*Dokumentacja motorola-sokoban*

Projekt został wykonany w Unity 2020.2.0f1. Poniżej znajduje się bezpośredni link do pobrania silnika graficznego

https://download.unity3d.com/download\_unity/3721df5a8b28/Windows64EditorInstaller/UnitySetup64-2020.2.0f1.exe

Po zainstalowaniu należy otworzyć program Unity. W zakładce File trzeba wybrać Open Project.. a następnie wybrać folder z rozpakowanym projektem gry

Aby uruchomić testy automatyczne, należy otworzyć TestRunner’a. Wybierz zakładkę Window - > Test Runner. Następnie upewnij się, że zakładka PlayMode jest zaznaczona, a następnie kliknij Run All lub kliknij lewym przyciskiem myszy na konkretny test, który chcesz uruchomić.

Do dyspozycji są następujące testy sprawdzające najbardziej elementarne mechaniki rozgrywki

BoxTest:

-BoxHasCorrectScriptForLevelLoaderScene - sprawdza czy Skrzynka posiada właściwy skrypt zależności

BoxHasEmptyTargetTransformForLevelLoaderScene - sprawdza czy Skrzynka ma pustą domyślną pozycję. Wartość ta może zostać nadana tylko przez Hero

HeroTest:

-HeroContainsHeroControllerScript - sprawdza czy Hero posiada właściwy skrypt

-HerosFieldsHaveCorrectModifiers - sprawdza czy do Hero zostały nadane wszystkie wymagane referencje do innych ważnych składników gry

-AnimatorFieldIsnotEmpty - sprawdza czy Hero ma Animator

-MovementKeysDownTest - sprawdza czy działa poruszanie się postaci

-HeroHasCorrectBackgroundSizeForBiggerScreens - sprawdza czy Hero ma poprawną skalę grafiki tła kamery, żeby nie pojawiło się niebieskie tło

LevelsManagementTest:

-AllLevelsExists - sprawdza czy są wszystkie pliki poziomów

-SerializeTestLevel - sprawdza czy z jednego pliku poziomu da się poprawnie zserializować poziom

MainGameMechnicsTest:

-GlobalManagerContainsGameMangerScript - sprawdza czy do GlobalManagers dołączony jest skrypt GameManager

-GlobalManagerContainsDataMangerScript - sprawdza czy do GlobalManagers dołączony jest skrypt DataManager

-GlobalManagerContainsCreativeLevelsManagerScript - sprawdza czy do GlobalManagers dołączony jest skrypt CreativeLevelsManager

-GlobalManagerContainsRankingManagerScript - sprawdza czy do GlobalManagers dołączony jest skrypt RankingManager

-GlobalManagerContainsAudioManagerScript - sprawdza czy do GlobalManagers dołączony jest skrypt AudioManager

-GlobalManagerContainsAudioSourceComponent - sprawdza czy do GlobalManagers dołączony jest komponent AudioSource

-AudioSourceContainsMainAudioClip - sprawdza czy do komponentu AudioSource zawartego w GlobalManagers podpięta jest właściwa muzyka

-CheckAvailabilityMainObjects - sprawdza czy główne obiekty gry występują

-CanvasHasEnabledOnlyMainContainer - sprawdza czy Canvas ma aktywowany tylko główny kontener

-EventSystemIsReadyToCheck - sprawdza czy system zdarzeń został poprawnie zainicjalizowany

Struktura folderu Scripts ze skryptami:

1. Folder Box
   1. Plik BoxScript.cs
      1. Klasa BoxScript
         1. Dziedziczenie:
            1. Klasa MonoBehaviour – podstawowa klasa z której pochodzi każdy skrypt edytora Unity
            2. Interfejs IContent
         2. Pola:
            1. Transform\_TargetTile – Komponent Transform należący do platformy na której znajduje się aktualnie skrzynka
            2. TileScript\_Current – Komponent TileScript należący do platformy na której znajduje się aktualnie skrzynka
         3. Metody:
            1. InitializeContent – po stworzeniu skrzynki należy ją zainicjalizować i połączyć z platformą
            2. Update – wywołuje się dokładnie co każdą klatkę – metoda wywoływana co klatkę, ale tylko gdy Transform\_TargetTile nie jest null’em
            3. ChangeTile – jedna z najważniejszych metod używana do zmiany platformy dla skrzynki
            4. Delay\_MoveChest – metoda która wykona się chwilę po fizycznej zmianie skrzynki. Używana do poprawnej animacji skrzynki
      2. Interfejs IContent

