**Project Title:**
AI-Powered Interactive Poetry Generation Application

**Project Description:**
This project aims to develop an interactive Unreal application featuring a 4 virtual character modeled, who can compose and recite poetry inspired by prominent Arab poetic styles. Each style is represented by a unique virtual avatar with its own recitation characteristics, bringing the distinct essence of each poet's voice and mannerisms into the experience."

This wording keeps the focus on the project without directly naming the poets, allowing for a more generalized approach.

Users can also **ask questions** to the virtual character, which responds using AI-based answers. The application leverages IBM Watson and OpenAI for text generation and voice processing, creating a rich, interactive experience.

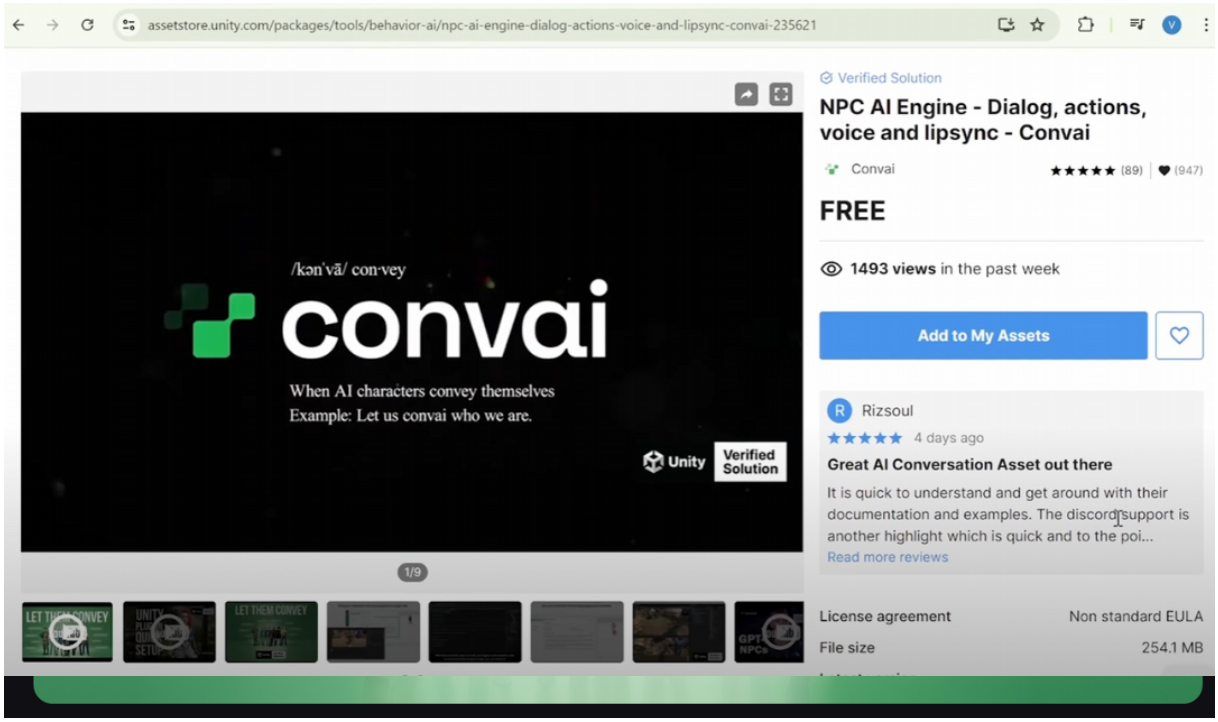
**Project Components:**

1. **Virtual Character (Avatar):**
    - The avatars are visually designed in Unreal to reflect each poet's unique persona and spirit.
2. **AI Poetry Generation:**
    - Utilizing **OpenAI's model** to train AI on the distinctive styles of each poet by feeding it a collection of their poems.
    - The model generates new poetic lines based on user prompts, allowing customization by theme (e.g., love, nostalgia, wisdom).
3. **Voice Technology:**
    - **IBM Watson Text-to-Speech** is used to convert generated text into audio, reciting poems in a voice that matches each poet's style.
    - Additional Speech-to-Text feature for converting user speech into text, allowing voice-based questions to the character.
4. **Interactive Question and Answer System:**
    - Using **IBM Watson Assistant** to process user questions (text and voice) and generate interactive responses based on AI-powered data.
5. **User Interface and Experience (UI/UX):**
    - Designing a flexible, appealing interface that lets users select a poet, ask questions, and listen to poetry.
    - Adding customization options such as adjusting the speed and tone of the voice for a more realistic, user-friendly experience.

