UNIWERSYTET WARMIŃSKO-MAZURSKI

W OLSZTYNIE

WYDZIAŁ MATEMATYKI I INFORMATYKI

**Kierunek: Informatyka**

**Daniel Jan Zamojski**

**Gra 2D typu Shooter z wykorzystaniem środowiska Unity**

Praca inżynierska wykonana

w ........................ (*nazwa katedry*)

pod kierunkiem

*dr Jacek Marchwicki*

Olsztyn 2015 UNIVERSITY OF WARMIA AND MAZURY

IN OLSZTYN

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

**Field of study: Computer Science**

**Daniel Jan Zamojski**

**2D shooter game made with Unity Engine**

Engineering Thesis written in

…………………….(nazwa katedry)

under supervision of

*dr Jacek Marchwicki*

Olsztyn 2015

Tematem pracy inżynierskiej jest stworzenie gry wideo należącej do gatunku Shooter za pomocą silnika Unity. Celem gracza przechodzenie etapów i pokonywanie przeciwników. Gra posiada kilka rodzajów broni oraz wrogów.

Opis pracy

Gry typu Shooter

Gatunek gier, gdzie głównym celem rozgrywki jest pokonywanie przeciwników za pomocą różnych broni, takich jak pistolety czy karabiny.

**Gry typu Shoot’em up**

Podgatunek gier typu Shooter. Zwykle przedstawiane za pomocą widoku z góry lub z boku. Postać gracza często ukazana jest jako pojazd lub statek kosmiczny. Grający powinien strzelać do wszystkiego co się rusza lub co stwarza zagrożenie. Należą do jednych z najszybszych rodzajów gatunków gier.

Gry typu Shoot’em Up testują refleks, spostrzegawczość i szybkość gracza oraz umiejętność zapamiętywania schematów i położenia przeciwników.

**Docelowe platformy**

Komputery osobiste

**Wykorzystane technologie**

Unity– silnik gry służący on do tworzenia gier dwuwymiarowych i trójwymiarowych dla wielu platform.

Język programowania C#

GIMP- (GNU Image Manipulation Program)- darmowy program open source służący do obróbki grafiki rastrowej.

Visual Studio – środowisko programistyczne stworzone przez Microsoft, służące do tworzenia programów komputerowych, stron i aplikacji internetowych oraz aplikacji mobilnych.

**Silnik gry**

Unity jest to wieloplatformowy silnik gry stworzony przez Unity Technologies. Służy on to tworzenia gier 3D i 2D, symulacji komputerowych oraz animacji. Pierwszy raz zapowiedziany i wydany w czerwcu 2005 roku podczas Apple Inc.'s Worldwide Developers Conference.

Silnik Unity umożliwia tworzenie w 2D oraz 3D oraz oferuje podstawowe skrypty API w języku C# .

Podczas projektowania gier 2D, Unity umożliwia importowanie sprite’ów oraz zaawansowane renderowanie świata 2D.

Z kolei dla gier 3D, zapewnia wsparcie dla

Jedną z wielu zalet silnika Unity, jest jego kompatybilność. Wiele platform, takich jak Windows, Android, konsole siódmej i ósmej generacji jest przez niego obsługiwana.

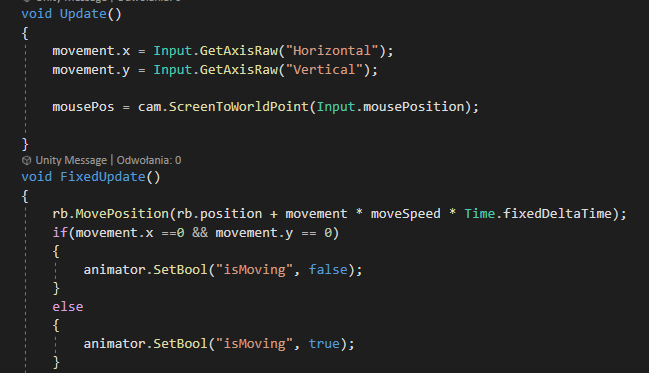
Według danych z 2018 roku, około połowa gier mobilnych na rynku oraz 60% gier AR i VR została stworzona za pomocą silnika Unity.

**Historia gier wideo**

**Rozgrywka**

**Poruszanie postacią**

Aby przemieścić się bohaterem, należy nacisnąć przycisk odpowiedzialny za ruch w danym kierunku(W – w górę, A- w lewo, D – w prawo, S – w dół). Następnie postać poruszy się z ustaloną wcześniej prędkością. Po napotkaniu przeszkody, gracz nie będzie mógł pójść dalej. Podczas ruchu, bohater jest odpowiednio animowany.

