Praca inżynierska – Adam Martenka 😊

Spis treści

[1. Wprowadzenie 3](#_Toc63774347)

[1.1. Zastosowanie 3](#_Toc63774348)

[1.2. Dlaczego Unity 4](#_Toc63774349)

[2. Przegląd istniejących rozwiązań 4](#_Toc63774350)

[3. Projekt 5](#_Toc63774351)

[3.1. Podstawowe elementy projektu w Unity 5](#_Toc63774352)

# Wprowadzenie

Tematem mojej pracy dyplomowej jest stworzenie generatora wirtualnego terenu na potrzeby wizualizacji trójwymiarowej. Program stworzony został w technologii Unity w języku C#. Aplikacja umożliwia wygenerowanie dowolnego terenu oraz dodanie na nim elementów wody. Poza samą możliwością stworzenia terenu istnieje również możliwość dodania elementów flory i fauny do stworzonego świata. Dodane zwierzęta potrafią chodzić, szukają wody i pożywienia oraz rozglądają się, czy w pobliżu nie ma jakiegoś innego zwierzęcia, które mogli by upolować, lub przed którym powinni uciekać. Niektóre osobniki odżywiają się dodanymi przez nas roślinami a inne poszukują zwierząt, które mogliby zjeść. Poza roślinami, którymi odżywiają się niektóre istoty, istnieje możliwość dodania różnego gatunku drzew. Pełnią one jednak jedynie funkcje ozdobne.

Postanowiłem stworzyć taki generator, ponieważ chce pogłębiać swoją wiedzę w pisaniu gier w technologii Unity. Ponadto zawsze fascynowało mnie zachowanie dzikich zwierząt, a możliwość obserwacji ich zachowań w stworzonym przez nas terenie wydała mi się bardzo interesująca.

## Zastosowanie

Podstawową funkcjonalnością projektu jest generowanie terenu na potrzeby wizualizacji trójwymiarowej oraz obserwacja dodanych na niego zwierząt i roślin. Wygenerowany teren może posłużyć nie tylko do obserwacji ale również do paru innych rzeczy. Między innymi, stworzony świat może zostać użyty w innym projekcie. Napisany przeze mnie program może posłużyć jako teren do innej gry napisanej w Unity. Stworzony świat może również posłużyć jako gotowa scena do programów, które zajmują się tworzeniem trójwymiarowych grafik lub animacji takich jak 3ds Max lub Blender.

Stworzony przeze mnie program może być również użyty jako narzędzie do symulacji i wizualizacji. Czasami zdarza się, że zachodzi potrzeba zwizualizowania jakiegoś terenu na potrzeby budownictwa lub podczas projektowania ogrodów. Wtedy program może zostać użyty do odtworzenia istniejącego już terenu i sprawdzenia czy istnieje możliwość stworzenia zaplanowanych wybudowania jakiegoś budynku, osiedla czy stworzenia ogrodu zoologicznego.

Należy pamiętać, że projekt można jeszcze rozwijać. Po zaimplementowaniu dodatkowych elementów, aplikacja może posłużyć do wielu więcej rzeczy. W przypadku dodania elementów pasu startowego czy lądowiska oraz odpowiedniego modelu kokpitu i wzbogaceniu o skrypty umożliwiające lot, lądowanie i startowanie, projekt może posłużyć jako symulator lądowania samolotów czy helikoptera.

Tak naprawdę projekt może zostać użyty do wszelakich gier czy też symulacji, gdzie potrzebny jest otwarty świat.

## Dlaczego Unity

Projekt stworzony został przeze mnie w technologii Unity, czyli bardzo popularnym i rozbudowanym silniku gier. Umożliwia on tworzenie gier na platformę PC ale i na urządzenia mobilne. Najbardziej popularnym językiem w którym są one pisane jest C#. Niegdyś pisano również językach takich jak Boo czy UnityScript(czyli, czegoś stworzonego na wzór JavaScript). Obecnie jednak gry pisze się wyłącznie w języku C#. Najpopularniejszymi grami stworzonymi w Unity są: Angry Birds 2, Among Us czy Hearthstone.

Technologia ta może być kojarzona z niszowymi produkcjami ponieważ panel przedstawiający fakt, że gra stworzona jest na silniku Unity, który pokazywany jest przy starcie gry, wymagany jest w produkcjach, które generują przychód poniżej 100.000$ rocznie. W przypadku, gdy gra zarabia więcej producent silnika nie wymaga aby jego logo prezentowane było na starcie gry.