## Project developing:

## Install the Convai Plugin

1. **Download the Plugin**:
    - Open the **Unreal Engine Marketplace** by clicking on **Marketplace** in the top toolbar of Unreal Engine.
    - In the Marketplace, search for **Convai**. If it's not available in the Marketplace, visit Convai's official website, where they may provide a download link.




2. **Add to Project**:
    - After you find the Convai plugin, click on **Install** to add it to your Unreal Engine library.
    - Select the project you want to use it with, then click **Add to Project**.
3. **Enable the Plugin**:
    - Go back to your Unreal Engine editor, open your project, and go to **Edit > Plugins**.
    - In the Plugins menu, search for **Convai** to find it, and enable the checkbox next to it.
    - Unreal Engine may prompt you to restart. Click **Restart Now** to apply the plugin.

## Step 2: Set Up Convai Account and API Key

1. **Create a Convai Account**:
    - Go to **Convai's website** and create an account if you haven't done so already.
2. **Get Your API Key**:
    - Once logged into your Convai account, look for an **API Key** (usually in the account settings or developer section). This key allows Unreal Engine to connect to Convai's servers.
    - **Copy** the API key as you'll need it soon.

## Step 3: Configure Convai Plugin in Unreal Engine

1. **Open Project Settings**:

    - In Unreal Engine, click **Edit > Project Settings**. This will open a window with various settings for your project.
2. **Locate Convai Settings**:
    - In the **Project Settings** window, use the search bar at the top left to type **Convai**.
    - When you see **Convai Settings** under Plugins, click on it.
3. **Enter Your API Key**:
    - You'll see a field for the API Key. Paste your API key here.

## Step 4: Add a Convai Character to Your Scene

1. **Open the Content Browser**:
    - In Unreal Engine, locate the **Content Browser** panel, usually at the bottom of the editor.
    - If you don't see it, go to **Window > Content Browser** to open it.
2. **Find the Convai Character Blueprint**:
    - Inside the Content Browser, look for a **Convai** folder or similar. Open it to find a blueprint for the Convai character (usually named something like `BP_ConvaiCharacter`).
3. **Drag the Character into the Level**:
    - Click on the Convai character blueprint, then drag and drop it into your level (the main view window where you design your game environment).
4. **Adjust the Character's Settings (Optional)**:
    - Double-click the Convai character in the level to open its **Details** panel. Here, you can adjust properties like **appearance**, **animation**, and other options.

## Step 5: Set Up Conversational Interactions

1. **Open the Character Blueprint**:
    - In the Content Browser, right-click on the Convai character and select **Edit Blueprint**. This opens the blueprint editor for the character.
2. **Add the Convai Component**:
    - In the **Components** tab (usually on the left), click **Add Component** and search for **Convai Component**. This allows the character to respond to spoken or typed input.
3. **Configure Voice Settings**:
    - With the Convai Component selected, look in the **Details** panel to find settings like **Voice Type** and **Language**. Adjust these settings based on your preferences.
4. **Set Up Event Triggers**:

- In the **Event Graph** (the main scripting area of the Blueprint), you can add events like **keyboard inputs** or **mouse clicks** to trigger conversations with the character. For example:
  - **Right-click** in the Event Graph.
  - Search for **Input Key** and choose a key (like **E**) to start the conversation.
  - Connect this to the Convai Component's **Start Conversation** node.