Interfejs został przygotowany z myślą o obiektach, które będą mogły zostać umieszczone na platformie.

1. Folder Creative
   1. Folder Canvas
      1. Pliki:
         1. Canvas\_Creative.cs
            1. Klasa Canvas\_Creative

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Panel\_InitializeGrid – referencja do obiektu InitializeGrid

Panel\_Tools – referencja do obiektu InitializeGrid

Panel\_Save – referencja do panelu Save

Metody:

Awake - wstępne zainicjalizowanie obiektu – wstępna inicjalizacja obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości – łączenie się z innymi obiektami i nadanie im właściwych wartości

* + - 1. IsMouseOverCanvas.cs
         1. Klasa IsMouseOverCanvas

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

MouseIsOverUI – referencja do skryptu wykrywającego czy myszka jest na elementem UI

Metody:

Awake - wstępne zainicjalizowanie obiektu

Update – wywołuje się dokładnie co każdą klatkę

* + - 1. Panel\_InitializeGrid.cs
         1. Przestrzeń nazw Creative

Klasa Panel\_InitializeGrid

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Slider\_Grid\_X – referencja do Slider’a dla osi X

Slider\_Grid\_Y– referencja do Slider’a dla osi Y

Text\_X – referencja do tekstu do osi X

Text\_Y– referencja do tekstu do osi Y

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

Submit – przesyła dane slider’a

OnSliderXChange – wywoła się kiedy Slider o nazwie SliderX zmieni swoją wartość

OnSliderYChange – wywoła się kiedy Slider o nazwie SliderX zmieni swoją wartość

GetStart - łączenie się z innymi obiektami i nadanie im właściwych wartości String

OnClick\_Back – zdarzenie przy naciśnięciu przycisku wstecz

* + - 1. Panel\_SaveScript.cs
         1. Klasa Panel\_SaveScript

Dziedziczy z klasy MonoBehaviour

Pole InputField\_LevelName – referencja do InputField o nazwie LevelName

Metoda OnClick\_SaveToFile – wywoła się po naciśnięciu przycisku zapisującego poziom

* + - 1. Panel\_ToolsScript.cs
         1. Przestrzeń nazw Creative

Klasa Panel\_ToolsScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Toggle\_RemoveTile - – referencja do Toggle o nazwie RemoveTile

Toggle\_TileType– referencja do Toggle o nazwie TileType

Toggle\_Box– referencja do Toggle o nazwie Box

Toggle\_HeroSet– referencja do Toggle o nazwie HeroSet

Metody:

Awake - wstępne zainicjalizowanie obiektu

OnClick\_Back – zdarzenie przy naciśnięciu przycisku wstecz

* 1. Pliki:
     1. CameraController.cs
        1. Przestrzeń nazw Creative
           1. Klasa CameraController

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Transform\_Hero – referencja do komponentu Transform należącego do Hero

mouseClickPos – Vector3 pozycja kliknięcia myszki

mouseCurrentPos – Vector3 aktualna pozycja myszki

padding - odstęp

\_lastDistance – ostatnia odległość od kliknięcia a pozycji myszki

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości ClickPosition

wasMoved – wartość true gdy po przytrzymaniu myszki nastąpi przesunięcie (dragMouse)

Color\_DisabledPlatform - kolor platformy gdy zostanie wyłączona (usunięta z poziomu)

lastTileScriptWithHero – skrypt platformy z hero

Metody:

Awake - wstępne zainicjalizowanie obiektu

Update – wywołuje się dokładnie co każdą klatkę

* + 1. CreativeDataManager.cs
       1. Przestrzeń nazw Creative
          1. Klasa CreativeDataManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

LevelData\_Serializable – aktualny obiekt poziomu

Metody:

Awake - wstępne zainicjalizowanie obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

* + 1. CreativeGridManager.cs
       1. Przestrzeń nazw Creative
          1. Klasa CreativeGridManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Prefab\_TileCreative – prefab z platformą dla sceny “Creative”

Transform\_TileContainer – kontener dla platform (TRANSFORM)

PaddingTile – odstęp pomiędzy platformami

Metody:

Awake - wstępne zainicjalizowanie obiektu

GenerateGrid – pozwala wygenerować siatkę platformą (X i Y)

ChangeContentOnTile – pozwala ustanowić zawartość dla platformy

* + 1. CreativeManager.cs
       1. Przestrzeń nazw Creative
          1. Klasa CreativeManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pole Prefab\_GlobalManagers – prefab globalnych menadżerów

Metoda Awake - wstępne zainicjalizowanie obiektu

* + 1. HeroCreativeScript.cs
       1. Przestrzeń nazw Creative
          1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
          2. Metoda ChangeContainer – pozwala na zmianę pozycji hero

1. Folder Environment
   1. Folder TileContainer
      1. Plik TileBoxContainerCounter.cs
         1. Klasa TileBoxContainerCounter
            1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
            2. Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

On\_BoxPlace\_Change – wywoływane gdy jakaś skrzynka zmieni platformę

BoxOnPlace – liczba skrzynek na właściwych platformach

MaxBoxPlaces – liczba wymaganych do ułożenia skrzynek

* + - * 1. Metody:

Awake - wstępne zainicjalizowanie obiektu

Refresh – pozwala sprawdzić czy w danym momencie zaszły jakieś zmiany w pozycji skrzynek

1. Folder Generic
   1. Pliki:
      1. AudioManager.cs
         1. Klasa AudioManager
            1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
            2. Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

AudioSource – reprezentuje źródło dźwięku

AudioSourceEffects – reprezentuje źródło efektów dźwiękowych

AudioClip\_Click – zawiera dane audio wywoływane przy kliknięciu

AudioClip\_ClickSecond - zawiera dane audio wywoływane przy ponownym kliknięciu

AudioClip\_BgMusic1 – zawiera dane audio użyte jako tło muzyczne

AudioClip\_BgMusic2 - zawiera dane audio użyte jako tło muzyczne

* + - * 1. Metody:

Awake - wstępne zainicjalizowanie obiektu

SceneManager\_sceneLoaded – steruje odtwarzaniem tła muzycznego

ChangeVolume – zmienia poziom głośności źródła dźwięku

PlayOnce – uruchamia efekt dźwiękowy odtwarzany raz

PlayClickSoundNormal – uruchamia efekt dźwiękowy pierwszego kliknięcia

PlayClickSoundSecond – uruchamia efekt dźwiękowy ponownego kliknięcia

* + 1. ClickSoundScript.cs
       1. Klasa ClickSoundScript
          1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
          2. Pole i – służy do sterowania metodą Awake
          3. Metoda Awake - wstępne zainicjalizowanie obiektu
    2. CreativeLevelsManager.cs
       1. Klasy:
          1. CreativeLevelsManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

customLevels\_Serializable – zawiera dane kreatora poziomów

Metody:

Awake - wstępne zainicjalizowanie obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

SaveState – zapisuje dane stworzonego poziomu do pliku Json

* + - * 1. CustomLevels\_Serializable

Pole levelData\_Serializables – zawiera listę danych kreatora poziomów

* + 1. DataManager.cs
       1. Klasy:
          1. DataManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu – zapewnia globalny dostęp do klasy DataManager

Prefab\_Tile – pole przypisane za pomocą Unity.Inspector. Jest to prefab kafelka

TilesData – Lista wszystkich rodzajów kafelków

ContentTileData – Lista wszystkich rodzajów zawartości dla platformy. Np. Skrzynka. Bohater…

