**Dokumentacja techniczna gry**

W tym miejscu będziemy prezentować programy stworzone na potrzeby projektowania gry, wraz z szczegółowym wyjaśnieniem ich działania. Każdy z programów będzie omówiony pod kątem jego funkcjonalności i wykorzystania w grze.

Kolejność omawianych programów jest losowa

MusicPlayerScript.cs – jest to skrypt, który ma za zadanie zarządzać odtwarzaniem muzyki i regulacją dźwięku w grze.

Ustawia dźwięk na wartość domyślną



Start:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

ObjectMusic – znajduje obiekt z tagiem music i przypisuje go do zmiennej.

AudioSource – pobiera utworzony w grze komponent „AudioSource” i przypisuje go do zmiennej

PlayerPrefs – odpowiada za zapisywanie wartości dźwięku w pramięci gry.

AudioSource.volume i volumeSlider.value – ustawine są wartości na bazie zmiennej MusicVolume

Update:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Aktualizacja głośności "AudioSource" na podstawie wartości "MusicVolume", która synchronizuje wartość suwaka "volumeSlider".   
  
Następnie zapisuje tę wartość do PlayerPrefs(czyli, jak zresetujemy grę, wartość będzie taka jak poprzednio), a debug, służyły do śledzenia wartości (podczas testów)

Kiedy była już obsługa suwaka, potrzebny był skrypt, który pozwoliłby odtwarzać muzykę pomiędzy różnymi scenami (i żeby utwór nie uruchamiał się przy przejściu na kolejną scenę od nowa)

DoNotDestoy.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten skrypt ma zapobiegać „niszczeniu”(niszczenie, ogólnie wywołuje się przy przejściu na kolejną scenę, czyli żaden obiekt z poprzedniej sceny nie przenosi się do kolejnej) obiektu z tagiem „music” przy zmianie sceny, o ile nie istnieje już obiekt z takim tagiem. Jeżeli pojawi się obiekt nadmiarowy, to zostanie zniszczony, jeżeli jest tylko jeden, zostaje on zachowany przy zmianie sceny.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten fragment jest wykorzystywany do zrealizowania muzyki na scenie Areny, sprawdza scenę, jeżeli jest Areną, usuwa obiekt, który ma tag = „music”, następnie szuka, czy w danej scenie, jakiś obiekt ma tag „BattleMusic”, jeśli tak to uruchamia go.

LoadNextSceneScript.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Bardzo prosty skrypt, który ma za zadanie, uruchomić konkretną scenę, np. kiedy przypisujemy akcje do przycisku (weaponsmith, arena itp.)

MainMenu.cs

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Skrypt, który bardzo szybko został zastąpiony przez LoadNextSceneScript.cs wykorzystywany był w początkowej fazie projektu, do przechodzenia do następnej sceny

CreateTimeStamp.cs – skrypt który ma za zadanie mierzyć czas gry

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Ten fragment kodu tworzy znacznik czasu przy uruchomieniu obiektu w grze i zapisuje go do pliki time.txt

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Druga część kodu pokazuje implemetracje metody Create\_time

„DateTime now” – zapisuje do zmiennej aktualny czas

long currentTime = dateTimeOffset.ToUnixTimeSeconds(); - polega na konwertowaniu czasu na liczbę sekund

ostatnia linijka z tego fragmentu kodu pokazuje zapisanie czasu do pliku tekstowego

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Ostatnia część tego skryptu pozwala na zapisanie znacznika czasu i późniejsze obliczenie oraz wyświetlanie czasu który upłynął od momentu tego zapisu.

LeaderBoard.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten skrypt opisuje strukturę tabeli w bazie danych

Warto zwrócić uwagę na pola: [Serializable] i [NonSerialized] są to pola które mają lub nie mają być przechwycone/przesyłane przez sieć.

