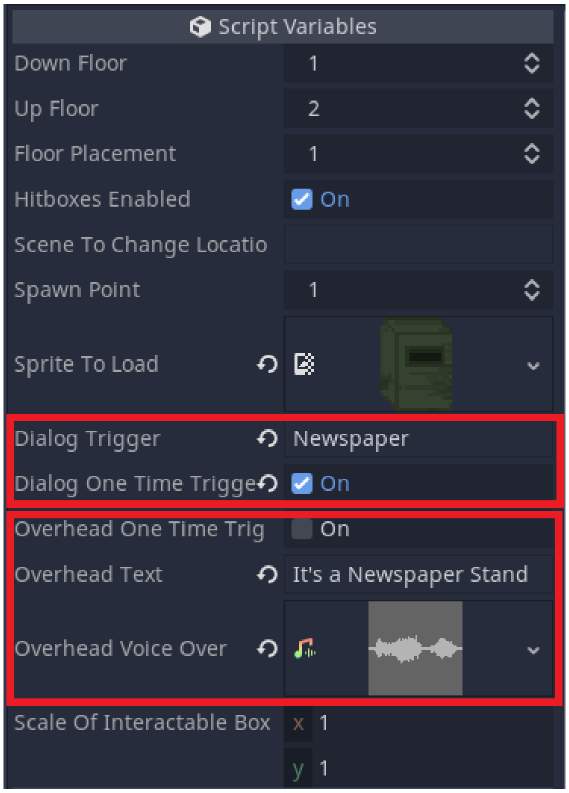
What Lies Beneath

Fiverr Milestone Overview - Milestone #3 Completion

Links

Imgur Link of Gifs: https://imgur.com/a/OIQX7q9
Github Link: https://github.com/ConnorBS/What-Lies-Beneath

Using Dialog



Script Variables	
Down Floor	1
Up Floor	2
Floor Placement	1
Hitboxes Enabled	☑ On
Scene To Change Locatio	
Spawn Point	1
Sprite To Load	
Dialog Trigger	Newspaper
Dialog One Time Trigge	☑ On
Overhead One Time Trig	☐ On
Overhead Text	It's a Newspaper Stand
Overhead Voice Over	
Scale Of Interactable Box	x 1 y 1

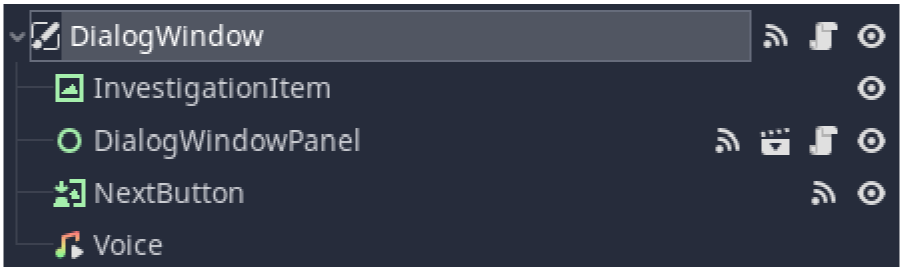
Interactable_Objects now have additional editor options you can configure. This is broken up into Dialog and Overhead

Dialog Trigger looks up the trigger in the CSV Sheet(res://Dialog/Table-Of-Dialog.csv), and pulls the text file associated with it, and overlays the dialog overtop of the screen (outside of the viewport)

Overhead will simply play the text above the player's head. Text speed,voice playback, and visuals are controlled by the player node.

Setting the one time Triggers to true will disable the dialog/overhead after the first play. Until saving is programmed, these will reset on scene loading.

Review of Structure



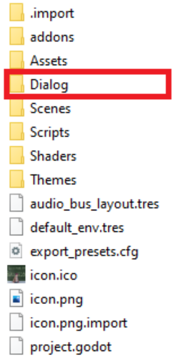
Dialog Window Structure

This scene is broken up into a few different components, The parent (DialogWindow), handles the parsing of the text, and feeds the content to the node DialogWindowPanel.

DialogWindowPanel is going to handle the display of the text, as well as the choices of the player and let's the DialogWindow know, so it can get the next line of text, show the next button, play the voice line, or hide everything but the InvestigationItem for players to look at more closely

This scene shouldn't really be edited unless you want to make global changes, as it will edit all dialog. You can instance off of this though and make your changes there if you want, but I would recommend you making your own commands using the [] commands, and scripting what you want to do when your text processor encounters the string you see in the [] brackets. Current commands will be discussed below.

Dialog Window File Structure

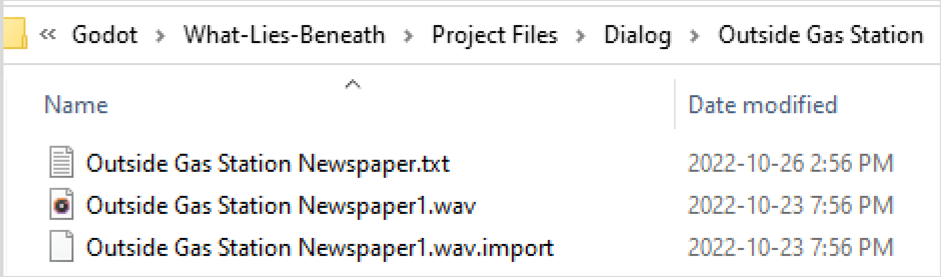


This DIalog Milestone had a new folder added to the project: Dialog
The Dialog File Structure is broken up by some voice over folders like "Journal Pages" and "Overhead" files, and then folders that share the level names.

