Avatars (UMA), SALSA LipSync Suite, and

Ready Player Me avatars
- YouTube videos documenting installation and example scenes

Using *AI Studio*

AI Studio has several dependencies that require licensing and importing before all the sample scenes can be compiled and run. The target platform for running sample scenes is a Windows 10 or later PC running Unity 2020.3 or later. There are two sets of sample scenes included in *AI Studio*. “Simple” scenes demonstrate the core architecture and “City” scenes with additional dependencies add a game world and humanoid avatars. One “City” scene, *Avatar Wars*, adds an additional dependency on Ready Player Me.

Running the Simple Scenes

To run the Simple scenes which demonstrate the core architecture, the following licenses must be obtained, and accompanying Unity packages must be installed.

AI Studio

- Download and install the SRI *AI Studio* package from the Unity Asset Store or via GitHub or from here.
- You may get *missing Prefab* errors due to the dependencies *…/Prefabs/Environment.pref* has on the CITY package, but this can be ignored when running the Simple scenes.

OpenAI

- An OpenAI account and API key are required to run all sample scenes. Once you have obtained the key, you must create a folder *.openai* under your home folder with a file *auth.json* containing the following:

> { private_api_key : "YOUR_KEY_HERE" }

- Alternatively, you can modify the OpenAI Unity API authentication type from “Local File” to “String” and enter your OpenAI API key directly via the Unity Inspector.
- Add and import the OpenAI Unity API package from GitHub.
- Add the *Scripting Define Symbol* **USE_COM_OPENAI_API_UNITY** to the *Player Project Settings* under *Script Compilation*.

Microsoft Speech Services

- A subscription key for the Microsoft Azure Speech Service.
- Once you have obtained the key, you must set the environment variables COGNITIVE_SERVICE_KEY and COGNITIVE_SERVICE_REGION.
- Alternatively, you can enter the key and region fields directly via the Inspector.
- Download and install the Speech SDK for Unity Package.
- Add the *Scripting Define Symbol* **USE_MICROSOFT_COGNITIVESERVICES_SPEECH** to the *Player Project Settings* under *Script Compilation*.

There are three Simple scenes that you can run under *Skylands Research Institute/AI Studio/Scenes*:

- Simple 2 NPC - This scene demonstrates dialogue between two NPCs.
- Simple 3P Player and 1 NPC - One third-person player and one non-player.
- Simple 3P Player and 1 NPC - One third-person player and two non-players.

This YouTube video demonstrates how to create a Unity project from scratch that can run the Simple example scenes.

Running the City Scenes

To run the City scenes which include a city game world and avatar characters, the following licenses must be obtained, and accompanying Unity packages must be installed.

Cinemachine

- The Cinemachine package may be imported via the Unity Package Manager through the Unity Registry.
- Add the *Scripting Define Symbol* **USE_CINEMACHINE** to the *Player Project Settings* under *Script Compilation*.

Unity Starter Assets

NOTE! Unity Starter Assets First and Third-person package scripts are included in the AI Studio package so the following steps to import and modify these packages are for information only. It should not be necessary to import these packages. However you will need to add the *Scripting Define Symbol* STARTER_ASSETS_PACKAGES_CHECKED to the *Player Project Settings* under *Script Compilation*.

Starter Assets – Third Person Character Controller

- The Unity Asset Store Starter Assets - Third Person Character Controller, a free asset, must be imported and a minor C# script change must be applied to make the controller work with UMA
- Accept the import of any dependencies.
- Changes to StarterAssets\Scripts\ThirdPersonController.cs
  - Expose _animator field (line 104):

    > [SerializeField] private Animator _animator;

  - Use attached Animator if set (line 139 and 158):

    > if (!(_hasAnimator = _animator)) _hasAnimator = TryGetComponent(out _animator);

  - Make method OnFootstep public (line 372)
  - Make method OnLand public (line 384)

Starter Assets – First Person Character Controller

- The Unity Asset Store Starter Assets - First Person Character Controller, a free asset, must be imported.

UMA 2 - Unity Multipurpose Avatar

- The Unity Asset Store UMA 2 - Unity Multipurpose Avatar, a free asset, must be imported.

CITY package

- The Unity Asset Store CITY package, a free asset, must be imported.

SALSA LipSync Suite

- The Unity Asset Store SALSA LipSync Suite, **a paid asset**, must be imported.
- **NOTE!** You will also need to download and install the *OneClick UMA DCS* add-on package from here.
- SALSA will *automatically* add the *Scripting Define Symbol* **CMS_SALSA** to the *Player Project Settings* under *Script Compilation*.

Mixamo Animations

- Import a set of Mixamo animations from here.

This YouTube video demonstrates how to extend the Unity project configured to run the Simple scenes to run the City scenes with the exception of *Avatar Wars* which will be covered in the next section.

Running *Avatar Wars*

The scene *City Scene Avatar Wars 2 NPC* requires the installation of *Ready Player Me*.

Ready Player Me package

- Follow these instructions to setup RPM and install the software into Unity. Make sure to also import all the loader samples.
- Import an RPM avatar from here.

This YouTube video demonstrates how to extend the Unity project configured to run the City scenes to run *Avatar Wars*.