

UNIwersYTET WARMIŃSKO-MAZURSKI
W OLSZTYNIE

WYDZIAŁ MATEMATYKI I INFORMATYKI

Kierunek: Informatyka

Piotr Bońkowski

Tworzenie gry komputerowej pod tytułem "Plemię"

Praca magisterska inżynierska

wykonana

w Katedrze Fizyki Relatywistycznej

pod kierunkiem

dr Sławomira Kuleszy

Olsztyn 2015 rok

**University of Warmia and Mazury
In OLSZTYN**

FACULTY OF MATHEMATICS AND SCIENCE

Field: Informatics

Piotr Bońkowski

Creating a computer game called "Plemię"

Master Thesis Engineering
made
in the Department of relativistic physics
under the direction of
Dr. Sławomir Kulesza

Olsztyn 2015 years

Wstęp

Celem pracy było stworzenie gry komputerowej. Zakładałem, że będzie to gra strategiczna w widoku TPP ze sterowaniem armią, przez wydawanie prostych rozkazów, takich jak np. " idź do celu". Gra została stworzona na silniku Unity3D.

W rozdziale pierwszym przedstawiono historie gier elektronicznych , opis różnych gatunków gier z przykładami oraz opisane zostały silniki do tworzenia gier. W następnym rozdziale opisano stworzoną przeze mnie grę "Plemię" oraz narzędzia które użyłem do stworzenia jej. W ostatnim rozdziale przedstawiłem testy gry "Plemię" oraz możliwe zmiany w grze.

Rozdział 1

Tworzenie gier komputerowych

1.1. Historia tworzenia gier elektronicznych

Za pierwszą grę elektroniczną uważa się "Tenis for Two" (Tenis dla dwojga). Była to gra, w której dwóch graczy gra w tenisa stołowego na oscyloskopie. Następną ciekawą grą była gra "Spacewar!", gra również dla dwóch graczy, polegała na sterowaniu statkiem kosmicznym i strzelaniu do siebie nawzajem. To pierwsza gra, która zawierała tło oraz stosowała grafikę wektorową. Pierwsza konsola do gier nazywała się "MagnavoxOdyssey" i nie była zbyt popularna ze względu na marketing i dostępność. Najpopularniejszą grą w tej konsoli był "Pong". Inną ważną konsolą była "Fairchild Channel F", która jako pierwsza posiadała procesor i zapoczątkowała drugą generację konsol. W tej generacji kolejną ważną konsolą była "Video Computer System" (później zwana "Atari 2600"), która jako pierwsza posiadała wymianę kartridżów.

Przez następne lata powstawały coraz to mocniejsze konsole, ale nie wprowadzały szczególnie nic nowego, po czym nastąpiła zapaść na rynku gier aż do trzeciej generacji. Powstało wtedy wiele gier popularnych po dziś dzień, takie jak: "Mario", "The Legend of Zelda", "Final Fantasy" od Nintendo czy "Sonic the Hedgehog" od SEGA. W tym czasie pojawiły się, ciesząc się popularnością automaty, gdzie królowały takie gry jak: "Pac-Man", "Space Invaders", "Asteroids", "Night Driver", natomiast na komputery zaczęły powstawać takie gry jak: "Defender", "Battlezone", "3D Monster Maze", "Dungeons of Daggorath", "Pole Position" czy "Zork".

Niedługo potem czwarta generacja konsol wprowadziła grafikę 16-bitową. Powstawały takie gry jak: "Final Fantasy" czy "Street Fighter". Na komputery natomiast powstała gra "Dune II", którą uznaje się za prekursora gier RTS, "Alone in the dark", która zapoczątkowała survival horrory, "Doom" natomiast zapoczątkowała first person shootery. Powstała też pierwsza przenośna konsola do gier "GameBoy". Piąta generacja skupiła się na ulepszeniu grafiki, wydajniejszym sprzęcie oraz zaczęto używać płyt CD. Upowszechniono gry trójwymiarowe oraz rozgrywkę wieloosobową. Powstały też pierwsze gry MMO, takie jak: "Ultima Online" czy "EverQuest". Szusta generacja wprowadziła Internet w konsolach oraz dalej rozwijała założenia poprzednich generacji. Zapoczątkowała też masową produkcję gier. Obecnie

trwa siódma generacja. Rozwinęła ona znacznie konsole przenośne, smartfony i tablety. Powstały też różnego rodzaju kontrolery ruchowe.

