HOCHSCHULE
ANSBACH

Fakultät Medien

Easy Dialogue für Unreal Engine

Projektdokumentation im Studiengang Visualisierung und Interaktion in digitalen Medien

vorgelegt von

Danny Karim Sehili

Matrikelnummer 00154863

Contents

1 Easy Dialogue for Unreal Engine

EasyDialogue is a Plugin developed for *Unreal Engine*, aiming to assist developers in creating games with in-game dialogue. It provides an architecture with many tools to speed up the process of designing dialogues with NPCs inside the game world, directed towards developers of dialogue-heavy games, providing a solid base to design dialogues. The Plugin has been developed in Unreal Engine 5.5.1 and is hosted on GitHub (https://github.com/Sehilius/EasyDialogue-UE5). It is important to note that the plugin has been built based on another plugin called *Generic Graph* (https://github.com/jinyuliao/GenericGraph). However, because *Generic Graphs* is another third party plugin and not available on the Epic *Unreal Marketplace*, *Easy Dialogue* could not be built with a dependency on *Generic Graphs*, since the engine does not allow this. This means that users have to install *Generic Graphs* manually.
For its features, *Easy Dialogue* lets developers:

- create dialogues through a graph structure
- set the speaker's name
- set the dialogue box text
- use different text styles (font, colored, small)
- specify pauses of customizable length after words
- set music dependent on the dialogue
- specify voice lines or sounds for every dialogue box
- a typewriter animation style
- specify sounds for characters to be played when interacted with
- specify answers the player can give to a dialogue box
- specify text styles for hovered and unhovered answers
- create branching dialogue structures based on the player's answer
- bind events to dialogue boxes
- switch the dialogue of a character based on events
- set the sprites for the dialogue box, names, answers, answer backgrounds, and an icon that appears when the text is finished
- set the text style for answers

- set sounds to be played wen proceeding in dialogue, moving between answers, selecting an answers, and toggling fast forward
- set the dialogue assets with code

For the typewriter, the user can set the speed and the sounds the typewriter should play for every typed character. This supports multiple sounds to be played randomly. Additionally, the typewriter comes with a fast forward function that speeds up the text typing and automatically proceeds to the next dialogue box.

While the tool has not been conceptualized as a tool for games where the player character can talk, it definitely provides enough control and tools to still be used as such.
It is recommended to only use the small text by itself, since when combining small and bigger text and using the small text as the first in a line, there will be jumping in text as soon as bigger text appears. This is an issue which,unfortunately, could not be fixed through *Unreal Engines*'s UMG system.

2 Development and Concept

Throughout the development, a lot of things have been taken into account to deliver an easy experience. To make plugin classes clear, all class names begin with "EasyDialogue".
The plugin consists of the following classes:

- WBP_EasyDialogueBox: the dialogue box widget
- WBP_EasyDialogueAnswers: the widget containing the answers
- WBP_EasyDialogueAnswersItem: the widget representing a singular answer
- BP_EasyDialogueBox: graph node containing dialogue box settings
- BP_EasyDialogueEdge: graph edge connecting nodes based on answers
- AC_EasyDialogueComponent_Player: the component to put on the player pawn
- AC_EasyDialogueComponent_NPC: the component to put on interactable objects
- BP_EasyDialogueTextAnimator: class to animate text
- BP_EasyDialogueAudioHandler: class to play music and sounds

- BPDA_EasyDialogueSettings: settings to setup the dialogue containing sprites, sounds, and the style data table
- BPDA_EasyDialogueGeneralSettings: contains the current settings
- BPFL_EasyDialogueHelperFunctions: contains static function used within the plugin architecture
- E_EasyDialogueTextAnimationStyle: enum to select the text animation
- E_EasyDialogueTextStyle: enum to select the text style
- F_EasyDialogueTextStyle: struct to create the dialogue text with different styles

The plugin is based on *Unreal Engine*'s RichTextBlock class, which allows to use different styles from a data table through the use of macros. EasyDialogue simplifies this process through a struct class, that contains a text style enum and the text. An array of this struct will then be combined into the final text, automatically generating the macros based on the selected enum. This had to also be specifically programmed to work with the text animator, only placing macros to the currently generated text.
When the game starts with a AC_EasyDialogueComponent_Player on the player pawn, it will create the WBP_EasyDialogueBox. From here, the
WBP_EasyDialogueBox will be spawning the BP_EasyDialogueTextAnimator, which is necessary for the typewriter animation. WBP_EasyDialogueBox will be made visible at the start of a dialogue and hidden when it ends.
Inspired by many other games, a "proceed icon" has also been implemented, which appears as soon as the text finished animating, signifying that the player can press the proceed button to proceed to the next text box.
To create a tool that is easy to use, the GenericGraph plugin has been utilized, which allows the creation of graphs of node and edge classes. BP_DialogueBox has been used for the graph nodes and BP_DialogueEdge for the edges connecting the nodes. While nodes represent the actual information to create the dialogue box widget, the edge class has been created for branching dialogues, where the player can give an answer.
WBP_EasyDialogueAnswers handles the creation, navigation and selection of player answers. WBP_EasyDialogueAnswersItem are the separate answer objects, containing functions to change their appearance based on their selection.
Data assets based on the BPDA_EasyDialogueSettings contain the dialogue settings, which include the sounds for proceeding, giving an answer, moving between answers, enabling fast forward, and disabling fast forward. They also contain the sprites to be used for the dialogue widgets, which are the sprites for the dialogue