Do wykonania mojego projektu postanowiłem wykorzystać technologie Unity z wielu różnych przyczyn. Przede wszystkim chciałem aby narzędzie, z którego będę korzystał w żadnym stopniu nie ograniczało moich pomysłów. Silnik ten ma ogromnie możliwości i w realizacji aplikacji ograniczała mnie jedynie wyobraźnia, moje umiejętności programowania oraz stopień znajomości samego Unity.

Innym powodem, dla którego zdecydowałem się na użycie tej technologii jest fakt, że bardzo chciałbym pogłębić swoje umiejętności tworzenia gier w Unity ale i w programowaniu przy pomocy języka C#.

# Przegląd istniejących rozwiązań

# Projekt

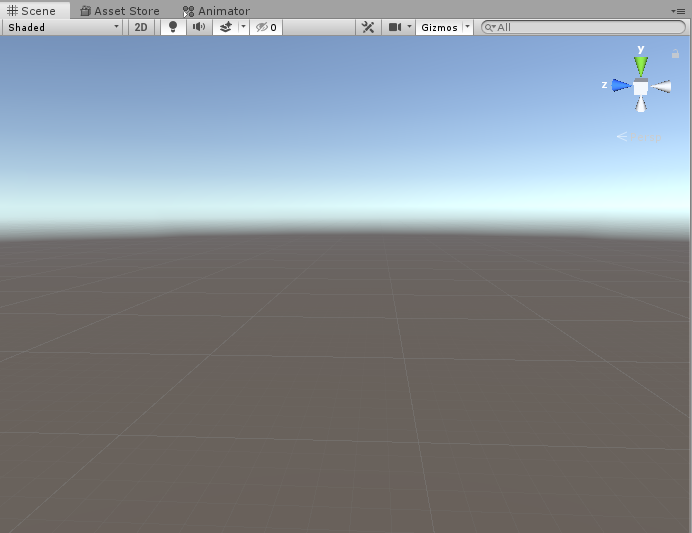
Każdy projekt stworzony w Unity ma kilka podstawowych elementów, bez których nie mógłby funkcjonować. Do stworzenia mojego projektu użyłem właśnie tych elementów ale również dodanego przez Projektanta silnika sklepu w którym można, również za darmo pobrać elementy świata przedstawionego, dzięki czemu nie musiałem wszystkich elementów tworzyć od zera. M.in. rzeczami, które pobrałem były: wygląd zwierząt, drzew i krzewów.

W tym rozdziale postaram się je omówić podstawowe elementy silnika a następnie, opiszę jak wyglądają one na przykładzie mojej aplikacji. Przedstawię powiązania między elementami w mojej aplikacji oraz w jaki sposób poszczególne skrypty działają na dane rzeczy.

## Podstawowe elementy projektu w Unity

Podczas tworzenia gry w Unity mamy wpływ na wiele rzeczy. Miedzy innymi możemy zdecydować, czy gra, którą chcemy tworzyć ma być w wymiarze 2D czy 3D, możemy wybrać edytor tekstu , w którym kod zamierzamy pisać czy … . Są jednak elementy, których użycie jest narzucone nam przed producenta silnika i aby móc stworzyć interesującą nas grę czy też symulacje jesteśmy zmuszeni ich użyć. Poniżej przedstawię te najważniejsze:

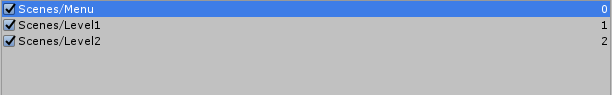
**Scena**



Rysunek 1 Podstawowy widok sceny w Unity

Scena jest obszarem gry, na który można dodawać elementy, które chcemy dodać do naszej rozgrywki. Może być również wykorzystana jako niezależny od innych scen poziom gry lub można na niej stworzyć Menu, które wyświetlane będzie na początku gry. W projekcie może istnieć więcej niż jedna scena, a przechodzenie pomiędzy nimi może być zaprogramowane w skrypcie. Widoczna na Rys.1. scena, to widok podstawowy, który widzimy po uwożeniu gry.

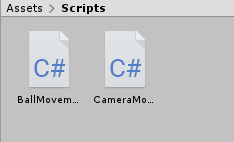
Stworzone sceny posiadają swój numer, co ułatwia zaprogramowanie płynnego przechodzenia między nimi.