MovingDelta – Szybkość z jaką porusza się skrzynka

AllLevelsData\_Serializable – lista wszystkich poziomów dostępnych w etapie 2 i 3

Stage2\_AllSavedLevelsData\_Serializable – wszystkie zapisane poziomy z etapu 2

Stage3\_AllSavedLevelsData\_Serializable - wszystkie zapisane poziomy z etapu 3

Metoda Awake - wstępne zainicjalizowanie obiektu – wywołana przed wygenerowaniem pierwszej klatki. Służy do zainicjalizowania wszystkich list

* + - * 1. Stage2\_SavedLevels

Pole savedLevels\_Serializables – lista zapisanych poziomów dla etapu 2

* + - * 1. Stage3\_SavedLevels

Pole savedLevels\_Serializables – lista zapisanych poziomów dla etapu 3

* + - * 1. TileData

Pola:

TileName – nazwa typu platformy

TileTypes – typ platformy (droga, ściana..)

Tile\_Sprite\_Bg – Grafika platformy

* + - 1. Typy wyliczeniowe:
         1. DifficultyLevel – trudność poziomu
         2. TileTypes – Typ kafelka
         3. ContentTile – zawartość kafelka
         4. Scenes – wszystkie dostępne sceny w grze
    1. GameManager.cs
       1. Klasa GameManager
          1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
          2. Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

CurrentStage – zawiera aktualnie wybrany etap

* + - * 1. Metoda Awake - wstępne zainicjalizowanie obiektu
      1. Typ wyliczeniowy Stages – zawiera etapy
    1. PlayerManager.cs
       1. Klasa PlayerManager
          1. Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

PlayerNick – zawiera nazwę gracza

* + - * 1. Metoda Awake - wstępne zainicjalizowanie obiektu
    1. RankingManager.cs
       1. Klasy:
          1. RankingManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

ranking\_Serializable – dane rankingu

Metody:

Awake - wstępne zainicjalizowanie obiektu

LoadState – wczytuje dane do rankingu

SaveState – zapisuje dane z rankingu do pliku Json

* + - * 1. Ranking\_Serializable

Pole rankingElement\_Serializables – zawiera listę rekordów z rankingu

* + - * 1. RankingElement\_Serializable

Pola:

Id – identyfikator gracza

playerName – nazwa gracza

rankingValue – punkty gracza

1. Folder Hero
   1. Plik HeroController.cs
      1. Klasa HeroController
         1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
         2. Pola:
            1. Animator\_Hero – interfejs Animator służący do kontroli animacji postaci
            2. SpriteRenderer\_Hero – render sprite’a postaci
            3. AudioClip\_Breathing – zawiera dane dźwięku oddychania postaci
            4. AudioClip\_Moving – zawiera dane dźwięku przesuwania skrzynki przez postać
            5. AudioClip\_Walking – zawiera dane dźwięku chodzenia postaci
            6. paddingTile – margines wewnętrzny dla tile’sa
            7. Transform\_TargetTile – dane o położeniu docelowego tile’sa
            8. TileScript\_Current - skrypt bieżącego tile’sa
            9. HeroStateAnimation\_Current – bieżący stan animacji postaci
            10. Hero\_Direction\_Current – bieżący kierunek w którym ustawiona jest postać
            11. \_heroIdleAnimation – animacja postaci gdy jest w spoczynku
            12. \_heroStateAnimation\_Current – bieżący stan animacji postaci
         3. Metody:
            1. Start - łączenie się z innymi obiektami i nadanie im właściwych wartości
            2. Update – wywołuje się dokładnie co każdą klatkę
            3. IsSomeoneKeyDown – sprawdza czy przycisk odpowiedzialny za poruszanie postacią został wciśnięty
            4. Instance – pozwala uzyskać globalny dostęp do obiektu\_OnEndLoadingLevel
            5. MoveTo – przesuwa postać na wybrany przez gracza tile’s
            6. GetNextTile – pobiera dane następnego tile’sa
         4. Typ wyliczeniowy HeroStateAnimation – stany animacji postaci
      2. Typ wyliczeniowy Direction – kierunki poruszania się postaci
2. Folder LevelLoader
   1. Folder Canvas
      1. Folder Panel\_Menu
         1. Pliki:
            1. Panel\_Stage2Script.cs