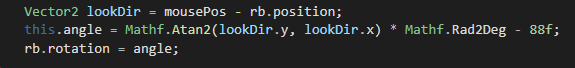


**Obracanie postacią**

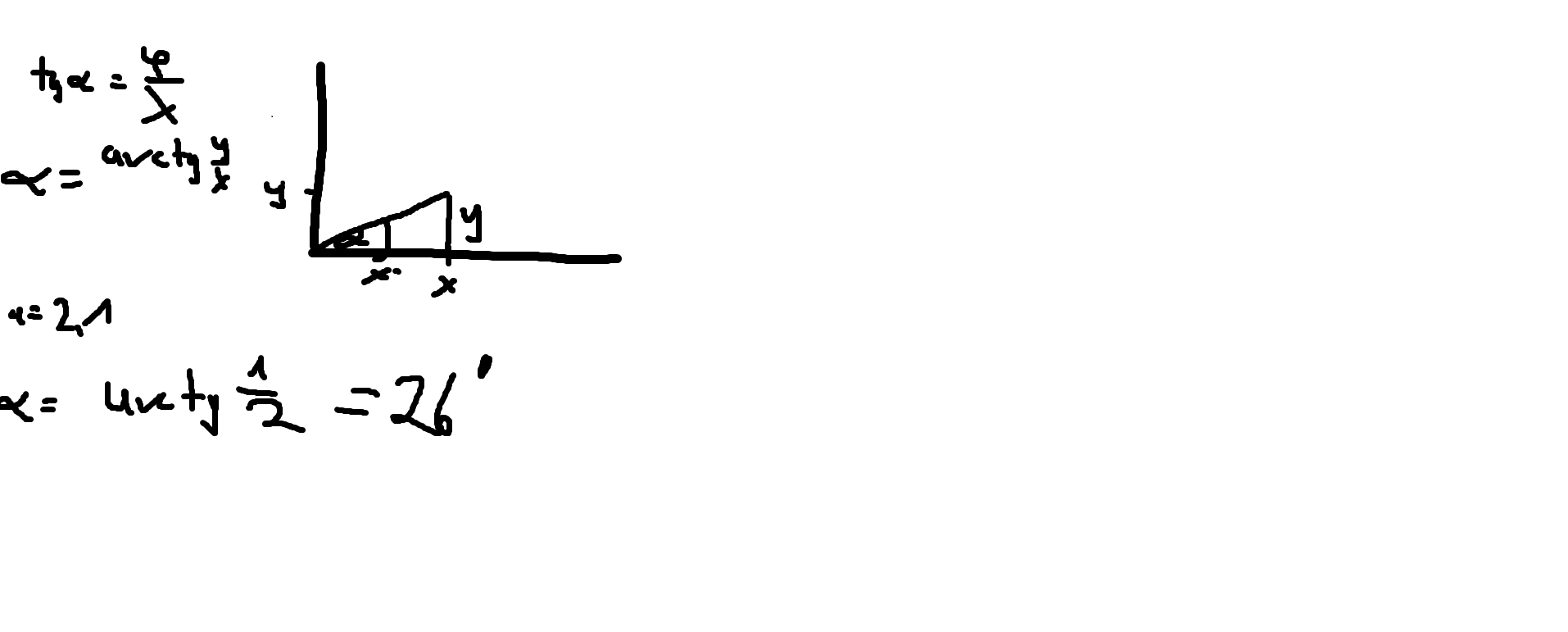
Postać obraca się zgodnie z ruchem myszy. Kąt obrotu jest wyznaczany za pomocą metody, która wykorzystuje funkcję cyklometryczną arcus tangens y/x, gdzie y i x to pozycje wektora lookDir. Następnie wynik tej operacji jest zmieniany z radiany na stopnie.

Otrzymany kąt przypisuje się potem do parametru rotacji RigidBody2D postaci.

lookDir to wektor różnicy pozycji myszki(mousePos) i pozycji gracza(rb.position).

