Brain Game - Documentation

## – What did you research during the implementation? Provide a link to the resources you used.

During the implementation, I researched using Unity's UI Toolkit to design the main menu screen and dialogue boxes. I explored incorporating sound effects, such as typewriter effects, to enhance gameplay immersion. Additionally, I delved into dynamically triggering dialogues in Unity using C#. My research primarily relied on Unity documentation for scripting with coroutines and managing UI interactions, with a specific focus on handling trigger events and orchestrating dialogue sequences effectively.

<https://docs.unity3d.com/Manual/UIElements.html>

<https://www.youtube.com/watch?v=BG6NCgkbbiA>

<https://learn.unity.com/tutorial/ui-toolkit-first-steps>

Various Open-source AI tools

<https://www.studytonight.com/game-development-in-2D/audio-in-unity>

<https://medium.com/@victormct/how-to-play-sound-effects-in-unity-a-step-by-step-guide-73e5e1d8e100>

<https://learn.unity.com/tutorial/working-with-audio-components-2019-3>

## – What did you try? What worked? What did not work?

I attempted to use the dialogue scripts to trigger specific dialogues based on player interactions. Initially, I set up the dialogue manager and dialogue data structures to handle different sets of dialogues for various game states or scenes. However, I encountered issues with ensuring that the correct dialogue would display at the intended trigger points.

What worked initially was setting up the basic structure of dialogue management and integrating it into the game flow. This involved instantiating dialogue UI elements and starting coroutines to display text over time, synchronized with audio effects.

What did not work smoothly was ensuring that the dialogue triggered corresponded accurately to the specific trigger points in the game. I faced challenges with correctly passing and managing dialogue indices or references, leading to incorrect dialogues being displayed or no dialogue at all in some cases.

To address these issues, I refined how dialogue indices were assigned and passed between scripts, ensuring that each trigger point accurately referenced the intended dialogue within the data structure. This process involved debugging, refining coroutine timings, and testing extensively to ensure that dialogues displayed correctly based on player interactions and game events.

## – What did you learn?

I learned valuable lessons in C# scripting and leveraging Unity's UI Toolkit to manipulate UI elements dynamically. Specifically, I enhanced the DialogueTrigger script to instantiate a dialogue menu prefab when the player collides with a trigger zone. This script effectively retrieves dialogue content from a DialogueManager based on an index specified in the Inspector, ensuring dialogues trigger correctly. Through this process, I gained insights into the significance of organized event handling and the effective use of coroutines in Unity for managing dialogue sequences. Mastering the instantiation of UI elements on-the-fly and managing user input during dialogue playback proved essential to achieving smooth gameplay interactions.

## – How can you apply what you learned to future games?

## Looking forward, my improved proficiency in C# scripting will significantly enhance my ability to develop more robust and accessible game functionalities. By leveraging these enhanced skills, I aim to implement more sophisticated interactive dialogue systems in future games, enriching player engagement and deepening narrative immersion. This involves further refining trigger-based events and optimizing coroutine usage to ensure seamless transitions between dialogues and gameplay events.

## Professionally, I have encountered challenges in managing asynchronous events and synchronizing UI elements effectively with gameplay logic. To address these challenges, I have refined my approach to coroutine management, conducted thorough debugging using Unity's console, and iteratively tested gameplay interactions. These efforts are crucial in delivering seamless player experiences and improving overall game quality.

## By continuing to apply these techniques and methodologies, I aim to elevate the standard of my game development projects, ensuring they are not only technically sound but also compelling and immersive for players.

## – In addition, what did you find most challenging professionally? How did you overcome it?

Professionally, one of the most challenging aspects was initially grappling with how to effectively utilize Unity's UI Toolkit elements and structuring scripts to seamlessly display dialogue in a narrative-driven manner. Accessing and manipulating UI Toolkit elements posed a learning curve, requiring me to understand the intricacies of UI element management within Unity's framework.

**Research and Learning**:

I dedicated time to thoroughly research Unity's UI Toolkit documentation and tutorials. This helped me grasp essential concepts such as accessing UI elements by name or type, updating text dynamically, and managing UI state changes.

**Script Structuring**:

I focused on designing well-structured scripts that modularly handled dialogue sequences. This involved breaking down the dialogue display process into manageable steps, using coroutines for text animation, and implementing event-driven triggers to synchronize dialogue with gameplay events.

By navigating these challenges and leveraging the learning experiences gained, I've significantly strengthened my skill set in UI development and narrative scripting within Unity. These efforts have not only enhanced my professional capabilities but also enriched the quality and immersive storytelling potential of my game projects.