These folders will contain all your dialog.txt, voice.wav files, and investigation.pngs.

Inside the Level Folders, should have a text file that matches the level name, followed by the trigger, broken by spaces. For audio files or images, include a number that is related to the line number in the text file, without a space.

Example:

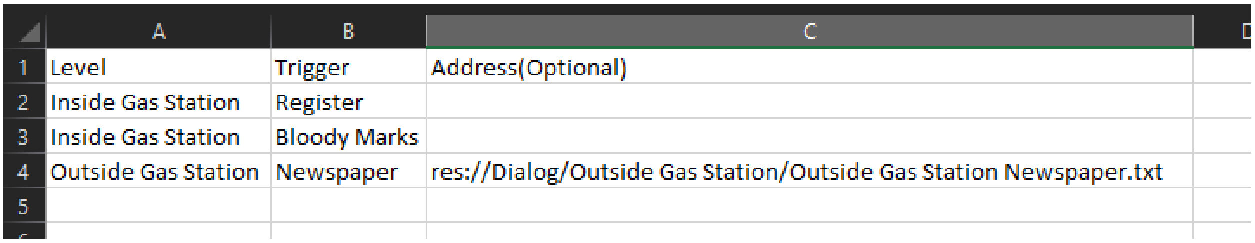


« Godot › What-Lies-Beneath › Project Files › Dialog › Outside Gas Station

Name	Date modified
Outside Gas Station Newspaper.txt	2022-10-26 2:56 PM
Outside Gas Station Newspaper1.wav	2022-10-23 7:56 PM
Outside Gas Station Newspaper1.wav.import	2022-10-23 7:56 PM

In this example, there is no investigation image, but we can see that when the Newspaper trigger is played in the Outside Gas Station level, there is a voice over played for the first line of text.

Table Of Dialog.csv



	A	B	C
1	Level	Trigger	Address(Optional)
2	Inside Gas Station	Register	
3	Inside Gas Station	Bloody Marks	
4	Outside Gas Station	Newspaper	res://Dialog/Outside Gas Station/Outside Gas Station Newspaper.txt
5			

In order to manage all your dialog, the file Table-Of-Dialog.csv is your lookup table. If it’s not in this csv, it doesn’t exist for the script and reports an error in the DialogManager.

This csv helps you keep track of the dialog content you need/have in the game, as well as help reduce lookup times by comparing all file structures/naming conventions in the dialog folder. This also lets you manually define a text file location, in case you want to call some dialog from somewhere unexpected. In the above image, only line 4 has an address. This is an optional field, as the script knows where the file ‘should’ be located, but in case you had some general dialogs you wanted to deploy, you can refer to them directly in a common folder, instead of a level folder structure.

Dialog.txt

Dialog has the ability to write out normal text, but you can add bbcode to add colours to the text, and you add branching dialog paths within the conversation, and you can add checks for previous dialog decisions to write out custom text based off of those decisions.

Regular Text

Text to display is done line by line, by making a new line in the txt file, a new dialog window will play.

The name of the speaker should be at the start of the line followed by a colon. If there is no colon indicating that there isn't a speaker name, the name of the speaker is carried over. If there is no speaker name defined, or there is no text before the colon, the Dialog window will hide the speaker name. This is useful if you wanted to add text like a description of character actions.

PC indicates it's the Player Character, which you can let players name, or just keep the name Ethan in the Global Player information. Otherwise in your dialog, you can just use Ethan and it will disable any dynamic name changes.

Example:

```
PC: I woke up…
But am I still dreaming?
```

BBCode Text

BBCode is enabled for the dialog windows as well, to grant you all the controls that you would expect. BBCode for Godot can be found here. To utilize it in your text file, as we also use square brackets to determine choices, etc., you will need to put quotation marks around your BBCode to indicate to the parser that you want to pass this into BBCode.

Example:

```
PC: Weird, the newspaper in here is dated "[color=red]"1993"[/color]". I guess it's not used all that much.
```

Making Choices

Making choices will be it’s own dialog window, and should be it’s own line. You can assign a speaker with the choice, but the choices are determined by numbers, and should start with 1.

Choices are saved with the level and trigger, and the order of how the choices appear in the text document. It will be important to keep the order of choices straight in your end/design documentation if you plan on referring to them later on. It’s highly recommended that you do not put too many choices in one dialog to maintain easy tracking.

Once the choices are laid out, you can then indicate the outputs for each choice by making a separate line with the choice number in square brackets. If you want to provide the same text to each decision, or tie it all back together, you can then use the command RESUME in square brackets, and dialog will resume as per normal.

