

Wstęp:

W ramach mojego projektu zaliczeniowego użyto gotowego już projektu, który można znaleźć pod [tym linkiem](#). Jako, że był to już gotowy projekt, trochę czasu zostało poświęcone na przeczytanie istniejącego już kodu i wydzielenie miejsc w których można wtłoczyć własne rozwiązania, oraz możliwość łatwego wycięcia istniejącego już w projekcie SoundManager'a.

Wszystkie stworzone lub dodane pliki znajdują się w folderze „Zaliczenie Dźwięk”. Wydzielone tam zostały również osobne foldery na inne rodzaje plików i komponentów używanych w Unity, aby zachować porządek.

Znajduje się tam również przygotowana scena „MainGame”.

Zamierzenia i uzasadnienie:

Głównym zamierzeniem było dodanie pojedynczego skryptu SoundManager, w którym odpalane byłyby wszystkie dźwięki występujące w grze, zależnie od potrzeby. Stworzenie jednej instancji klasy zarządzającej odpalaniem dźwięków pozwalałoby na możliwość odniesienia się do niej z każdego miejsca w projekcie. Dzięki temu unika się potrzeby przekazywania referencji do SoundManagera w każdym miejscu w kodzie w którym należy odpalić jakiś dźwięk.

SoundManager pobiera dźwięki oraz konfigurację z podpiętego do niego ScriptableObject. Dzięki temu w bardzo przystępny sposób jest możliwość podmieniania dźwięków lub tworzenia różnych konfiguracji, zależnie od potrzeb.

Do SoundManagera podpięte zostały również trzy obiekty z komponentem AudioSource. Po kolei są to:

- SfxAudioSource
- MusicAudioSource
- UIAudioSource

Jako, że gra dzieje się na niewielkim obszarze w świecie 2D, nie ma potrzeby rozmieszczania AudioSource'ów w świecie gry i można je trzymać w jednym głównym obiekcie. W ten sposób można bardzo łatwo przygotować prefab zawierający wszystkie potrzebne elementy, do którego należy później jedynie podpiąć odpowiednie dźwięki za pomocą ScriptableObject.

Jednym z zamierzeń było dodanie AudioMixera, który posiadałby trzy grupy zgodne z wymienionymi powyżej AudioSource'ami. Następnie, używając snapshotów, można dzięki temu obniżyć głośność efektów dźwiękowych i muzyki podczas pauzy w grze, pozostawiając dźwięki UI niezmiennymi.

Kolejnym zamierzeniem było dodanie suwaka głośności w panelu pauzy, aby użytkownik był w stanie przyciszyć lub podgłośnić dźwięk zależnie od potrzeb. Do realizacji miał posłużyć AudioMixer i wystawiony parametr głównej głośności.

Jednym z mniejszych zamierzeń było dodanie dźwięku do akcji najechania na guzik w menu pauzy.

Co udało się zrealizować, napotkane problemy:

Udało się zrealizować wszystko co zostało wymienione w sekcji Zamierzenia i uzasadnienie.

Jedynym problemem napotkanym podczas implementacji rozwiązań było płynne przechodzenie pomiędzy snapshotami w AudioMixerze.

Funkcja `TransitionTo()` wywoływana na snapshotcie nie działała w intuicyjny sposób. Jeżeli zmienna podawana jako parametr funkcji była zbyt wysoka, choć przejście było płynne, to nie wykonywało się ono w pełni. Dopiero po ustawieniu wartości parametru na bardzo niską (0.01), przejście wykonuje się w pełni, lecz jest bardzo nagłe.

Przy wielokrotnym naciskaniu klawisza Esc jest to bardzo słyszalne i może być nieprzyjemne dla użytkownika.