Dead-Simple Dialogue Documentation

Demo on itch!

FYI: This is a **non-branching**, minimalist dialogue system, intended for **linear**, narrative, shorter-form games. Binary conditional triggers are supported; but if you need branching or fully stateful dialogue, I highly recommend ink! It's free and has out-of-the-box Unity integration.

Because it's event-driven and decoupled, you can use DSD to trigger or respond to behavior in other systems like FMOD or camera controllers. For a real-world example, see Claymore Ham: Road 2 Reboot.

Demo comes with some general-purpose utilities you're more than free to repurpose, and demo visuals are public domain (see LICENSE.txt in demo folder for sources).

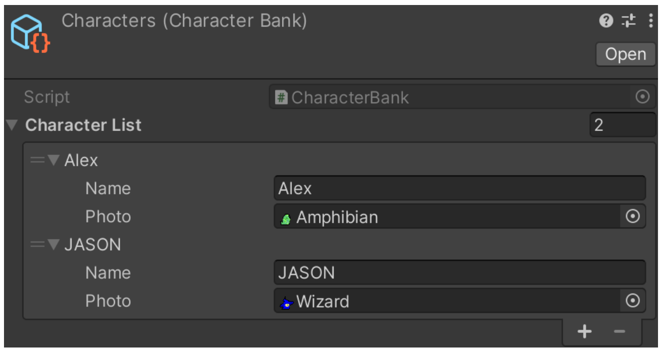
Contents

How does Dead-Simple Dialogue work?

DSD came out of a distributed, multi-person project where non-technical narrative designers needed to be able to hand off a single manuscript to a Unity-facing programmer, from which "cutscenes" could be programmatically extracted and then linked to in-game situations.

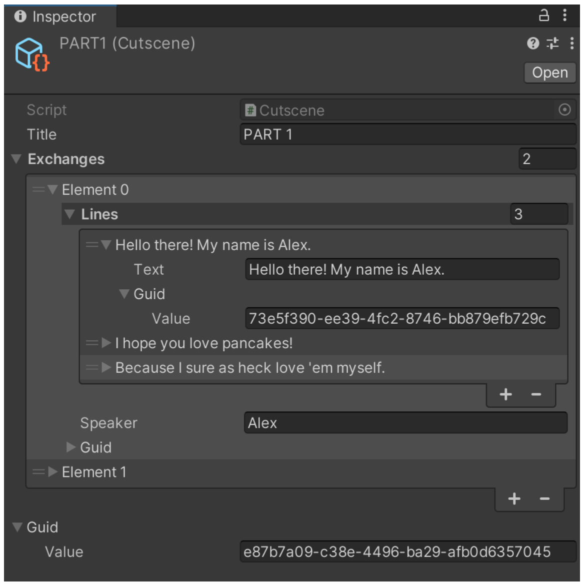
The resulting solution is a one-way translation system that converts marked-up text files to Unity-usable scriptable objects (a series of cutscenes and a character bank).

A runtime dialogue manager singleton responds to cutscene triggers, and passes corresponding manuscript text to canvas elements as needed. (Serialized GUIDs are used for coordinating cross-scene / scene-to-asset event references.)



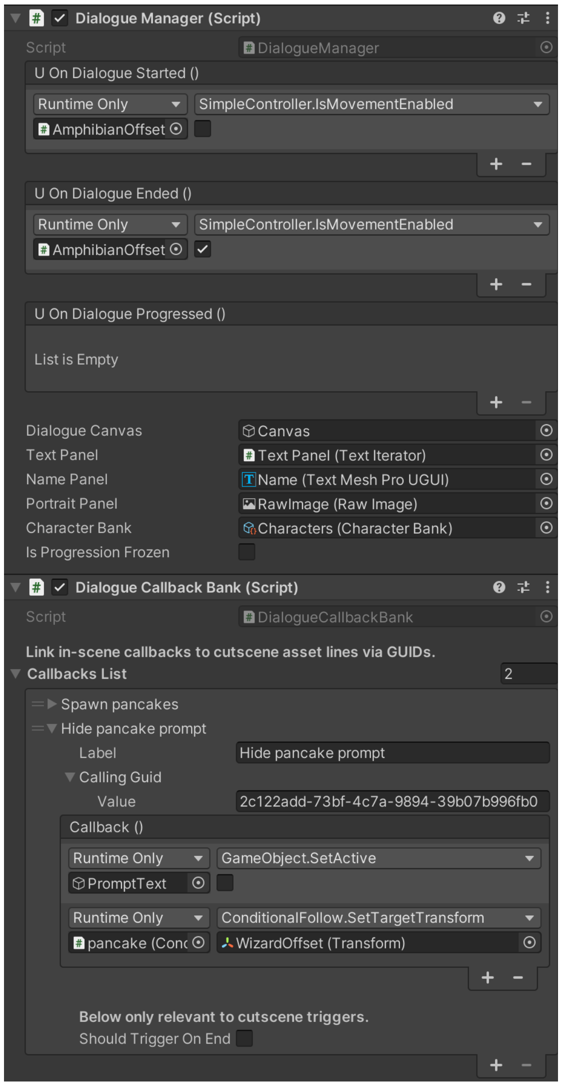
Characters (Character Bank)
Open
Script CharacterBank
Character List 2
Alex
Name Alex
Photo Amphibian
JASON
Name JASON
Photo Wizard