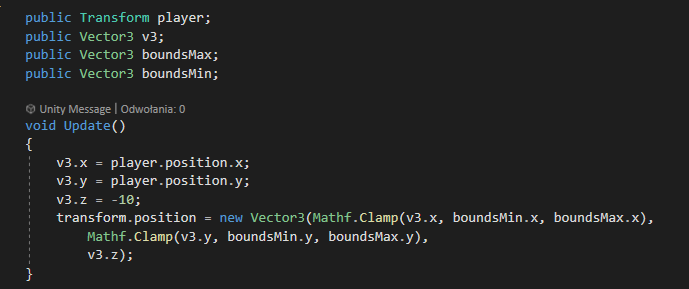


//przykład do poprawy



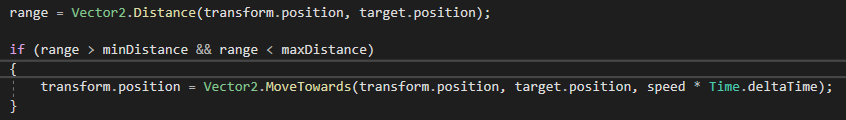
**Kamera**

Mapa oraz wszystkie jej elementy ukazywane są w formie dwuwymiarowej. Kamera przemieszcza się zgodnie z ruchem gracza. Jeśli bohater będzie w pobliżu krawędzi mapy, kamera zablokuje się do momentu odejścia od okolicy granicy.

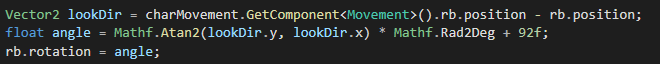


**Poruszanie i obrót przeciwnika**

Przeciwnicy mogą poruszać się prosto na gracza, wykorzystując przy tym funkcję silnika Unity MoveTowards, w której podaje się aktualną pozycję gracza oraz przeciwnika.



Obracanie się przeciwnika jest obsługiwane w podobny sposób co obrót gracza, z tą różnicą, że zamiast aktualnej pozycji myszy na ekranie podaje się aktualną pozycję gracza.



**Interfejs użytkownika**

Pasek zdrowia

Graficzna reprezentacja stanu zdrowia gracza



Broń i amunicja

Nazwa aktualnej broni oraz obecny stan amunicji i magazynka.



Punkty

Obecny stan zdobytych punktów.



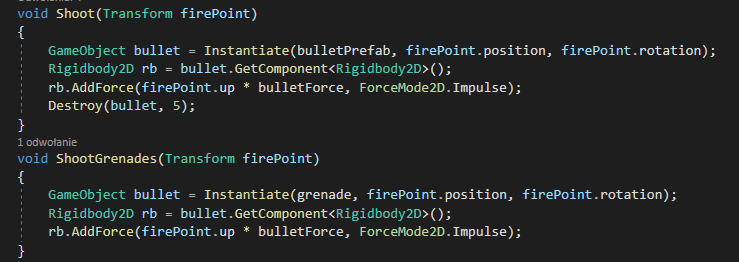
Ikony umiejętności

Ukazują czy możliwe jest użycie danej umiejętności oraz ile czasu pozostało do zakończenia oczekiwania.



**Strzelanie**

Postać ma przypisany do niej punkt z którego będą wydobywać się pociski. Punkt ten jest zawsze w tym samym miejscu w stosunku do bohatera. Po naciśnięciu lewego przycisku myszy, tworzony jest nowy obiekt (klon) pocisku, który porusza się z odpowiednią dla danej broni prędkością.



**Pociski**

W momencie wystrzału, tworzony jest klon prefabrykatu, w którym zawarte są informacje, takie jak:

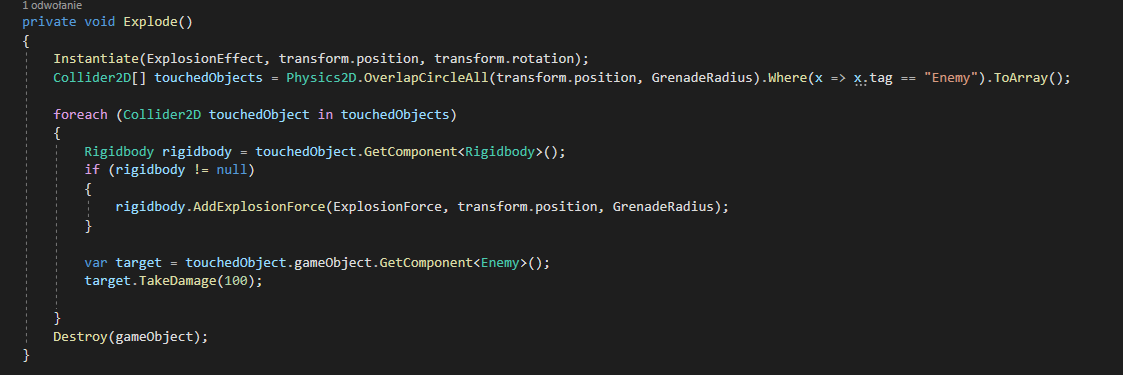
Wygląd pocisku,

Wielkość pocisku,

Obrażenia pocisku,

Krawędzie odpowiadające za kolizję

Wyjątkowym typem pocisku jest granat, który podczas wybuchu, zadaje obrażenia wszystkim przeciwnikom w ustalonym wcześniej promieniu.