name box, dialogue text box, proceed icon, hovered answer, normal answer, and the answer's background. Lastly, they also contain the style data table for the fonts. The dialogue settings asset to be used is specified in the data asset based on the BPDA_EasyDialogueGeneralSettings class, called DA_EasyDialogueGeneralSettings. All widgets read the settings sprites and style data table and are setup in their "On Pre-Construct" events, which allows displaying them within the editor.
The level's music will also be played through the BP_EasyDialogueAudioHandler. In the setup process, users have to place it inside the level and can then set the level music inside the placed instance. For other sounds, the BP_EasyDialogueAudioHandler either reads the sounds to play directly from the settings or receives the data as inputs.
BP_EasyDialogueComponent_Player is the component to be added to the player pawn. This component does need to be set up within the class it has been added to. It requires receiving bindings to disable player movement during dialogue and reenable it after it's finished. It also provides all events to control dialogue-specific things like interacting, proceeding, navigating between answers, locking in an answer, and toggling fast forward.
BP_EasyDialogueComponent_NPC is the component added to all objects that are interactable. Here, one can specify the current dialogue asset to be played when interacted with. Additionally, one can specify "DialogueStartSounds" that play when the interaction starts. This supports multiple sounds which will be chosen randomly.
WBP_EasyDialogueBox is the central class of the entire concept, possessing many events to call other events on other classes. It sets up the dialogue box based on the nodes and edges of the dialogue asset graphs. It also switches between dialogue boxes and handles the selection of the next one based on player answers.
During development, the problem that there might be issues with contrast between answer text and sprite has been noticed. This has been countered by defining separate text styles for the answer that is currently hovered and answers that are not. It was also needed to disable the fast forward as soon as an answer is needed. It was also needed to disabled fast forward after a dialogue ends, so the next one does not start fast forwarded. Scenarios like not providing a speaker name for the dialogue, such for using the dialogue as the game itself telling the player something, have also been considered. This led to the functionality of automatically hiding the name box if no name is given.
The simple use of components, being able to set assets within the
BPDA_EasyDialogueSettings, spawning classes through other classes, using data tables, and giving the opportunity to bind events and set up the controls like one wants contributes a lot to the freedom the tool has to offer and aims for high customizability. The user is also not forced to provide every asset, since the code

has been built to run also without asset references, not returning any errors.

3 Demo Map

Easy Dialogue comes with a simple demo map, where various features of the plugin are demonstrated. To load this Map, *Unreal Engine*'s third-person template is required. The demo map uses premade settings with sprites and sounds and style data table. All sprites used were self-created. The map contains three NPCs the player character can interact with, using classes from the third person template. These NPCs are "Mysterious Old Man", "Carlos" and "Vitta". These NPCs represent typical NPCs in RPG games, where the player is thrown into a large world that is suffering great danger.
This information is given by the mysterious old man, who denies the player character's claim of being the hero to save the world. The player character does not speak through a dialogue box, but rather "off screen", similar to how it's done in other RPG games like *Hollow Knight* or *Pokémon*. The dialogue box of the old man highlights the word "hero" in blue, signalizing the importance of the player character. Here, it is also presented how one can change the music mid dialogue, as it switches to a more dramatic song when the dialogue gets more dramatic.
Carlos presents the feature of player answers, asking the player if they want to buy a potion. The player can answer this with "Sure!" or "I think I'll pass...", changing the following dialogue with Carlos based on their answer.
Vitta a demonstration of how developers using *Easy Dialogue* can bind events to a dialogue box, resulting in the dialogue box firing the assigned event when being displayed. It also gives an example of how the dialogue of a character can be changed when certain conditions are met. Vitta lost her ball and cannot find it. When the player comes up to her, she uses the opportunity to give him a challenge, which is to find her ball. This fires the event of spawning the ball inside the map. After the player takes the ball, Vitta's dialogue asset will be changed, reacting to the player finding the ball when the player comes back to talk to her. Vitta's dialogue also highlights the red ball with the color red, to signifiy the importance of the object.
Carlos and Vitta also present the use of the small text style, symbolizing a quiet mumbling more done to themselves instead of towards the player.
All characters have a dialogue start sound. The mysterious old man also demonstrates the usage of voice lines for a specific dialogue box. All NPC sounds have been self-recorded.

4 Installation and Setup Guide

A detailed guide on how to install and setup the plugin is given in the README on the GitHub page of the project. Additional information about the plugin is also to be found there.

5 Sources

Font: https://www.dafont.com/spicy-style.font

Sounds: https://pixabay.com/de/sound-effects/selection-sounds-73225/

Music: https://www.chosic.com/download-audio/28063/ & https://www.chosic.com/download-audio/26012/