Extracted character bank



Inspector
PART1 (Cutscene)
Open
Script Cutscene
Title PART 1
Exchanges 2
Element 0
Lines 3
Hello there! My name is Alex.
Text Hello there! My name is Alex.
Guid
Value 73e5f390-ee39-4fc2-8746-bb879efb729c
I hope you love pancakes!
Because I sure as heck love 'em myself.
Speaker Alex
Guid
Element 1
Guid
Value e87b7a09-c38e-4496-ba29-afb0d6357045

Parsed cutscene



Dialogue Manager (Script)
Script DialogueManager
On Dialogue Started ()
Runtime Only SimpleController.IsMovementEnabled
AmphibianOffset ☐
On Dialogue Ended ()
Runtime Only SimpleController.IsMovementEnabled
AmphibianOffset ✓
On Dialogue Progressed ()
List is Empty
Dialogue Canvas Canvas
Text Panel Text Panel (Text Iterator)
Name Panel Name (Text Mesh Pro UGUI)
Portrait Panel RawImage (Raw Image)
Character Bank Characters (Character Bank)
Is Progression Frozen ☐
Dialogue Callback Bank (Script)
Script DialogueCallbackBank
Link in-scene callbacks to cutscene asset lines via GUIDs.
Callbacks List 2
Spawn pancakes
Hide pancake prompt
Label Hide pancake prompt
Calling Guid
Value 2c122add-73bf-4c7a-9894-39b07b996fb0
Callback ()
Runtime Only GameObject.SetActive
PromptText ☐
Runtime Only ConditionalFollow.SetTargetTransform
pancake (Conc WizardOffset (Transform)
Below only relevant to cutscene triggers.
Should Trigger On End ☐

Dialogue Manager + callback bank (can be placed on separate game objects)

Markup language overview:

Getting started with DSD is simple. All manuscript files (usually of type **.txt**) on their first line need the word *"BEGIN"* and on the last line the word *"END"*.

Cutscenes begin with the word *"TITLE: "* followed by a title of your choosing.
Speakers within cutscenes are indicated by the word *"NAME: "* followed by who is speaking.

Speaker text is line-break sensitive; speech is fed to the dialogue canvas line-by-line.

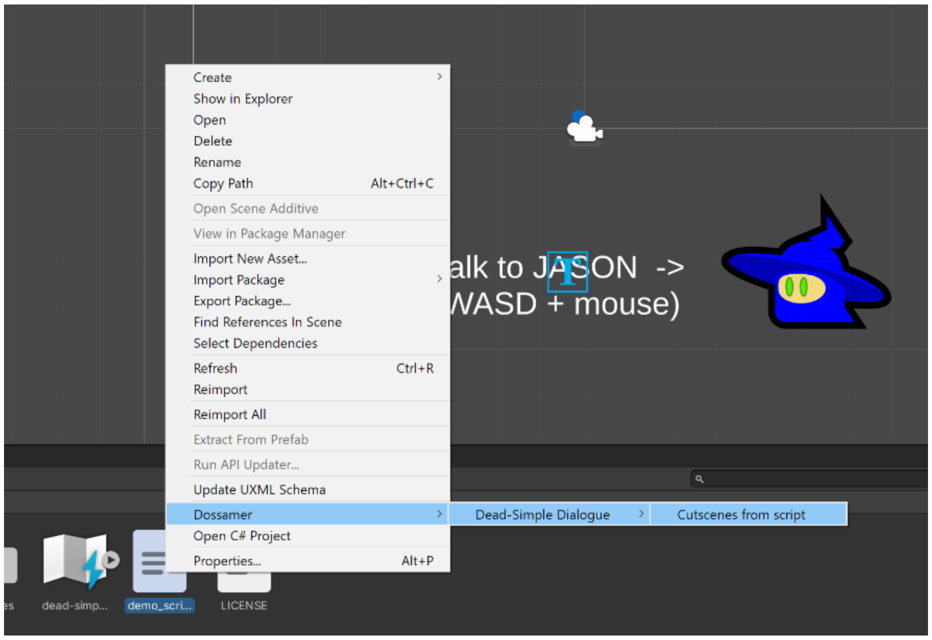
When a speaker is done speaking, on a new line, type "/".
When a cutscene is finished, on a new line, type "/".

