# Ink Dialogue System Documentation

Must Use Ink Unity Integration Package:

https://assetstore.unity.com/packages/tools/integration/ink-unity-integration-60055

1. Choice Event (Class)
2. Dialogue Canvas (Mono Script)
3. Dialogue Ink Manager (Mono Script)
4. Dialogue System Display (Mono Script)
5. Dialogue Trigger (Mono Script)
6. Dialogue Trigger Examples (Mono Scripts)
7. Option Button (Mono Script)

1- Choice Event: Class made to store the value string of a choice, and the event that goes along with it. So that it may be triggered later.

2- Dialogue Canvas: Singleton class. This MonoScript should be attached to the canvas where the dialogue should be. It will instance whatever displays are passed to it from Dialogue Trigger. Nothing more needs to be done to set it up.

3- Dialogue Ink Manager: Singleton class. This MonoScript should be attached to the GameManager or wherever the text of the dialogues is in scene. It handles The Ink file so the canvas can get the text to display, or the options to choose. You only need to pass the text file generated by ink.

4- Dialogue System Display: this should be attached to every prefab display that you want to use in your game. It can be a display for a npc, or a display for a party member. You need to attach (if any) the following:

- Title: TextMeshProUGUI Component (Currently takes the first tag in the line)
- Text: TextMeshProUGUI Component (The Text Dialogue)
- Image: ImageUI (Currently doesn't take any value, might be added in a later version) (I recommend using the title to swap between them)
- Option Parent: Transform (Where the Options will be instanced, must handle child arrangement) (For Option Displays)
- Option Button Prefab: OptionButton (the option prefab, the button that will appear per choice available) (For Option Displays)
- Continue Button: Button To be added the onClick Event Continue. (For Text Displays)

This prefab will be passed to the Dialogue Trigger responsible for the dialogue.

5- Dialogue Trigger: this should be attached to every GameObject That somehow will trigger dialogue. It only calls Dialogue canvas and pass the Events necessary. Can be destroyed after calling the dialogue. You need to attach the following:

- Text Display: Dialogue System Display (Prefab for normal dialogue)
- Option Display: Dialogue System Display (Prefab for Options dialogue)
- Knot name: string (where the dialogue should pick up from the ink file) (Must be the same as one knot in the ink file.)
- OnStartEvent: UnityEvent (Event To trigger at start of the dialogue)

- OnCloseEvent: UnityEvent (Event To trigger at the end of the dialogue)
- ChoiceEvents: List<Choice Event> (List with all the choices with repercusions (if any) they will trigger their Unity Event if the string matches with a choice made)

6- Dialogue Trigger Examples: Simple classes that showcase how Dialogue Trigger can be used. Currently there is a class for: OnDestroy, OnAwake, and OnTriggerEnter. You can make more for your current needs. Ex: DialogueTriggerOnAwake will call Dialogue Trigger at the start of the game/scene.

7- Option Button: Prefab for Dialogue System Display (Options), will Display the text of the option, and if possible, the choice index. You need to attach the following:

- Choice Text: TextMeshProUGUI Component (The text of the choice)
- Index Text: TextMeshProUGUI Component (The index of the choice) (if any)