Example:

```
PC: Bloody Marks are left on the wall
: [1] Leave it alone [2] Taste it to see if it's blood
[1]
PC: Let's leave it alone...
[2]
PC: It tastes metallic...
Ugh, I don't feel so good
[RESUME]
I think I need to get out of here
```

Recalling Choices

To refer to another choice made in another dialog,use square brackets and list your location, target,dialog position,dialog choice. Dialog position starts the count at 0, and each additional choice is +1 in the text file. The choice number should match the number choice provided in the choice text.

Using the above example for and you wanted to see if they tasted blood, you will need to know the Level Name and the Trigger in the csv, but you can then call and check by:

```
[Inside Gas Station, Bloody Marks,0,2]
```

If you want to check additional choices, just continue adding to this array.

If you wanted to display text if none of the above requirements match, you can use ELSE in square brackets

Example:

[Inside Gas Station, Bloody Marks,0,2, Water Temple, Bloody Shirt,4,2]
PC: Why do I keep tasting things that are clearly blood? Something is wrong with me
PC: Oh Well, I am just going to taste this just in case it isn't.
PC: Nope, it's definitely blood.
[ELSE]
PC: It's clearly blood, I am going to leave it alone

Milestones

Player Movement Milestone (Commit 07baaef in Movement)

- ☑ ~~General Movement~~
  - ☑ ~~Walking~~
  - ☑ ~~Running (Hold-Shift)~~
  - ☑ ~~Hidden Stamina Bar~~
  - ☑ ~~Climb Ladder~~
  - ☑ ~~Fall/Drop~~
- ☑ ~~Interactions~~
  - ☑ ~~Push/Pull Box/Climb Box~~
  - ☑ ~~Kneeling (Looting/Interacting)~~
  - ☑ ~~Interactable Object (Shimmer)~~
- ☑ ~~Basic Combat~~
  - ☑ ~~Aiming (across the walkable axis)~~
  - ☑ ~~While Aiming you can only Walking~~
  - ☑ ~~Button to pull out gun~~
- ☑ ~~Game Camera~~
  - ☑ ~~Dynamic Camera Primarily on Character~~
- ☑ ~~Additional Features~~
  - ☑ ~~Splashscreen~~
  - ☑ ~~Project Image~~
  - ☑ ~~Level Change~~

Menu System Milestone(Commit f218118 in Menu)

- ☑ ~~Silent hill style (scrolling inventory)~~
    - ☑ ~~Options to: Use, Reload, Equip, Remove~~
    - ☑ ~~Gun Equipment Slot~~
    - ☑ ~~Still Screenshot of the Game in the top left with colour overlay indicating health~~
- ☑ ~~Map option - Placeholder Screen~~
- ☑ ~~Memo - Journal Pages~~
    - ☑ ~~Left right over Journal Objects~~
- ☑ ~~Important Objects~~
    - ☑ ~~Shows Keys~~
    - ☑ ~~Shows Map Fragments~~
- ☑ ~~Options with: Music and Sound Control~~
- ☑ ~~Global Player State~~
    - ☑ ~~Health~~
    - ☑ ~~Equipped Item~~
    - ☑ ~~Inventory~~
    - ☑ ~~Important Inventory Items~~
    - ☑ ~~Map Fragments~~
    - ☑ ~~Location~~

Dialogue Milestone (Commit 8f60f51 in Dialog)

- ☑ ~~Dialog with branching options as well as object Inspection~~
- ☑ ~~Interact-able Objects in Level where it triggers dialogue~~
- ☑ ~~Interact-able Objects in Level where it triggers text above the player’s head~~
- ☑ ~~Save Dialog Choice Tree in Global Player State~~
- ☑ ~~Additional Features~~
    - ☑ ~~Animation for scene changes~~
    - ☑ ~~Voice Over for Journal Entries~~

Map System Milestone

- ☐ Collecting fragments of maps to reveal more of the map (accessible via the menu system)
- ☐ Metroid Style Map where map highlights which location you are in
    - ☐ Track if you have been in the space
    - ☐ Track if the door is locked (show by X)

Combat Milestone

- □ Player Combat
    - □ Shooting Gun While Aiming
    - □ Melee Animation
    - □ Check on Hit/Provide Damage
    - □ Taking damage
    - □ HP Recovery (Syringe)
    - □ Game Restart
- □ Player Inventory
    - □ Check Equipped Item (Gun)
    - □ Provide Damage on Check

Enemies Milestone / Save Point Milestone

- □ Enemy AI Types Created
    - □ The Lost – The remnants of a person. Doesn't focus on the player and shuffles on its own, until it sees the player and goes straight to the player. Takes 2-3 hits
    - □ Rabid Dog – Sprints to player on line of sight.
- □ Enemy Interactions
    - □ On Player Touch, Damages Player
    - □ Attack Animations
        - □ Attack hitboxes
        - □ Critical hitbox damage
    - □ On Kill, Enemy is destroyed and removed
    - □ On Entry, Enemy States are stored and loaded back

- □ Object in world where the character can save
- □ Start Menu
    - □ New Game
    - □ Load Game
    - □ Quit
- □ On Death, Load Last Save