1.2. Cele gier

Od czasu powstania pierwszych gier wiele się zmieniło, technologia poszła do przodu, a twórcy gier wpadają na coraz to nowe pomysły. Obecnie w tworzeniu gier ogranicza nas ludzka wyobraźnia oraz posiadane zasoby. Gry praktycznie z dnia na dzień się rozwijają. Powstają coraz to nowe ich gatunki. Oto kilka najpopularniejszych przykładów:

- Zręcznościowe - gry tego typu polegają na refleksie i odpowiednim reagowaniu na działanie gry, by osiągnąć jej cele. Gry te wymagają od gracza szybkiego reagowania na to, co się dzieje na ekranie monitora. Gry zręcznościowe były pierwszymi grami elektronicznymi, dlatego też inne gatunki zawierają ich pewne elementy. Przykładami gier zręcznościowych są takie gry jak: Pac-Man, Mario, RaymanOrigins, Call of duty.



Rys.1 - scena z gry RaymanOrigins

- RPG - gry tego typu polegają na sterowaniu postaci lub ich niewielkiej grupy. Postacie te zdobywają poziom w celu ich rozwoju. Gry tego typu zawierają rozbudowaną fabułę z wieloma wątkami pobocznymi. Gry te kończą się zazwyczaj, gdy wykonana zostanie zadania z głównej fabuły. Przykładami są gry takie jak: Wiedźmin, Baldur'sGate, ElderScrolls.



Rys.2-scena z gry Wiedźmin

- Strategie- polegają na osiągnięciu celów lub jak najlepszych wyników poprzez planowanie i strategiczne myślenie. Rozgrywka dzieje się w nich na dwa sposoby: w czasie rzeczywistym lub turowym. Gracze zazwyczaj kierują pewnymi zasobami oraz grupą postaci, których liczbę można zwiększać i nią dysponować. Od graczy oczekuje się przemyślanej taktyki oraz sprytu. Przykładami tego typu gier są: Cywilizacja , Starcraft , Heroes.



Rys.3-scena z gry Cywilizacja V

- Wyścigi - w tego typu grach gracz lub gracze ścigają się i/ lub starają się dotrzeć do celu w jak najkrótszym czasie na wyznaczonej trasie w pojazdach mechanicznych. Do tego gatunku należą: Needfor speed, The Crew, Grid.



Rys.4-scena z gry The Crew

- Sportowe - w grach tych kieruje się postacią lub drużyną postaci, które wykonują wyczyny sportowe, starając się osiągnąć jak najlepszy wynik. Często w grach tego typu wymagane jest odpowiednie wyczucie czasu lub taktyczne podejście. Przykładami tych gier są: Fifa, Tony Hawk's American Wasteland czy Deluxe SkiJump



Rys.5--scena z gry FIFA 2014

- MMO - jest mieszaniną różnych gatunków, przeznaczoną dla znacznie ilości graczy tylko przez Internet. Gracze walczą samotnie lub w grupach przeciwko sobie lub komputerowi. Przykładami są: World of Warcraft, Guild Wars, League of Legends.



Rys.6- scena z gry World of Warcraft

- Przygodowe - w grach tych opowiadane jest pewne wydarzenie, kryjące zazwyczaj jakąś tajemnicę z perspektywy głównego bohatera. Gracz musi w niej wykorzystać dostępne przedmioty do dotarcia w konkretne miejsce lub odkrycia tajemnicy. Nazywane też są interaktywnymi książkami. Do gier tego gatunku należą: Black Mirror, Syberia, Fahrenheit.



Rys.7-scena z gry Black Mirror

- Symulatory - jak sama nazwa wskazuje, gry te symulują określone w grze czynności, np. życia (Simsy) lub sterują czymś (symulator traktora) itp.