Wrogów znajdujących się w promieniu wybuchu wyszukuje się za pomocą metody silnika Unity Physics2D.OverlapCircleAll, która wykrywa każdy obiekt znajdujący się w promieniu eksplozji(GrenadeRadius), a następnie filtruje po odpowiednim Tagu(w tym przypadku Enemy) oraz dodaje obiekty do tabeli. Następnie każdemu przeciwnikowi zadawane są ustalone wcześniej obrażenia.

Jeśli pocisk natrafi na przeciwnika lub gracza, wywoływana jest metoda odpowiedzialna za otrzymywanie obrażeń, po czym klon jest usuwany.

Jeśli wystrzelony został granat, wybucha on podczas kolizji z przeciwnikiem lub obiektem statycznym. (DOROBIĆ)

Jeśli pocisk nic nie napotka na swojej drodze przez określony czas, nabój zostanie usunięty lub wybuchnie w przypadku granatu.

**Punkty**

Gracz zdobywa punkty poprzez pokonywanie przeciwników. Każdy z wrogów posiada ilość punktów, jakie może zdobyć użytkownik.

Jeśli bohater otrzyma obrażenia, jego punktacja zmniejszy się.

**Kolizje**

Kolizje są obsługiwane przez silnik Unity. Obiekty posiadają BoxCollider2D lub TileMapCollider2D, które tworzą krawędzie odpowiadające za kolizje. W przypadku wystąpienia zderzenia może zostać wywołana metoda np. odpowiadająca za otrzymywanie obrażeń. Dodatkowo bohater jest blokowany przez obiekt posiadający Collider.



Przykładowy BoxCollider2D dla obiektu statycznego

**Umiejętności**

Bohater ma możliwość użycia dwóch umiejętności:

Sprint- na krótki czas prędkość gracza zwiększa się

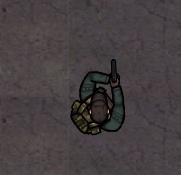
Niewrażliwość- na krótki czas gracz jest odporny na obrażenia

Każda z umiejętności ma swój własny czas trwania. Po jego upływie, możliwość użycia zostaje zablokowana na pewien okres.

**Bronie**

Gracz ma do dyspozycji cztery bronie: pistolet, karabin, strzelba oraz granatnik. Każda z nich ma swoje właściwości.

Pistolet – małe obrażenia, niewielka szybkość pocisku, długi czas pomiędzy kolejnymi wystrzałami.



Karabin – duże obrażenia, wysoka szybkość pocisku, krótki czas pomiędzy kolejnymi wystrzałami



Strzelba – obrażenia wahają się od małych, po duże w zależności od ilości trafionych pocisków, średnia szybkość pocisku, długi czas pomiędzy kolejnymi wystrzałami



Granatnik – duże obrażenia, mała szybkość pocisku, długi czas pomiędzy kolejnymi wystrzałami.



Każda z broni ma swoją własną amunicję, która jeśli się wyczerpie, uniemożliwi dalsze oddawanie strzałów.

**Przeciwnicy**

W grze można napotkać kilka rodzajów przeciwników.

Żołnierz z karabinem - oddaje on strzał w stronę gracza co określony czas. Może także poruszać się w stronę bohatera, jeśli znajdzie się on w promieniu wykrywania.



Nożownik - Przemieszcza się on do gracza i jeśli jest w określonym promieniu, zadaje mu obrażenia.



Wieżyczka – wytrzymały oraz zadający duże obrażenia wróg. Nie może się przemieszczać. Strzela w kierunku gracza, jeśli będzie odpowiednio blisko.



Boss – ostatni typ przeciwnika. Posiada on kilkukrotnie więcej zdrowia od zwykłego wroga oraz zadaje więcej obrażeń. Każdy z nich ma unikalny sposób poruszania się.



