

Projekt i implementacja gry komputerowej z wykorzystaniem technologii Microsoft DirectX 11

Krzysztof Marciniak
106574

Piotr Przybysz
106602

Mikołaj Szychowiak
106580

Ryszard Wojtkowiak
106609

22 stycznia 2015

Spis treści

1	Wstęp	3
1.1	Cel i zakres pracy	3
2	Przegląd zagadnień teoretycznych	3
3	Przegląd narzędzi	3
3.1	Własny silnik graficzny w języku C++	3
3.2	Unreal Engine 4	3
3.3	Unity	3
3.4	OGRE	3
3.5	CryEngine 3	3
3.6	Unreal Engine 3	3
3.7	Unreal Development Kit (UDK)	3
3.8	Maya	3
3.9	3DS Max	3
3.10	Blender	3
3.11	Adobe Flash - Scaleform	3
4	Praca własna	3
4.1	Projektowanie	3
4.1.1	Projekt silnika graficznego	3
4.1.2	Modele 3D	3
4.1.3	Projekt poziomego	3
4.2	Implementacja	3
4.2.1	Wykorzystanie technologii DirectX11 w UDK	3
4.2.2	Zarządzanie projektem	3
5	Podsumowanie	4
6	Literatura	4
7	Dodatki	4

1 Wstęp

1.1 Cel i zakres pracy

2 Przegląd zagadnień teoretycznych

3 Przegląd narzędzi

3.1 Własny silnik graficzny w języku C++

3.2 Unreal Engine 4

3.3 Unity

3.4 OGRE

3.5 CryEngine 3

3.6 Unreal Engine 3

3.7 Unreal Development Kit (UDK)

3.8 Maya

3.9 3DS Max

3.10 Blender

3.11 Adobe Flash - Scaleform

4 Praca własna

4.1 Projektowanie

4.1.1 Projekt silnika graficznego

4.1.2 Modele 3D

4.1.3 Projekt poziomu

4.2 Implementacja

4.2.1 Wykorzystanie technologii DirectX11 w UDK

4.2.2 Zarządzanie projektem

W celu sprawnej organizacji pracy w zespole wykorzystano metody zarządzania projektami.

Początkowo użyto modelu kaskadowego. Jego zaletą jest sekwencyjność, pozwalająca oddzielić procesy analizy problemu, projektowania, implementacji oraz testowania i późniejszego utrzymania projektu. Metoda Waterfall, wykorzystująca ten model, nie jest jednak pozbawiona wad. Dotyczą one głównie dużych projektów, a więc nie miały miejsca w przypadku oprogramowania tworzonego w czteroosobowym zespole programistów. Korzystając z tej metody oszczędza się czas na planowaniu, faza ta zajmuje zaledwie 1/4 czasu pracy nad projektem. Rezultat końcowy zostaje ustalony jeszcze przed rozpoczęciem implementacji, podobnie jak poszczególne przyrosty implementacji. Każdy przyrost miał trwać 2 tygodnie i zawierał określone zadania. Rezultatem końcowym był produkt posiadający wartość biznesową, w tym przypadku - w pełni działająca gra. Co istotne, wartość biznesową produkt miał zyskać dopiero w przedostatnim przyroście.

W niedługim czasie po zaplanowaniu prac nad projektem okazało się, że metoda ta nie jest wystarczająca, zaczęło się pojawiać opóźnienie w pracach, które mogło spowodować pogorszenie jakości produktu. Zdecydowano się więc skorzystać z metod zwinnych. Metodyki Agile charakteryzują się stałą jakością, a sterowane są zakresem. W omówionej wcześniej metodzie Waterfall zakres był stały, zmieniała się jedynie jakość, a chęć utrzymania wysokiej jakości powodowała opóźnienie względem planu.

Ostatecznie zastosowano metodykę zwinną zbliżoną do metody Scrum. Zadania podzielono na część dotyczącą rozpoznania danego zagadnienia i część implementacyjną. Realizowane były tygodniowe sprinty (przyrosty). Na koniec każdego z nich otrzymywany był prototyp posiadający pewną funkcjonalność. Ze względu na wcześniejszy nieudany eksperyment z metodą Waterfall wartość biznesową projekt zyskał dopiero po kilku przyrostach. Oznacza to, że pierwsze efekty nie były stysfakcjonujące, jednak prezentowały postęp prac. Każdy sprint rozpoczynał się spotkaniem zespołu na którym omówiono pozostałe zadania oraz zaplanowano jakie funkcjonalności będą implementowane w kolejnym przyroście. W efekcie udało nam się zachować jakość wykonania, modyfikując nieznacznie zakres dostępnych funkcjonalności.

5 Podsumowanie

6 Literatura

7 Dodatki