Rys.8-scena z gry The Sims 4

- Akcji- w tego typu grach gracz bierze udział w wydarzeniach, na które powinien reagować. Gry te skupiają się na fabule przeważnie liniowej. Są one mieszanką gier przygodowych i zręcznościowych. Przykładem jest Assassin'sCred czy Dishonored, Thief.



Rys.9- scena z gry Assassin's Creed 3

- Logiczne - gracz w tych grach rozwiązuje różnego typu zagadki, przeważnie starając się osiągnąć jak najlepszy wynik lub czas. Gra wymaga od graczy logicznego myślenia i sprytu, czasem odrobiny strategii. Do tych gier zalicza się: Hearthstone , Tetris, Puzzle Quest.



Rys.10-scena z gry Hearthstone

Obecnie coraz więcej gier miesza ze sobą różne cechy, dlatego trudno w niektórych przypadkach stwierdzić, do którego gatunku bezpośrednio należy.

1.3. Narzędzia do tworzenia gier

Do tworzenia gier komputerowych można używać wielu różnych programów. Jedne służą do robienia grafiki 3D (Blender, 3ds max) i 2D (Gimp, Photoshop), inne do obróbki dźwięku, ale najważniejsze są silniki graficzne. Przykładami takich silników są programy takie jak:

- Unity 3D
- Unreal Engine
- GameMaker: Studio
- Cryengine

Ja w mojej pracy korzystałem z Unity 3D. Jest to zintegrowane środowisko do tworzenia gier 2D i 3D, które można wykonywać na 21 platform, między innymi Windows, Linux, Android, IOS, PS4 i wiele innych. Wykorzystuje trzy rodzaje programowania C#, Java oraz Boo. Oprócz gier można też tworzyć różnego rodzaju interaktywne multimedia. Zawiera zaawansowane systemy, takie jak: UI, dzięki któremu można tworzyć różnego rodzaju panele, w tym też na obiektach, czy też Animator, w którym można łatwo i wygodnie tworzyć powiązania pomiędzy animacjami.

Dzięki każdemu z tych programów można stworzyć ciekawe i dobrze wyglądające gry. Wystarczy odpowiednia ilość czasu oraz praktyki.

Rozdział 2:

Praca nad grą "Plemię"

2.1. Założenia

Początkowe założenia, gdy zaczynałem pracę nad grą, były takie, by stworzyć grę strategiczną z widokiem TPP. System chciałem wzorować na grze Overlord, w której porusza się główną postacią, a wybrane jednostki podążają za postacią, wydając im proste polecenia, jak udanie się w wybrane miejsce, powrót czy utrzymywanie pozycji. Stylistykę świata chciałem przedstawić na zasadzie wojen plemiennych, podobną trochę do tej z gry "Populous: The Beginning", ale z lepszą grafiką. Grę przeznaczyłem na platformę Windows. Rozdzielczość natomiast jest do wyboru od 512x384 do 1600x900. Do grania nie potrzebny jest Internet. Gra jest przeznaczona dla jednego gracza. Sterowanie jest za pomocą klawiatury i myszy. Gracz może w każdej chwili zapisać grę oraz ją wczytać.



Rys.11-scena z gry Overlord

Rys.12- scena z gry Populous : The Beginning

2.2. Najważniejsze funkcjonalności

W grze steruje się postacią, która jest przywódcą plemienia. Można nią poruszać za pomocą klawiszy WASD, atakować wrogie jednostki lub chaty za pomocą klawisza E, który działa tylko wtedy, gdy wrogie obiekty są w zasięgu. Oprócz tego można go uzdrawiać (zwiększyć aktualną liczbę punktów zdrowia)

poprzez wciśnięcie przycisku samo leczenie. W przypadku śmierci gra się kończy porażką.

