Choose-Your-Own-Adventure Framework

for the Dialogue System for Unity

Table of Contents

Overview..... 1
Instructions..... 2
Frequently Asked Questions..... 3
Art Credits..... 3

OVERVIEW

The CYOA Framework is a bonus add-on for the Dialogue System for Unity. You can use it to make choose-your-own-adventure games.

To use the CYOA Framework, you must also import the Dialogue System into your project.

The CYOA Framework contains these subfolders:

- **Data**: Dialogue database file (your story) and text table (for localizing UI content)
- **Particles**: Particle effects for the example scene.
- **Prefabs**: Contains a Dialogue Manager prefab variant customized for this framework
- **Resources**: Assets that are loaded at runtime by the example scene.
- **Scenes**: The two scenes used by the CYOA Framework.
- **Scripts**: The scripts that run the CYOA Framework.
- **Textures**: Textures used by the menus.

To familiarize yourself with the CYOA Framework's features:

1. Add the scenes in the **Scenes** folder to your project's build settings.
2. Play the scene **Scenes/Title**.

INSTRUCTIONS

To use make your own story:

1. Add the scenes **Title** and **Story** to your build settings. **Title** should be the first scene in the build settings.

2. Open the **Title** scene and customize the UI appearance. There are two canvases:

- Dialogue Manager/Common Canvas: Contains a confirmation menu (ok/cancel) and the Language Menu. The Language Menu allows players to select the language they want to play the game in.
- Canvas: The canvas for the title scene. If you don't want to use a feature such as the Language Menu, deactivate the corresponding button in Title Menu.

3. Open the **Story** scene and customize the UI appearance. You can ignore the Dialogue Manager in this scene; it's a stand-in for testing your game without having to go through the Title scene.

4. Check over **Data/Text Table**. This contains localized text for the menus.

5. Edit **Data/Dialogue Database**, or start with your own new dialogue database. If you start with a new one, assign it to the Dialogue Manager GameObject in both scenes.

The story is contained in a conversation named "**Story**". Each gray dialogue entry is a page of the story. The blue (player) entries are responses.

If your conversation gets too big, you can connect to other conversations in the dialogue database to break it up.

Note how you can use sequencer commands to control activity in the scene. You can also use the **Conditions** and **Scripts** fields to add conditions and set variables in the Lua environment.

The CYOA framework introduces a new sequencer command, **SetSprite(GameObject, Sprite)** where GameObject is the name of a GameObject with an Image component and Sprite is the name of a sprite in a Resources folder.

FREQUENTLY ASKED QUESTIONS

Q: Why doesn't the SetSprite() sequencer command show my images?

A: In the Project view, inspect your image file. Change Texture Type to Sprite (2D and UI). Then click Apply. The SetSprite() sequencer command requires that the image is configured as a sprite.

Q: How do I handle text longer than the screen can hold?

A: In the Story scene, customize NPC Panel. This is where the text is displayed. Add a ScrollRect, and make NPC Subtitle Line the content of the ScrollRect. If you're not familiar with ScrollRects and Scrollbars in Unity UI, tutorials #9 and #10 on this page will help:
https://unity3d.com/learn/tutorials/modules/beginner/ui

If you have so much text that it may require scrolling, you probably won't want to use the typewriter effect; remove the Unity UI Typewriter Effect component from NPC Subtitle Line. In this case, you can remove NPC Subtitle Line Reminder and, in the dialogue UI, assign NPC Subtitle Line to Dialogue > Response Menu > Subtitle Reminder > Line.

Q: Can I play a sound file that applies to the whole project as a loop?

A: Yes. Create a new GameObject, or choose an existing GameObject. Say you've named the GameObject LoopingAudioObject. Add an AudioSource to it, and tick the Loop checkbox. Then use a sequencer command in this format:

```
Audio(Music/LopingSting, LoopingAudioObject)
```

This will play LopingSting (or whatever audio file you specify) through LoopingAudioObject's AudioSource.

ART CREDITS

- Cottage: http://www.clker.com/clipart-25732.html
- Gear from: http://www.clker.com/clipart-gear-11.html
- Music: "Ambler" Kevin MacLeod (incompetech.com)
  Licensed under Creative Commons: By Attribution 4.0 License
  http://creativecommons.org/licenses/by/4.0/