Klasa Panel\_Stage2Script

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Metoda OnClick\_SaveProgress – zapisuje stan rozgrywki po kliknięciu przycisku w etapie 2

* + - * 1. Panel\_Stage3Script.cs

Klasa Panel\_Stage3Script

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Metoda OnClick\_SaveProgress - zapisuje stan rozgrywki po kliknięciu przycisku w etapie 3

* + 1. Folder Panel\_Summary
       1. Plik Panel\_SummaryScript.cs
          1. Klasa Panel\_SummaryScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Button\_Continue – obiekt przycisku kontynuuj

Text\_Title – obiekt tekstu z tytułem

Text\_Score – obiekt tekstu z punktacją

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

OnClick\_ReturnToMainMenu – po kliknięciu przycisku powrotu wraca do menu głównego

OnClick\_NextLevel – po kliknięciu przycisku następny poziom uruchamia następny poziom

* + 1. Pliki:
       1. LevelLoaderCanvasManager.cs
          1. Klasa LevelLoaderCanvasManager

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Panel\_MainContainer – obiekt panelu głównego pojemnika

Panel\_Menu – obiekt panelu menu

Panel\_Summary – obiekt panelu podsumowania

Text\_BoxInRightPlace – obiekt tekstu z ilością dobrze ustawionych skrzyń

Text\_MovesCount – obiekt tekstu z ilością wykonanych ruchów

AudioClip\_ChestEvent – zawiera dane dźwiękowe przesunięcia skrzyni

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

OnBoxPlaceChange – zdarzenia dotyczące zmiany pozycji skrzyni

* + - 1. Panel\_MenuScript.cs
         1. Klasa Panel\_MenuScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Panel\_Stage2 – obiekt panelu etapu 2

Panel\_Stage3 – obiekt panelu etapu 3

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

OnClick\_Reload – po kliknięciu przycisku wczytuje scenę jeszcze raz

OnClick\_ReturnToMainMenu – po kliknięciu przycisku wraca do menu głównego

* + - 1. ParticlesUI2.cs
         1. Klasa ParticlesUI2

Dziedziczy z klasy MaskableGraphic – zawiera interfejsy odpowiedzialne za maskowanie grafiki

Pola:

particleTexture – textura użyta dla cząsteczki (ostatecznie nieużywana)

particleSprite – sprite (grafika dla cząsteczki)

\_transform – aktualny komponent Transform

\_particleSystem – aktualna referencja do komponentu PartcleSystem

\_particles – wszystkie wygenerowane cząsteczki

\_uv – mapa uv dla textury (nieużywana)

\_textureSheetAnimation – animacja dla textury (nieużywana)

\_textureSheetAnimationFrames – animacja ramek dla textury (nieużywana)

\_textureSheedAnimationFrameSize – animacja ramki dla textury (nieużywana)

mainTexture – główna textura (nieużywana)

Metody:

Initialize – zainicjalizowanie cząsteczek

Awake - wstępne zainicjalizowanie obiektu

OnPopulateMesh – w przypadku łączenia obiektu 3d (nieużywane)

Update – wywołuje się dokładnie co każdą klatkę

LateUpdate – wywołuje się dokładnie co każdą klatkę

* 1. Folder Loaders
     1. Plik LevelLoader.cs
        1. Klasy:
           1. LevelLoader

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Transform\_TileContainer – informacje o położeniu pojemnika tile’sów

PaddingTile – margines wewnętrzny tile’sa

Metody:

Awake - wstępne zainicjalizowanie obiektu

LoadLevel – wczytanie poziomu

GenerateGrid – generowanie siatki tile’sów

* + - * 1. LevelData\_Serializable

Pola:

Id – identyfikator poziomu

difficultyLevel – poziom trudności

gridData – dane siatki

grid\_X – pozycja x na osi siatki

grid\_Y – pozycja y na osi siatki

maxTime – maksymalny czas

minMovesToComplete – minimalna ilość ruchów

IsCustomLevel – czy poziom został stworzony przez gracza

CustomLevel\_Name – nazwa poziomu stworzonego przez gracza

IsCompleted – czy poziom został ukończony

Score – zdobyte punkty na danym poziomie

Moves – wykonana ilość ruchów

Time – czas

Metoda DeepCopy – zwraca kopię danych poziomu

* + - * 1. Tile\_Serializable

Pola:

Id – identyfikator

tileID – identyfikator tile’sa

contentID – zawartość tile’sa

positionInWorld – pozycja tile’sa

TileScript – skrypt tile’sa

Metody:

Tile\_Serializable

Tile\_Serializable – przypisanie identyfikatora

* 1. Pliki:
     1. LevelDataManager.cs
        1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
        2. Pola:
           1. Instance – pozwala uzyskać globalny dostęp do obiektu
           2. levelData\_Serializable – dane poziomu
           3. GameIsEnd – czy gra została ukończona
           4. OnMovesCountChange – zdarzenie wywoływane przy zmianie ilości ruchów
           5. MovesCount – aktualna ilość ruchów
           6. \_movesCount – początkowa ilość ruchów
        3. Metody:
           1. Awake - wstępne zainicjalizowanie obiektu
           2. Start - łączenie się z innymi obiektami i nadanie im właściwych wartości
           3. Instance – pozwala uzyskać globalny dostęp do obiektu\_OnEndLoadingLevel
     2. LevelLoaderManager.cs
        1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
        2. Pola:
           1. Instance – pozwala uzyskać globalny dostęp do obiektu
           2. Prefab\_GlobalManagers – obiekt prefab’a globalnego menadżera
           3. OnEndLoadingLevel – zdarzenie wywoływane po zakończeniu wczytywania poziomu
        3. Metody:
           1. Awake - wstępne zainicjalizowanie obiektu
           2. Start - łączenie się z innymi obiektami i nadanie im właściwych wartości
        4. Interfejs Delay\_Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

1. Folder MainMenu
   1. Folder CustomLevels
      1. Plik CustomLevelScript.cs
         1. Klasa CustomLevelScript
            1. Dziedziczenie:

Klasa MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Interfejs IPointerClickHandler – odpowiedzialny za wychwycenie kliknięcia

* + - * 1. Pola:

levelData\_Serializable – dane poziomu

savedLevelData\_Serializable – zapisane dane poziomu

* + - * 1. Metody:

OnPointerClick – ładuje wybrany poziom stworzony przez gracza po kliknięciu

ChangeSelectionState – zmienia wygląd stanu kliknięcia

* 1. Folder Panels\_Stages
     1. Folder FirstStage
        1. Plik FirstStageScript.cs
           1. Klasa FirstStageScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

MainContainer – obiekt głównego pojemnika

Panel\_DifficultyLevel – obiekt panelu wyboru trudności poziomu

Toggle\_Amateur – obiekt przełącznika na poziom trudności łatwy

Toggle\_Intermediate – obiekt przełącznika na poziom trudności średniozaawansowany

Toggle\_Expert – obiekt przełącznika na poziom eksperta

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

OnClick\_Play – rozpoczęcie gry po kliknięciu i wybraniu poziomu trudności

OnClick\_Exit – wyjście z aplikacji po kliknięciu

OnClick\_CloseDifficulty – zamknięcie okna wyboru trudności po kliknięciu

OnClick\_PlayWithSelectedDifficultyLevel – rozpoczęcie gry z wybranym poziomem trudności po kliknięciu

* + 1. Folder SecondStage
       1. Pliki:
          1. CampaignElementScript.cs

Klasa CampaignElementScript

Dziedziczenie:

Klasa MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Interfejs IPointerClickHandler – odpowiedzialny za wychwycenie kliknięcia

Pola:

Image\_SavedLevel – obiekt obrazu zapisanego poziomu

Text\_Points – obiekt tekstu zdobytych punktów

Image\_LevelThumbnail – obiekt obrazu miniaturki poziomu

Sprite\_LevelCompleted – obiekt sprite’a ukończonego poziomu

campaignElement\_Serializable – dane poziomu elementu kampanii

isAvailableContinue – czy możliwa jest kontynuacja poziomu

savedLevelData – zapisane dane poziomu

Metody:

Initialize – inicjalizuje element kampanii drugiego etapu

OnPointerClick – wybiera poziom do wczytania po kliknięciu

ChangeSelectionState – zmienia status wyglądu dla wybranego poziomu

* + - * 1. Panel\_CampaignScript.cs

Klasa Panel\_CampaignScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

Prefab\_CampaignElement – obiekt prefab’a elementu kampanii

RectTransform\_CampaignContent – informacje o pozycji zawartości kampanii

SelectedClearCampaignLevelToLoad – dane czystego poziomu kampanii do wczytania

Button\_Continue – obiekt przycisku kontynuuj

Button\_NewGame – obiekt przycisku nowa gra

Sprites\_Thumbnails – obiekt sprite’a miniaturek

campaignLevelScripts – lista skryptów poziomu kampanii

Metody:

Awake - wstępne zainicjalizowanie obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

SelectCustomLevelToLoad – wybiera poziom stworzony przez użytkownika do załadowania

OnClick\_Continue – kontynuuje rozgrywkę po kliknięciu

OnClick\_FromNew – zaczyna od czystego poziomu po kliknięciu

* + - * 1. SecondStageScript.cs

Klasa SecondStageScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

MainContainer – obiekt głównego pojemnika

Panel\_Campaign – obiekt panelu kampanii

Panel\_SetNick – obiekt panelu ustawiania nazwy gracza

Rank\_List – obiekt listy rankingowej

Button\_LoadSavedGames – obiekt przycisku wczytania zapisanych gier

Prefab\_RankingElement – obiekt prefab’a elementu rankingu

RectTransform\_RankingContent – informacje o pozycji zawartości rankingu

InputField\_Nickname – obiekt input’a przyjmującego nazwę gracza

Metody:

OnEnable – wykonuje się po aktywacji elementu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

OnClick\_Exit – wyjście z aplikacji po kliknięciu

OnClick\_Play – rozpoczęcie gry po kliknięciu i wybraniu poziomu

OnClick\_Back – zdarzenie przy naciśnięciu przycisku wstecz

OnClick\_Load – wczytanie poziomu po kliknięciu

OnClick\_Ranking – wczytanie rankingu po kliknięciu

OnClick\_SetNick – ustawienie nazwy gracze po kliknięciu

LoadRanking – wczytanie danych do rankingu

* + 1. Folder ThirdStage
       1. Plik ThirdStageScript.cs
          1. Klasa ThirdStageScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

MainContainer – obiekt głównego pojemnika

Panel\_CustomLevels – obiekt panelu poziomów stworzonych przez gracza

Prefab\_CustomLevel – obiekt prefab’a poziomu stworzonego przez gracza

RectTransform\_CustomLevelsContent – informacje o położeniu zawartości poziomów stworzonych przez gracza

Button\_Continue – obiekt przycisku kontynuuj

Button\_NewGame – obiekt przycisku nowa gra

Button\_Remove – obiekt przycisku usuń

customLevelScripts – lista skryptów poziomu stworzonego przez gracza

selectedGameObjectCustomLevel – wybrany obiekt poziomu stworzonego przez gracza

selectedLevelData\_SerializableToLoad – wybrane dane poziomu do załadowania

savedSelectedLevelData\_SerializableToLoad – zapisane wybrane dane poziomu do załadowania

Metody:

Awake - wstępne zainicjalizowanie obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