Oprócz przywódcy w grze występują trzy rodzaje jednostek:

- **Wojownik** - jednostka walcząca w zwarciu, gdy wykrywa wroga, atakuje go. Poza walką, gdy jest przydzielony do przywódcy, podąża za nim, gdy do obozu, przemieszcza się losowo pomiędzy wyznaczonymi punktami, natomiast przy braku przydziału stoi w miejscu. Wojownikom, przydzielonym do postaci gracza można wskazać miejsce i za pomocą LPM (wysyła pierwszą jednostkę z listy ze skryptu) kazać im udać się w danym kierunku, gdy nie wykrywają wrogów
- **Uzdrowiciel** - podchodzi do rannych jednostek w pobliżu i leczy ranną jednostkę (mającą aktualny stan zdrowia poniżej połowy ilości maksymalnej). Gdy tego nie wykonuje, jest przydzielony do przywódcy i podąża za nim, gdy do obozu przemieszcza się losowo pomiędzy wyznaczonymi punktami, natomiast przy braku przydziału stoi w miejscu.
- **Robotnik** - odnajduje najbliższe drzewo, pobiera z niego surowiec i znosi go do najbliższej chaty.

Jednostki po śmierci przewracają się i unoszą do góry. Gracz może tworzyć jednostki w chacie, przytrzymując przycisk CTRL otworzyć panel, w którym można wybrać i rozpocząć produkcję jednostki. Inaczej jest w przypadku wrogów, jednostki są produkowane automatycznie z wrogich obozów. Produkowane jednostki przez SI dzielą się na dwa rodzaje. Jedne są produkowane do obrony obozu, gdy jest ich mniej niż wybrana ilość przydzielona do obrony. Drugi rodzaj to jednostki wysyłane na wroga, których liczka rośnie po każdym ataku.

Zadania w grze są jasno określone. Po wykonaniu ich, gra przechodzi do ekranu informującego o nim. Rozgrywkę można w każdej chwili zapisać oraz wczytać.

2.3. Skrypty gry i wykorzystane narzędzia

Do stworzenia gry używałem silnika graficznego Unity3D oraz wbudowanego w niego zintegrowanego środowiska programistycznego MonoDevelop do pisania skryptów. Ważnymi elementami w tworzeniu gier są

skrypty, to one sterują wszystkimi funkcjami w grze. Poniżej wypisałem i opisałem wszystkie skrypty, które napisałem do stworzenia tej gry w kolejności alfabetycznej.

AI_Uzdrowiciela

Skrypt służy do sterowania jednostką o nazwie uzdrowiciel.

AI_Wojownika

Skrypt służy do sterowania jednostką o nazwie wojownik.

Atak_gracza

Skrypt, zajmujący się zadawaniem obrażeń (odjęcie punktów życia od aktualnego stanu zdrowia w celu jego wyzerowania) wrogiej jednostki przez gracza.

Atak_wojownika

Skrypt, zajmujący się zadawaniem obrażeń (odjęcie punktów życia od aktualnego stanu zdrowia w celu jego wyzerowania) wrogiej jednostce przez inną jednostkę.

blask_czaru

Przechowuje dane obiektu do blasku rzucanego czaru .

czary

Przechowuje dane o czarach.

czy_aktywny

Skrypt aktywuje przycisk do przejścia na wybraną scenę (sprawdza, czy wybrana jest scena).

czy_mysz_na_UI

Skrypt sprawdza, czy mysz znajduje się na elemencie UI.

do_kamery

Skrypt obraca elementy UI do kamery.

do_celu

Skrypt nakazuje jednostce, gdzie ma się udać.

do_sceny

Skrypt przechodzi do wybranej sceny.



Rys.13-scena z gry Plemię

dodanie_do_produkcji

Skrypt dodaje wybraną jednostkę do listy.

dowódctwo

Przechowuje listę danych obiektów.

ilość_surowce

Przechowuje surowiec do zebrania.

Jednostki_przynalazne_do_grupy

Skrypt wyszukuje i przechowuje listę jednostek z grupy.

Kamera_trzeciosobowa

Skrypt służy do obracania kamerą.

klasa_do_zapisu

Klasa zawierająca dane, służące do zapisywania danych, które są wykorzystywane przy zapisie i wczytywaniu gry.