SaveLoadFile.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Save\_to\_file – zapisuje tekst do określonego pliku   
Load\_to\_file – odczytuje tekst z określonego pliku

Skrypt jest wykorzystywany do interakcji z plikami podczas gry

ItemObj.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Skrypt ten ma za zadanie ustawić wartość pola tekstowego, kiedy obiekt zaczyna być aktywny i aktualizować go kiedy pojawiają się zmiany (oczywiście taką jaką przyjmuje wartość intValue)

ShowMoney.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Skrypt odczytuje wartość z pliku user\_data.json i wyświetla walutę użytkownika na ekranie za pomocą TextMesh Pro (pola tekstowego w Unity)

MessageHandler.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten skrypt przyjmuje ciąg znaków w formacie json, parsuje ten ciąg do JObject, wyciąga wartość z pola „message” i zwraca wartość jako ciąg znaków

Serwer: Program.cs

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Metoda „Main” uruchamia nasłuchiwanie na porcie: 55000

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Kiedy mamy już ustalony port, definiuje lokalny adres IP „127.0.1.1” do nasłuchiwania połączeń, następnie tworzy socet tcp/ip i powiązuje je z adresem IP i portem. Później ustala, że maksymalna ilość połączeń jest równa 10.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

W tej pętli serwer oczekuje na połączenie się klienta, kiedy połączenie zostanie zaakceptowane, odbiera dane od klienta i zapisuje w buforze. Przy użyciu MessageHandlera przetwarza te dane. Później tworzy odpowiedź w formacie json przy użyciu metody „json\_responder”. Następnie wysyła odpowiedź do klienta i zamyka połączenie w obu kierunkach, jeżeli w trakcie tego procesu wystąpi jakiś błąd, zostanie przechwycony przez obsługę błędów i go wypisze.

Handler.cs – jest to kod, który ma za zadanie obsługiwać różne rodzaje wiadomości przesłanych do serwera. Przyjmuje wiadomości, analizuje którą pętle spełnia i zwraca odpowiedź.

Rozpatrzone jest wiele możliwości takich jak m.in odczytywanie użytkowników z bazy danych, sprawdzanie czy użytkownik istnieje w tej bazie danych, sprawdza czy istnieje już nickname postaci. Handler w tym samym kodzie, również obsługuje przypadki w których klient wysyła dane do serwera/bazy danych m.in. dodanie nowej postaci do bazy danych, dodanie nowych danych logowania użytkownika do bazy danych

Obsługa błędów jest również zdefiniowana, czyli jeżeli nie będzie w stanie przetworzyć danej wiadomości, zwróci komunikat o błędzie

Serwer/db:

Tutaj będzie będą pokazane wszystkie funkcje związane operacjami na bazie danych.

Definiowanie tabeli w bazie danych:

User\_data.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten kod definiuje tabele w których w celu przechowywania danych użytkowników.

User\_credentials.cs

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Ten kod definiuje tabelę z nazwą użytkownika(hashowany) i hasłem użytkownika(również hashowanym)

Score\_scructur.cs

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Czcionka

Opis wygenerowany automatycznie

Ten kod definiuje tabelę, która jest wykorzystywana do leaderboard’a

Wyszukiwanie użytkownika w bazie danych

Find\_user\_cred.cs

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, Oprogramowanie graficzne

Opis wygenerowany automatycznie

Ten kod jest używany do sprawdzania, czy użytkownik i hasło istnieje w bazie danych. Metoda finduser łączy się z bazą danych i przeszukuje tabelę „Users\_credentials” w poszukiwaniu pasującego użytkownika i zwraca „true” albo „false”

Hashowanie

Hashing.cs

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Ten kod ma za zadanie przekształcić dane wpisane przez użytkownika na SHA-256 w celu bezpiecznego przychowania tych danych w bazie danych.

Odczytywanie danych

Reader.cs

Zawiera metody do odczytywania danych z bazy danych, a następnie zwracane są jako json, co ułatwia dalsze przetwarzanie i wyświetlanie tych danych.

Read.scoreboard – odczytuje dane wszystkich graczy i zwraca listę (nick, poziom doświadczenia, zwycięstwa i czas gry)

Leaderboard\_entry – przeszukuje bazę danych, aby znaleźć konkretnego gracza po jego nicku i zwraca jego dane

Read\_user\_data – odczytuje wszystkie dane gracza i ponownie zwraca listę (ID, nickname, level, itemlist…)

Read\_user\_info – szuka w bazie konkretnego gracza po jego nazwie użytkownika, jeżeli dane są to zwraca je, jeżeli nie wyświetla komunikat o braku danych.