**Pułapka ogniowa//dodać więcej w grze**

Jedynym rodzajem pułapek jest pułapka ogniowa. Zadaje ona obrażenia graczowi, jeśli znajduje się on w jej BoxCollider2D. Może występować nieprzerwanie lub pojawiać się i znikać, według ustalonego wcześniej czasu

**Przedmioty do podniesienia**

Podczas rozgrywki gracz może natknąć się na kilka przedmiotów, takich jak:

Amunicja –po podniesieniu zwiększy obecny stan amunicji w broni, której odpowiada



Apteczka – po podniesieniu zwiększy obecne zdrowie.

****

**Obiekty statyczne**

Obiekty statyczne można odnaleźć na każdym poziomie. Stanowią one przeszkodę, uniemożliwiającą graczowi ruch. Są to:

- zniszczone pojazdy

- budynki

- mur

- kamienie

- łóżka, stoły, krzesła



Przykładowy obiekt - zniszczony samochód

**Krajobraz**

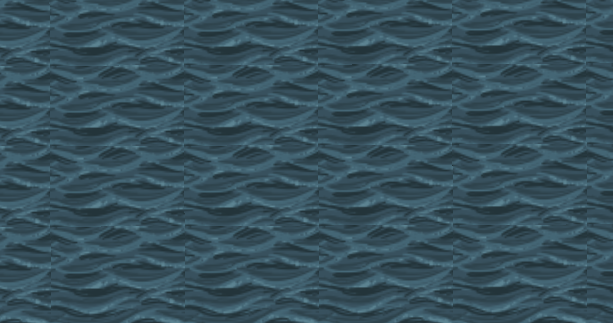
Podczas gry można napotkać:

- drogi

- mosty

- zbiorniki wodne

- tereny trawiaste i piaszczyste



Rysunek Przykładowy element krajobrazu – zbiornik wodny

**Poziomy**

Każdy poziom zawiera punkt startowy na którym pojawia się postać gracza oraz punkt końcowy, w którym po dotarciu bohatera przechodzi się do następnego etapu. Plansze posiadają elementy statyczne, przeciwników oraz elementy tła.



Przykładowy poziom.

**Menu**

Główny ekran pojawiający się po uruchomieniu gry. Umożliwia on rozpoczęcie rozgrywki, zmianę ustawień, wyświetlenie instrukcji oraz wyjście z gry.



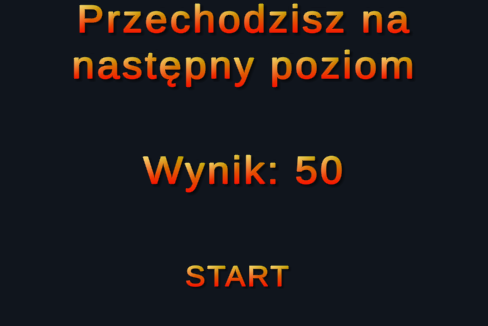
**Menu pauzy**

Po naciśnięciu przycisku odpowiadającego za pauzę, ekran staje się ciemniejszy oraz pojawiają się przyciski. Odpowiadają one za ponowne wznowienie gry, wyjście do menu głównego oraz wyjście z gry.



**Ekran zmiany planszy**

Wyświetla się, jeśli przechodzimy z jednego etapu na drugi. Można na nim ujrzeć obecny wynik. Po naciśnięciu przycisku START rozpocznie się rozgrywka na następnej planszy.



**Ekran zwycięstwa**

Wyświetla się on przy zakończeniu gry. Można na nim zobaczyć wynik końcowy. Za pomocą przycisków można przejść do menu głównego lub wyjść z gry.



**Ekran porażki**

Wyświetla się, jeśli zdrowie bohatera spadnie do zera. Za pomocą przycisków można przejść do menu głównego lub wyjść z gry.



**Widok gry**

**Warunki ukończenia gry**

**Zwycięstwo**

Aby ukończyć dany poziom, należy dotrzeć postacią do jego końca(np. wejście do budynku, samochód).

Do przejścia całej gry, należy przejść każdy etap bez utraty życia.

Porażka

Rozgrywka kończy się, jeśli zdrowie bohatera spadnie do zera.

**Dźwięk i muzyka**

//Do zrobienia

Na starcie każdego poziomu oraz w menu głównym odtwarzana będzie muzyka, która ucichnie w momencie zapauzowania gry.

//

Po oddaniu strzału lub przeładowaniu zostanie odtworzony odpowiedni dźwięk.