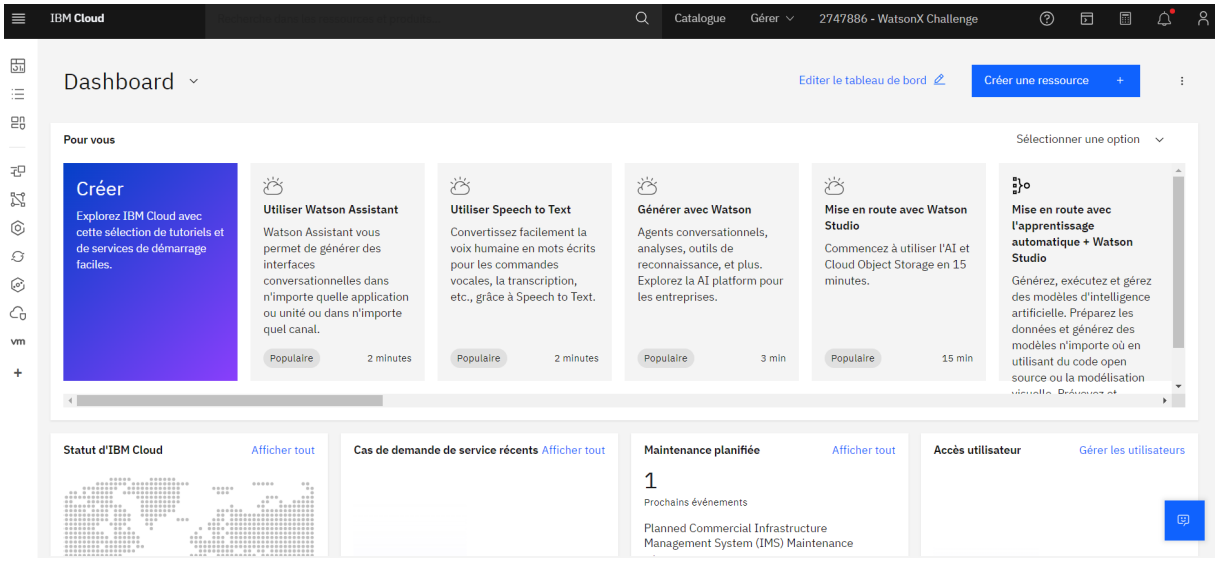
### Step 6: (Optional) Integrate API Calls

- You can use additional scripting to make advanced API calls to Convai, but this can be more complex. For now, you can focus on setting up the basics and return to this step when you're more comfortable with Unreal Engine.

# Custom voice Integration

In version 2.1.5 and higher, integrating custom voices has become much easier

### Integrating AI with IBM Watson




### Watson Setup:

- Create an IBM Cloud account.
- Activate Watson services such as Speech to Text, Text to Speech, or Conversation Assistant.
- Retrieve the API keys for each of the services.

### IBM Watson SDK for Unity:

- Install the SDK in your Unity project and connect it with the API keys.

**5. AI Features to Integrate**

- **Speech-to-Text:** Convert the player's voice into text.
- **Text-to-Speech:** Make your avatar speak.
- **Conversation AI (Chatbot):** Allow the avatar to answer questions using IBM Watson Assistant.

**6. Example Scripts** Here's a simple example to get started with Text-to-Speech using IBM Watson:

```
csharp
Copier le code
using IBM.Watson.TextToSpeech.V1;
using IBM.Cloud.SDK.Authentication.Iam;

public class WatsonTextToSpeech : MonoBehaviour
{
    private TextToSpeechService _service;

    void Start()
    {
        IamAuthenticator authenticator = new IamAuthenticator(apikey:
"your-api-key");
        _service = new TextToSpeechService(authenticator);
        _service.SetServiceUrl("your-service-url");
        _service.Synthesize(OnSynthesize, "Hello, I am a smart avatar",
voice: "fr-FR_ReneeV3Voice", accept: "audio/wav");
    }

    private void OnSynthesize(DetailedResponse<byte[]> response, IBMError
error)
    {
        if (error == null)
        {
            AudioClip clip = WavUtility.ToAudioClip(response.Result);
            AudioSource audioSource =
gameObject.AddComponent<AudioSource>();
            audioSource.clip = clip;
            audioSource.Play();
        }
        else
        {
            Debug.LogError(error.ToString());
        }
    }
}
```

**Text-to-Speech:** This script converts a phrase into audio with Watson and plays the sound in Unity.

In this tutorial, we will guide you through the process of integrating your own text-to-speech (TTS) device.

## 1. Disabling Audio Receiving

Starting from version 3.0.0, you can right click in Project tab, then select `Inworld > Default Settings`, and turn off audio by unchecking `Audio` option.

If you are using previous version, after version 2.1.6, you can turn off audio by unchecking the `Can Receive Audio` option in `Default Settings`. See the screenshot

## 6. Final Deployment

Once everything works well, you can set the Holobox to automatically launch the application every time it powers on.