Rysunek 2 Sceny przedstawiające menu oraz dwa poziomy, które są kolejno ponumerowane.

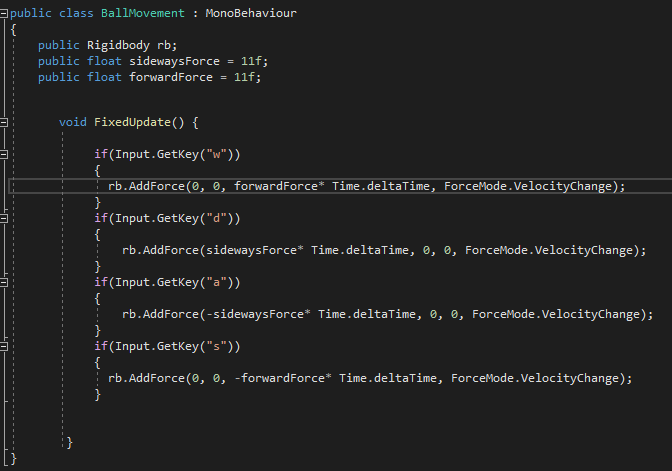
**Skrypty**

Skrypt, czyli kod napisany w języku C#, lub innym, który umożliwia nam pisanie w Unity. To dzięki nim mamy możliwość ingerowania w to, co dzieje się w naszej grze. To właśnie kod, który wpisujemy w tych plikach pozwala nam na zmienianie scen, ruch postaci lub ingerencje w dodanych przez nas elementów poziomu. Skrypty można tworzyć niezależnie od siebie ale najlepiej, aby każdy z nich odpowiadał za jedną rzecz, taką jak na przykład ruch, zmiana scen lub rozwój postaci.



Rysunek 3 Przykładowe skrypty

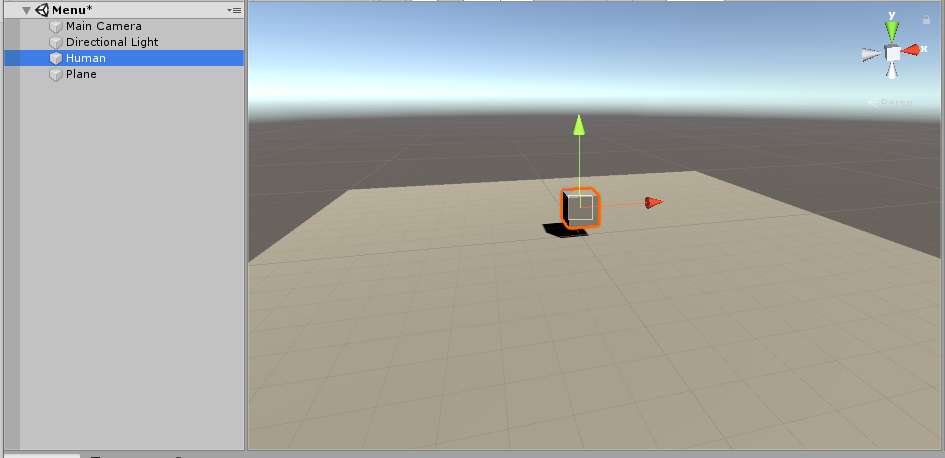
Na rysunku 4. widać zawartość zaprezentowanego powyżej skryptu BallMovement. Skrypt, napisany w języku C# odpowiedzialny jest za poruszanie się danego elementu za pomocą przycisków W, A, S i D. Skrypt ten został dodany do obiektu Ball.



Rysunek 4 Fragment skryptu BallMovement

**GameObjects**

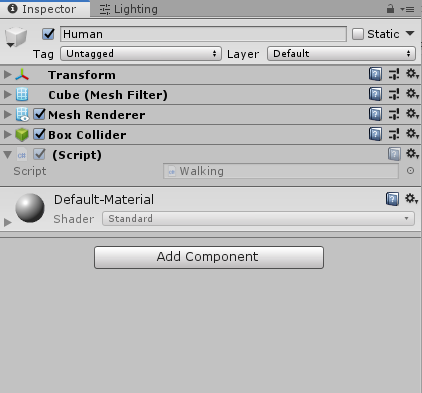
Są to elementy gry, które dodajemy do sceny. Funkcjonowanie ich zaprogramowane jest w skryptach. GameObject może mieć dowolną ilość skryptów, które mogą odpowiadać za każdą interakcje z światem stworzonym lub to co dany GameObject ma robić. Elementy te mogą być zwierzętami, drzewami, pojazdami, trawą czy skała a wszystko co mają robić można dzięki skryptom do nich przypisać.