Kolor

Skrypt przypasowuje właściwy materiał jednostkom w zależności od przynależnej grupy.

lista_kolorów

Przechowuje tablice materiałów do skryptu Kolor.

lokalizacja_zapis

Skrypt służy do wyboru lokalizacji pliku xml w zależności od przypisanej nazwy.

Mana

Skrypt przechowuje dane, dotyczące punktów many oraz zarządzania paskiem UI, który mu przynależy.

nadawanie_celu

Skrypt za pomocą myszy i zatwierdzeniu gracza wskazuje gdzie ma udać się jednostka, wchodząca w skład oddziału, którym dowodzi gracz. Cel dostaje jednostka, która jest pierwsza na liście.

nazwa

Skrypt sprawdza czy dany zapis istnieje. Jeżeli istnieje, wyświetla datę ostatniej modyfikacji, w przeciwnym wypadku wyświetla napis "wolny zapis".

obrona_obozu_SI

Skrypt sprawdza ilość jednostek w danym obozie i w przypadku, gdy jest ich za mało, co określony czas, tworzy nowy obiekt. W grze jest to jednostka o nazwie wojownik.

oddziały

Skrypt tworzy przyciski, do których przypisane są oddziały, dzięki którym gracz może zmieniać przynależność jednostek w zakresie numeru armii.

panel_budynku

Skrypt umożliwia graczowi aktywowanie panelu budynku w pobliżu.

patrol

Skrypt sprawia, że jednostki przypisane do obozu przemieszczają się pomiędzy punktami z listy.

pauza

Skrypt pozwala graczowi aktywować i dezaktywować panel pauzy.

playermove

Skrypt pozwala graczowi poruszać postacią.

podążanie

Skrypt sprawia, że jednostki podążają za czempionem lub postacią gracza.

produkcja

Skrypt odpowiada za pojawianie się wybranych do produkcji jednostek i przypisanie im odpowiednich danych.

przydzielony_obiekt

Skrypt przydziela obiekt, z którego oraz do którego zostaną przydzielone jednostki w panelu "Panel zarządzania armią".

przynależność

Skrypt przechowuje dane o przydziale do grupy oraz nr_armii.

przypisywanie_ilości

Skrypt przypisuje liczbę do danego przycisku, w zależności od rodzaju przypisanej jednostki i wybranego oddziału.



Rys.14-scena z gry Plemię

punkty_patrolowe

Skrypt przechowuje listę punktów patrolowych dla danego obiektu.

samoleczenie

Skrypt pozwala graczowi użyć funkcji leczenia ze skryptu czary z argumentami postać .

skład_armi

Skrypt przechowuje listy jednostek, należących do danego oddziału.

Surowce

Skrypt przechowuje i wyświetla ilość surowców.

szukanie_rannego

Skrypt wyszukuje jednostkę, której ilość obecnych punktów życia jest poniżej połowy oraz która należy do tej samej grupy co uzdrowiciel.

u_celu

Skrypt pomocniczy do skryptu "zadania", sprawdzający czy postać jest w pobliżu celu.

uzdrowienie_sojusznika

Skrypt przemieszcza jednostkę uzdrowiciela w kierunku celu oraz gdy dość się zbliży, używa funkcji leczenia ze skryptu czary z argumentami jednostki uzdrowiciela oraz celu.

wczytywanie_gry

Skrypt przerabia plik xml na klasa_do_zapisu i przypisuje zapisane w nim dane do obiektów.



Rys.15-scena z gry Plemię

wczytywanie_z_innej_sceny

Skrypt pomocniczy do skryptu wczytywanie_gry, umożliwiający aktywację skryptu po przejściu z innej sceny.

wybór_oddziału

Skrypt przechowuje, z którego i do którego zmieniana jest przynależność w panelu zarządzania armią.

wybór_sceny

Skrypt przypisuje nazwę sceny do późniejszego przejścia oraz wyświetla opis.

wyjście_z_gry

Skrypt wyłącza grę.

wykrywanie_wroga

Skrypt sprawdza, czy w zasięgu jednostki nie ma wroga.

wysyłanie_oddziału

Skrypt tworzy oddziały SI, które wysyła w kierunku wybranego celu.

wyłącz_emission

Skrypt po upływie określonego czasu wyłącza emisję w obiekcie particle system.

zadania

Skrypt zawiera wymogi, wyświetla informacje o nich oraz gdy są spełnione odpowiednio przechodzi do odpowiedniej sceny z informacją.

