Ink Narrative Scripting Language

[Official basics guide](https://www.inklestudios.com/ink/web-tutorial/)

Custom Unity-Ink Integration

1. Keywords in Ink scripts can affect the VN on the Unity level.
2. All VN\_Manager keywords such as [Character], must be case sensitive spelled the same as set up in Unity.
3. Ink syntax cannot interfere with the VN\_Manager keywords.

# **Dialogue**

| CharacterA: Hi I’m CharacterA! |
| --- |

1. “CharacterA” is the character name and will show up in the name box
2. “Hi I’m CharacterA!” is the content and will show up in the content textbox.
3. Space between colon and [content] is ignored
4. Using “Narrator” for [Character name] doesn’t show up in the name text box, but the content is displayed in italics.

| Hi I’m CharacterA! |
| --- |

1. A dialogue line with no character name is assumed to be spoken by the player character.
   1. Choice text is considered a dialogue line unless it is excluded with the square bracket Ink syntax.
2. Quotation marks in [content] aren’t displayed in the content text box. However if the content is choice text, the quotation marks are displayed.

# Emotion Tags (Not needed)

| CharacterA: Hi I’m CharacterA! #CharacterA\_happy |
| --- |

1. Changes Character A’s sprite to CharacterA\_happy as soon as the dialogue line is starting to be displayed.
2. # At the end of a line is Ink syntax for a tag. All tags are assumed to signify emotion changing corresponding to the line’s speaking character.

# **Command Calls**

1. Start a line with “>>>” to invoke command calls.
2. [command name]([arg0, arg1, ...]) is the standard command format.
3. “;” is used to separate multiple commands on a single line.
4. Normally command calls wait for the previous command to finish running
5. “!” at the beginning of a command call makes a command not be waited on

| >>> !CharEnter(CharacterA); CharEnter(CharacterB); |
| --- |

Will make CharacterA and CharacterB enter at the same time.

| >>> CharEnter(CharacterA); CharEnter(CharacterB); |
| --- |

Will make CharacterA enter. After CharacterA finishes entering, Character B enters.

1. Since the command lines are treated the same in Ink, any of the commands can use Ink syntax such as variable text

| VAR player\_character = "bob"  >>> CharEnter({player})  >>> CharEnter(bob) |
| --- |

These CharEnter commands both do the same thing.

Command Call Glossary

# **CharEnter**

CharEnter(character)

1. A character named “character” moves from off screen to on screen.
2. Character must have CharacterData in Unity with corresponding name.
3. The animation and sprite depends on the CharacterData.
4. If CharacterData doesn’t have a DefaultSprite the character will appear invisible
5. If CharacterData doesn’t have a CharacterBox the character box will appear invisible

# **CharExit**

CharExit(character)

1. A character named “character” moves off screen.
2. Character must have CharacterData in Unity with corresponding name.
3. Character must be present in the scene.
4. The animation depends on the CharacterData.

# **Wait**

Wait(duration)

1. Waits for duration in real time before continuing.
2. Accepts decimal and integer values.

# WaitUntilEvent

WaitUntilEvent(UnityEventCode)

1. Waits for UnityEvent with UnityEventCode to be invoked before continuing.

# **TextboxEnter**

TextboxEnter(textbox, cornerDecor)

1. VN textbox enters from off screen to on screen with specified cornerDecor.
2. Textbox must have TextboxData in Unity with corresponding name.
3. The animation and art assets depend on the TextboxData.
4. CornerDecor of corresponding name must exist in the TextboxData.

TextboxEnter(textbox)

1. Overload of TextboxEnter with no cornerDecor.
2. CornerDecor is instead invisible.

# **TextboxExit**

TextboxExit()

1. The current active textbox is moved off screen.
2. Note: Dialogue is still displayed on the textbox if it is off screen.

# **UpdateInkVar**

UpdateInkVar(inkVar, unityVar)

1. Retrieves the current value of unityVar in VN\_SharedVariables and sets the Ink VAR inkVar to that value.
2. Both inkVar must exist in Ink and unityVar must exist in VN\_SharedVariables.