Rysunek 6 Dodany do sceny GameObject o nazwie Human

Dodane przez nas obiekty mają swoje wbudowane przez producenta silnika cechy, które można zobaczyć w Inspectorze. Po naciśnięciu na konkretny element na scenie, wyświetla się on w panelu bocznym. Właśnie tam można dodać, stworzone przez nas skrypty.

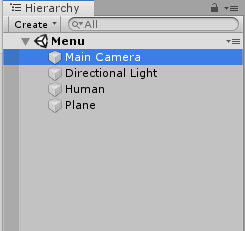
Do GameObjectu o nazwie Human został dodany skrypt o nazwie Walking, Rys.5.



Rysunek 7 Podgląd cech i funkcjonalności GameObjectu o nazwie Human

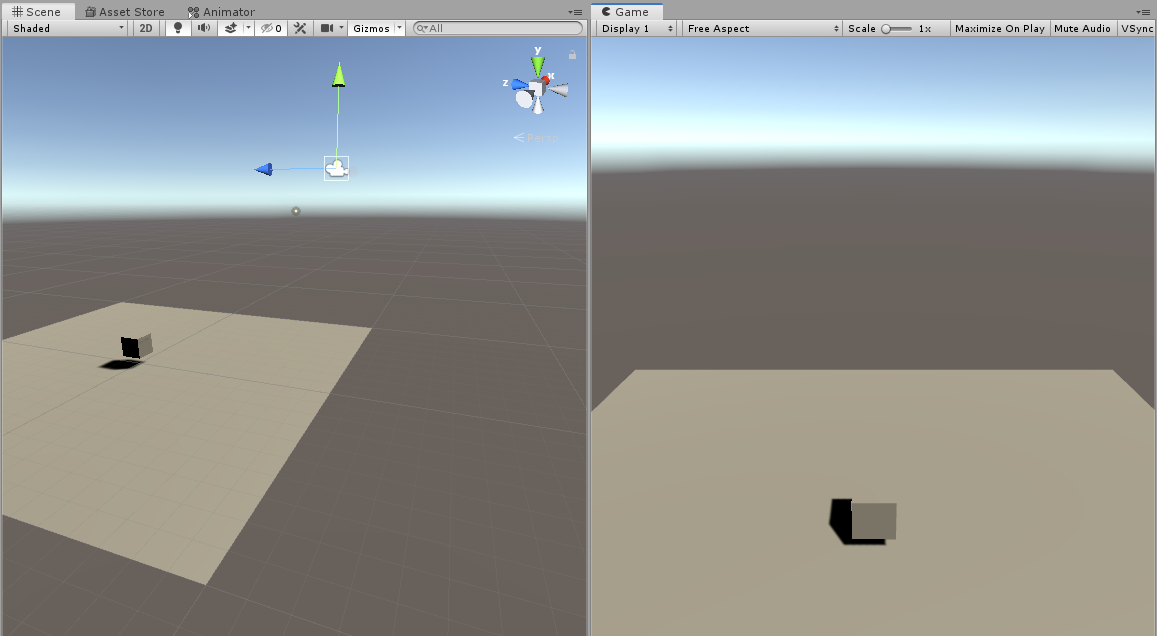
**Kamera**

W każdym projekcie Unity znajduje się kamera. Jest to jeden z elementów, jak scena, który jest dodany od momentu stworzenia projektu. Ten element, jest o tyle ważny, że gdyby nie on to nie widzielibyśmy w naszej grze niczego. W kadrze kamery znajdują się elementy, które widzi gracz po uruchomieniu gry.



Rysunek 8 Kamera opisana jest pod nazwą "Main Camera" w głównej hierarchii projektu

Ustawienie kamery gracz może zmieniać dowolnie. W oknie Scene widzimy wszystkie elementy w stworzonym świecie. Można również w nim ustawić kamerę oraz kadr, który widoczny będzie w grze. W oknie „Game”, widzimy to, co znajduje się w oku kamery, a zarazem to co widzi gracz po uruchomieniu gry.



Rysunek 9W oknie "Scene" zaznaczona jest kamera a w oknie "Game" widać co jej kadrze.

Ruch kamery można modyfikować dowolnie w skryptach, co zaprezentuje w późniejszym etapie pracy.