Zapis_gry

Skrypt pobiera i zapisuje do pliku xml dane obiektów, potrzebne do późniejszego odczytu.

zbieranie_sórowców

Skrypt steruje jednostką robotnik.

Życie

Skrypt przechowuje informacje o stanie życia obiektów .

Oprócz nich korzystałem też z wielu komponentów, które są wbudowaną grupą skryptów silnika graficznego. Najważniejsze, których używałem, wypisałem i opisałem poniżej.

Animator

Komponent zarządza animacjami obiektu , rozpisuje przejścia pomiędzy poszczególnymi animacjami, ustalając przejścia oraz warunki pomiędzy nimi.

Collider

Jest to grupa komponentów, odpowiedzialna za kolizje pomiędzy obiektami. W przypadku jego braku inne obiekty nie są w stanie reagować na obiekt.

NavMesh Agent

Służy do przemieszczania obiektu po NavMesh poprzez wyznaczenie po nim najkrótszej drogi do celu. Opiera się na zasadach A*.

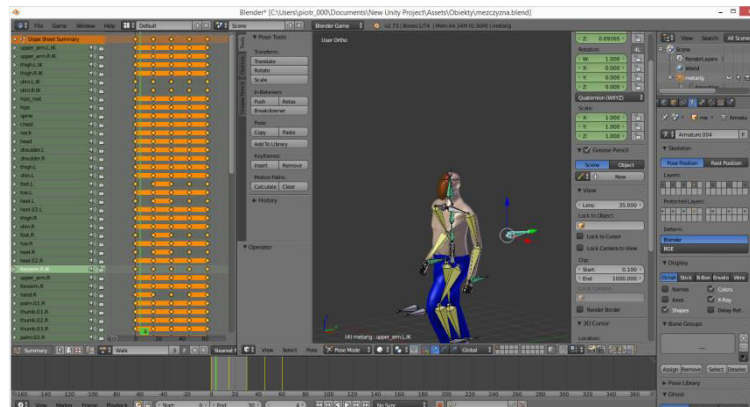
Transform

Ten komponent zawiera każdy obiekt. Służy on do określenia położenia obiektu względem rodzica (w przypadku jego braku scena jest rodzicem), rotacji obiektu oraz jego skali.

Żadna gra nie obejdzie się bez obiektów graficznych (może z wyjątkiem gier tekstowych, które w obecnych czasach używają co najwyżej tylko zdjęć bądź rysunków) czy to 3D czy 2D. Obiekty 2D rysowane są na polach o kształcie prostokąta, po wykonaniu ich oznacza się kanałem alfa pixeli, które mają być przezroczyste. Animacje w obiektach 2D polegają na podstawianiu innego obiektu, wyglądającego tak samo jak poprzedni obiekt, ale w innej pozycji.

Inaczej jest natomiast z obiektami 3D, najpierw tworzy się model poprzez umieszczanie punktów w przestrzeni, łączenie ich w odcinki i odpowiednie wypełnienie przestrzeni pomiędzy nimi. Następnie nakłada się na nie tekstury, które określają kolory pixeli w modelu (textury są obiektami 2D). Inaczej

wykonuje się animację. Zaczyna się od stworzenia "szkieletu" obiektu z kości, do których przypisuje się płaszczyzny modelu. Przypisane płaszczyzny będą zmieniać położenie w zależności od ruchu kości. Następnie wybiera się miejsce na osi czasu i znów odpowiednio ustawia się kości. Gdy już umieści się odpowiednią ilość takich punktów na osi, można wtedy wyręderować animację (kości będą się przemieszczać do wybranych punktów w wyznaczonym przez oś czasu okresie, czyli im dane punkty są bliżej, tym szybciej będą się przemieszczać) i ją zapisać. Ja używałem do tego programu Blender 3D.