LoadLevels – wczytuje poziomy

SelectCustomLevelToLoad – wybiera poziom stworzony przez gracza do załadowania

OnClick\_Continue – kontynuuje rozgrywkę po kliknięciu

OnClick\_FromNew – zaczyna od nowego poziomu po kliknięciu

OnClick\_Remove – usuwa poziom po kliknięciu

OnClick\_Exit – zamyka aplikację po kliknięciu

OnClick\_Create – uruchamia kreator poziomów po kliknięciu

OnClick\_NewGame – rozpoczyna nową grę po kliknięciu

OnClick\_Back – zdarzenie przy naciśnięciu przycisku wstecz

* 1. Folder Toggles
     1. Pliki:
        1. PanelStagesScript.cs
           1. Przestrzeń nazw MainMenu

Klasa PanelStagesScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

TogglesScript\_Instance – pozwala uzyskać globalny dostęp do obiektu

Panels\_Stages – obiekt panelów etapów

Metody:

Awake - wstępne zainicjalizowanie obiektu

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

SwitchPanel – zmienia panel

* + - 1. TogglesScript.cs
         1. Przestrzeń nazw MainMenu

Klasa TogglesScript

Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity

Pola:

Toggle\_0 – obiekt przełącznika etapów

Toggle\_1 - obiekt przełącznika etapów

Toggle\_2 - obiekt przełącznika etapów

Image\_Icon – obiekt obrazu ikonki

Toggle\_Sound – obiekt przełącznika dźwięku

Sprite\_SoundOn – obiekt sprite’a włączonego dźwięku

Sprite\_SoundOff – obiekt sprite’a wyłączonego dźwięku

Metody:

Start - łączenie się z innymi obiektami i nadanie im właściwych wartości

SetToggleActive – aktywuje przełączniki etapów

OnToggleValueChange0 – aktywuje panel etapu pierwszego przy zmianie wartości przełącznika

OnToggleValueChange1 - aktywuje panel etapu drugiego przy zmianie wartości przełącznika

OnToggleValueChange2 - aktywuje panel etapu trzeciego przy zmianie wartości przełącznika

OnClick\_Sound – odtwarza dźwięk przy kliknięciu

* 1. Pliki:
     1. LevelDataHolder.cs
        1. Klasa LevelDataHolder
           1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
           2. Pole LevelData\_Serializable – dane poziomu
           3. Metody:

Awake - wstępne zainicjalizowanie obiektu

Clone – zwraca kopię poziomu

* + 1. MainMenuCanvas.cs
       1. Klasa MainMenuCanvas
          1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
          2. Pola:

Instance – pozwala uzyskać globalny dostęp do obiektu

GameObject\_Title – obiekt tytułu

GameObject\_TogglesStages – obiekt przełączników etapów

* + - * 1. Metoda Awake - wstępne zainicjalizowanie obiektu
    1. MainMenuManager.cs
       1. Klasa MainMenuManager
          1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
          2. Pole Prefab\_GlobalManagers – obiekt prefab’a globalnych menadżerów
          3. Metoda Awake - wstępne zainicjalizowanie obiektu

1. Folder Tile
   1. Plik TileScript.cs
      1. Klasa TileScript
         1. Dziedziczy z klasy MonoBehaviour - podstawowa klasa z której pochodzi każdy skrypt edytora Unity
         2. Pola:
            1. Tile\_Serializable – dane tile’sa
            2. TileData – dane tile’sa
            3. Transform\_ContentContainer – informacje o położeniu zawartości pojemnika
         3. Metody:
            1. Initialize – inicjalizuje tile’sa
            2. SetContent – ustawia zawartość tile’sa
            3. RemoveContent – usuwa zawartość tile’sa
            4. GetContent – pobiera zawartość tile’sa
            5. Reinitialize – ponownie inicjalizuje tile’sa
2. Plik GameAssembly.asmdef – zawiera dane dla edytora Unity wykorzystywane do TestRunner’a

---------------------------------------------------------------------------------------------------------------------------