# **SetUnityVar**

SetUnityVar(varName, value)

1. Sets varName in VN\_SharedVariables to value.
2. Value must have the same type as the varName.

| >>> SetUnityVar(points, one) |
| --- |

Is invalid if points in Unity is an int.

# InvokeUnityEvent

InvokeUnityEvent(UnityEventCode)

1. Invokes UnityEvent with UnityEventCode
2. Add AddEventData

# **FadeBlack**

FadeBlack(endValue, duration)

1. Fades the black screen alpha to endValue in duration time.
2. endValue is 1 for fully black and 0 for no black.

FadeBlack(endValue)

1. Overload of FadeBlack that has a default duration of 1 second.

# **MoveBackground**

MoveBackground(newX, newY, duration)

1. Moves the current background image to (newX, newY) position in duration time.
2. Position is based on the image’s RectTransform anchor position.

MoveBackground(newY, newY)

1. Overload of MoveBackground that has a default duration of 1 second.

Unity VN Data Glossary

# **CharacterData**

1. Find in Data/Characters. Data containers for characters.
2. **ScenePosition**: Position of character when on screen.
3. **ScreenEdgeDistance**: When on and off screen, distance away from the screen edge corresponding to ScenePosition.
4. **CharacterTransition**: Enter and exit animation method.
5. **TransitionDuration**: Number of seconds the enter and exit animations take.
6. **CharacterSpriteScale**: Scale of character sprites relative to its native size.
7. **CharacterBoxScale**: Scale of character box relative to its native size.
8. **DefaultSprite**: Starting sprite when entering. Make sure it exists in CharacterSprites.
9. **CharacterBox**: Character box sprite behind the character.
10. **CharacterSprites**: List of all character's sprites

# **CharacterTransition / TextboxTransition**

1. Find in Data/Transitions. Data containers for character and textbox transitions.
2. **SlideCharacterTransition/SlideTextboxTransition**: [DOTween](http://dotween.demigiant.com/documentation.php) based animations
   1. **EnterEase**: Easing of enter animation
   2. **ExitEase**: Easing of exit animation
   3. [Resource for visualizing easing curves](https://easings.net/)
3. **TeleportCharacterTransition**: Outdated. Only for internal use.

# **TextboxData**

1. Data/Textboxes data containers for textboxes

# **Text**

1. TODO quick summary: prefabs

# **VN Settings**

1. TODO quick summary: text speed, others

# **Shared Variables**

1. AddEventData(VN\_EventData data)
   1. Adds event data to be referenced by Ink through VN\_EventData.eventCode
2. RemoveEventData(VN\_EventData data)
   1. Removes event data with matching VN\_EventData.eventCode

# VN\_EventData

1. Unity data class to store information about a UnityEvent used between Ink and Unity
2. VN\_EventData(UnityEvent eventTarget, string eventCode)
3. eventTarget is invoked or listened through Ink based on the eventCode
4. **UnityEventCode:** Referenced in this document for clarity to mean an eventCode of an existing VN\_EventData in VN\_SharedVariables

How to Access VN in Unity

# **Editing the VN Ink Script**

1. In Unity, open the Project panel. Inside the Assets folder, open the InkFiles folder. Click on the VN Inky file to open it in Ink.
2. Whenever you edit the Inky file, make sure to save it.
3. In Unity, open the Console panel. When you save the Ink file, the Console should print out a message saying “Ink compilation is completed”.
4. Open the VN\_Test scene in Assets/Scenes
5. Press the Unity play button to test.
6. Some errors in the VN command calls should be caught and printed to the Console if you check Debug Enabled on the VN\_Manager component. However, not all errors will be explicitly printed since the debug mode is completely custom.

# **Importing and Playing Inky Files**

1. Drag and drop your Inky file into the InkyFiles folder.
2. The Inky file should automatically generate a json file of the same name next to it.
3. In the Hierarchy panel of the VN\_Test scene, click on the VN\_ManagerObject. In the VN\_Manager script, drag and drop the json file into “Ink JSON asset”.
4. It should be ready to play at that point.