Rys.16-zrzut obrazu z Blendera pokazujący model postaci z gry Plemię

Rozdział 3:

Testy gry

3.1. Kontrola funkcjonalności

Gra została przetestowana na kilku komputerach i na wszystkich działała poprawnie. Przed rozpoczęciem każdej gry można zmienić rozdzielczość gry na taką, jaka mu najbardziej odpowiada do ekranu. Minimalnych wymagań sprzętowych nie jestem w stanie jednoznacznie określić, ale bez problemu działa na sprzęcie o parametrach(jest to najslabszy komputer na jakim testowałem grę) : Procesor Intel Core i5 2,67GHz , RAM 4GB, NVIDIA GeForce GTX 260.

3.2. Subiektywna ocena grywalności

Gra nie jest specjalnie trudna. Nie jest może szczególnie wciągająca, ale działa poprawnie. Pierwszą planszę wykonałem w odpowiednich rozmiarach. Rozłożyłem ilość wrogów odpowiednio do posiadanych jednostek. Druga plansza natomiast jest trochę za duża oraz trudniej jest znaleźć cele misji. Czasami trudno jest trafić postaci gracza w chatę.



Rys.17-scena z gry Plemię

3.3. Sugestie do zmian i udoskonaleń

W grze można sporo pozmienić. Można poprawić modele, nałożyć bardziej szczegółowe tekstury. Dodać jednostkom rozkaz odwrotu by podczas walki móc wycofać jednostki. Stworzyć jednostki o nowych zdolnościach, np. łuczników czy magów. Przydało by się stworzyć nowe czary, które będą zadawać obrażenia lub osłabiać przeciwników. Można poprawić animację poprzez poprawienie przechodzenia pomiędzy animacjami. Ciekawie byłoby również stworzyć fabułę do gry czy urozmaicić system walki poprzez dodać elementy nosowości, np. nie trafienie w cel, blok czy unik. Nie zaszkodzi też urozmaicić otoczenie czyli dodać obiekty niezależne, które będą tworzyć tło.

Zakończenie

W grze zostały zrealizowane podstawowe cele, które sobie założyłem. Wyniknęło jednak kilka trudności, związanych z animacją. W momencie śmierci jednostki powinny się przewracać, a niestety, z powodu wcześniejszego błędu, obracają się względem środka obiektu zamiast u jego spodu, dlatego po śmierci jednostki unoszą się w górę.

W grze nie zaszkodziły by zmiany takie jak poprawienie grafiki, dodanie jednostkom rozkaz odwrotu, stworzenie nowych jednostek czy czarów. Oprócz tego można zoptymalizować animację, stworzyć fabułę, urozmaicić system walki i urozmaicić otoczenie.

Summary

In my work I created game. I call "Plemię" who is a strategy in TPP. In first chapter is about game and tools to create game. Chapter two says about my game and tools which I used to created game. In last chapter is about game test and possible changes in game.

Spis literatury

"Projektowanie gier w środowisku Unity 3.x" WillGoldstone

"Blender Mistrzowskie animacje 3D" Tony Mullen

<http://unity3d.com/>

<http://polskikursblendera.pl/>

<http://pl.wikipedia.org/>

<http://www.gry-online.pl/>

Spis rysunków

Rys.1- scena z gry RaymanOrigins

Rys.2-scena z gry Wiedźmin

Rys.3-scena z gry Cywilizacja V

Rys.4-scena z gry The Crew

Rys.5-scena z gry FIFA 2014

Rys.6- scena z gry World of Warcraft

Rys.7-scena z gry Black Mirror

Rys.8-scena z gry The Sims 4

Rys.9- scena z gry Assassin's Creed 3

Rys.10-scena z gry Hearthstone

Rys.11-scena z gry Overlord

Rys.12- scena z gry Populous : The Beginning

Rys.13-scena z gry Plemię

Rys.14-scena z gry Plemię

Rys.15-scena z gry Plemię

Rys.16-zrzut obrazu z Blendera pokazujący model postaci z gry Plemię

Rys.17-scena z gry Plemię