If you make mistakes, it's okay! The built-in parser will tell you via an error message in the editor where your manuscript is marked up improperly. Additionally, a demo manuscript is included in the "Demo" folder of the package, as well as below, for your reference.

Demo Markup:

```
BEGIN

TITLE: PART 1

NAME: Alex

Hello there! My name is Alex.
I hope you love pancakes!
Because I sure as heck love 'em myself.
/

NAME: JASON

Oh boi, I hate those!
You suck!
LOL
/
/

TITLE: PART 2

NAME: JASON

AGGHHH screw you!!!
I knew you would sneak me pancakes!
/

NAME: Alex

Teeheehee
/
/

END
```

Turning manuscripts into cutscenes:

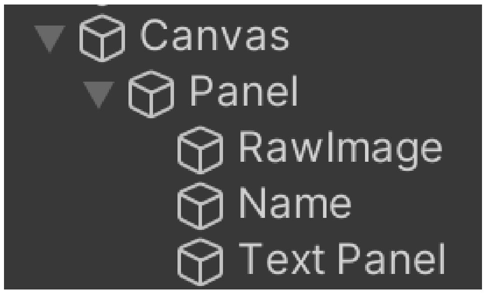


Right-click on your desired manuscript. Navigate to "Dossamer/Dead-Simple Dialogue" and click "Cutscenes from script".

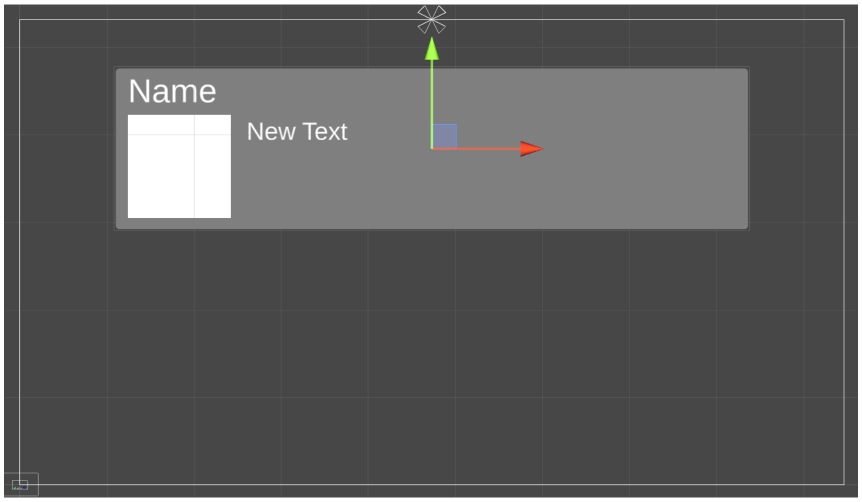
This will create a folder containing cutscene files locally; you'll be asked if it's alright to overwrite a folder of the same name if it already exists in the local directory. (**Be careful!** Overwriting deletes the old folder contents for good, so there's no going back. Overwriting cutscenes will also **break existing cutscene references**, so make backups!)

Setting up the dialogue canvas

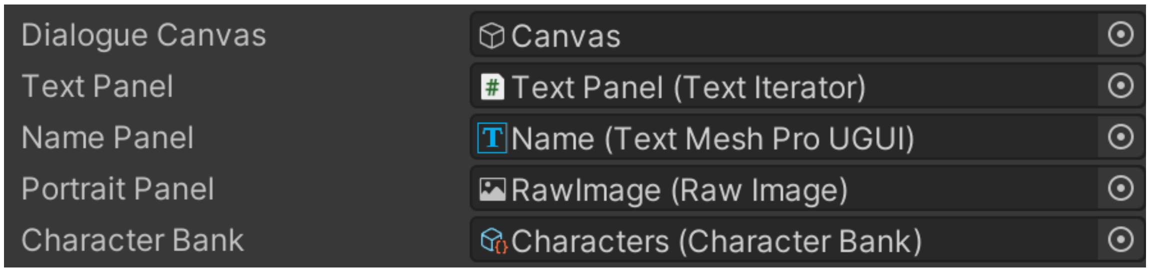
The dialogue canvas where your manuscript lines will be displayed lives in the scene hierarchy, and should look roughly like this:



The actual positioning of the children elements is up to your discretion; in the demo, it's done like this:

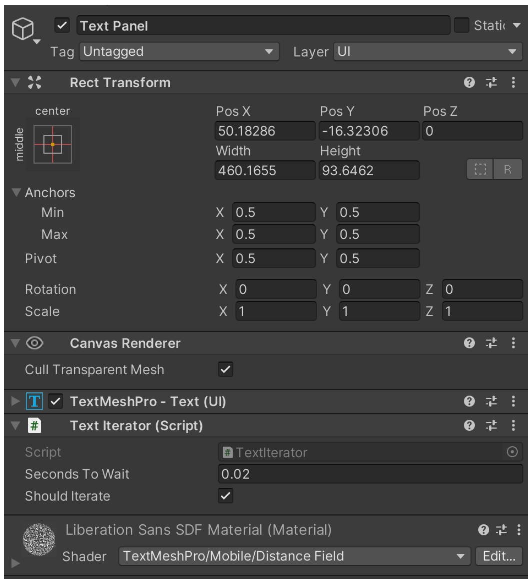


Plug the various canvas objects into your scene's dialogue manager instance, so that it can manipulate the content on-screen.



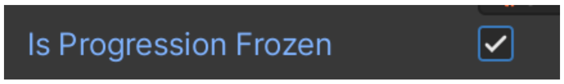
Dialogue Canvas	Canvas
Text Panel	Text Panel (Text Iterator)
Name Panel	Name (Text Mesh Pro UGUI)
Portrait Panel	RawImage (Raw Image)
Character Bank	Characters (Character Bank)

Your Text Panel object will need a Text Iterator component, which handles printing characters to the screen sequentially (this can be disabled in-component).

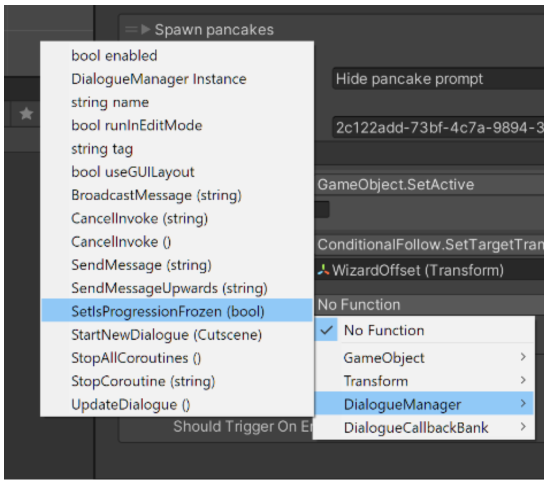


```
✓ Text Panel                                   Stati
Tag Untagged                      Layer UI
Rect Transform
center              Pos X         Pos Y         Pos Z
middle              50.18286      -16.32306     0
                    Width         Height
                    460.1655      93.6462                R
Anchors
  Min               X 0.5         Y 0.5
  Max               X 0.5         Y 0.5
Pivot               X 0.5         Y 0.5
Rotation            X 0           Y 0           Z 0
Scale               X 1           Y 1           Z 1
Canvas Renderer
Cull Transparent Mesh  ✓
✓ TextMeshPro - Text (UI)
Text Iterator (Script)
Script              TextIterator
Seconds To Wait     0.02
Should Iterate      ✓
Liberation Sans SDF Material (Material)
Shader  TextMeshPro/Mobile/Distance Field          Edit...
```

Freezing dialogue progression mid-cutscene



The script-accessible IsProgressionFrozen boolean field lets you disable cutscene playback indefinitely; e.g. because you want to pause the game, or not let dialogue continue til some scripted event in-game completes, like a camera pan. A setter method for it can be invoked from within Unity events.



Dependencies

This package depends both on TextMesh Pro and the New Input System packages being enabled and available in your project. You'll get compile-errors otherwise!

Support

Because this isn't my full-time job, I unfortunately can't make an outright guarantee of technical support; understanding this, I've priced this asset comparatively lower than others on the store. Use this asset at your own risk, under no implication or guarantee of warranty.

I do, however, try to respond as best I can to questions sent to email (contact@dossamer.io) and on the Dossamer discord. Thank you for understanding!

Roadmap (not guaranteed):

- Explicit support for localization
- Smarter scene regeneration via manuscript diffing, ideally that doesn't break references
- Better GUID UX

Special thanks to Team Ham & Cheese for getting me to finish the initial implementation of this system; y’all are great :)