POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

Wydzial Informatyki

Katedra Sieci Komputerowych
Sieci Urządzeń Moblinych

Kamil Warpechowski
Nr albumu 10709

Praca inżynierska
Promotor:
dr inż. Michał Tomaszewski

tutaj będzie zajebisty tytuł

Warszawa, 18 maja 2016

Spis treści

1 Cel pracy

Celem niniejszej pracy jest stworzenie platformy do budowania gier oraz interaktywnych animacji prezentowanej za pomocą rozszerzonej rzeczywistości sterowaniej za pomocą zdalnego kontrolera.

Przykłady zastosowanie zestawu aplikacji:

- Prezentacje przestrzeni architektonicznych
- Rozrywka
- Reklama miejscach użyteczności publicznych (np. centra handlowe)

2 Rozszerzona rzeczywistość

Rysunek 1: Wizualizacja Microsoft HoloLens
źródło: https://www.microsoft.com/microsoft-hololens/en-us/why-hololens

3 Wykorzystane technologie

3.1 Unity

Unity jest obecnie najpopularniejszą platformą do tworzenia gier na wiele platform.

3.1.1 Dlaczego Unity

Najnowsza wersja posiada natywne wsparcie do rozszerzonej oraz wirtualnej rzeczywistości. Narzędzie te posiada prosty, ergonomiczny interfejs co ułatwia pracę.
Bardzo pomocnym dodatkiem do narzędzia jest „Assets Store". Jest to wirtualny sklep z komonentami do tworzenia gry. W projekcie zastosowałem tekstury i obiekty 3d pochodzące z tego źródła.

3.1.2 Alternatywne rozwiązania

	Unity	Unreal Engine
Wsparcie języków programowania	C#, JavaScript, Boo	c++
Obsługa wielu ekranów	Tak	Nie

Tablica 1: Porównanie silników gier

3.1.3 Wybór języka programowania

Środowisko Unity wspiera obsługę skryptów (animacje oraz logika biznesowa) w kilku językach programowania: C#, UnityScript (zmodyfikowana wersja JavaScript) oraz w przeszłości Boo. Podjąłem decyzję, by w projekcie użyć język C#, gdyż ów język jest najbardziej stabilny, posiada najbardziej rozbudowaną dokumentację oraz jest to najbardziej popularny język w specjalistycznej literaturze. Dodatkowym udogodnieniem jest to, iż język posiada wiele wbudowanych klas (np. do obsługi połączeń TCP) oraz niezliczoną ilość zewnętrznych bibliotek.

3.1.4 Unity IDE

Środowisko Unity jest multiplatformowe. Aplikacje można używać na dowolnym sytemie operacyjnym. Jednakże edycja skryptów odbywa się za pomocą zewnętrznego narzędzia. W systemie Mac OS jest to MonoDevelop, natomiast w systemie Windows jest to VisualStudio w wersji Community. Opisywana aplikacja początkowo była tworzona na systemie Mac OS, jednakże kołopoty ze środowiskiem MonoDevelop spowodwały decyzje o przeniesieniu środowiska na system Windows. Subiektywnie mogę stwierdzić, że stabilność oraz komfort pracy jest dużo lepszy w systemie Windows. Dodatkową alternatywą dla MonoDevelop może okazać się Visual Studio Code. Jest to prosty multiplatformowy edytor posiadający obsługę języka C# .

3.1.5 Licencja i koszty

Unity jest zamkniętym, licencjonowanym oprogramowaniem. Darmowa wersja (Personal Editon) pozwala na nielimitowane użycie, jednakże jest to okrojona edycja. Szersze informacje o ograniczeniach wersji Personal zawarte są w kolejnych rozdziałach. Licencja pozwala na komercyjne użycie przy limicie zarobków na poziomie stu tysięcy dolarów. Komercyjna wersja (Professional Edition) jest płatna w modelu subskrybcyjnym (75 dolarów za miesiąc)[1]. Na potrzeby opisywanego projektu zasotosowano Unity w wersji Personal Edition.

3.2 Android

3.3 Połączenie sieciowe

Największym wyzwaniem było stworzenie dwukierunkowego protokołu komunikacyjnego pomiędzy serwerem (aplikacja napisana w środowisku Unity) oraz dowolnym kontrolerem lub w przyszłości innym urządzeniem wysłającym dane do aplikacji.

3.3.1 UNET

Unity wspiera natywną obsługę multiplayer - UNET, jednakże jest to zamknięty protokół. Komunikacja możliwa jest tylko pomiędzy aplikacjami stworzonymi w tym środowisku.

[1] https://store.unity3d.com/subscribe

3.3.2 SOAP

3.3.3 REST

3.3.4 RTP

3.3.5 TCP

4 Aplikacja główna

4.1 Świat gry

4.2 Aktorzy

4.3 Kamery

Platforma Unity wspiera do 8 wirtualnych kamer [2]. Każda z tych kamer może być prezentowana na oddzielnym fizycznym ekranie.

4.4 Logika biznesowa

4.5 Serwer komunikacyjny

5 Aplikacja mobilna - kontroler

6 Środowisko uruchomieniowe

Powyżej opisana aplikacja została uruchomiona testowo w labolatorium na Polsko-Japońskiej Akademii Technik Komputerowych.

6.1 Serwer

Serwer został uruchomiony na komputerze przenośnym(laptop) posiadającym kartę graficzną umożliwiajacą podłączenie dwóch zewnętrznych ekranów - projektorów. Pierwszy z nich został połączony za pomocą złącza cyfrowego HDMI, natomiast drugi łączem DVI.

6.2 Aplikacja mobilna - kontroler

7 Dalszy rozwój

[2] http://docs.unity3d.com/Manual/MultiDisplay.html

Spis rysunków

Spis tablic

Literatura

[1] H. Partl: *German TEX*, TUGboat Vol. 9„ No. 1 ('88)