

**WYDZIAŁ INŻYNIERII METALI I INFROMATYKI STOSOWANEJ**

KATEDRA INFORMATYKI STOSOWANIEJ I MODELOWANIA

**PRACA INŻYNIERSKA**

Wirtualny spacer po terenie AGH z wykorzystaniem technologii VR.

*Virtual walking around the AGH using VR technology.*

Autorzy: *Paweł Brzoza, Marcin Szumlański*

Opiekun pracy: *dr inż. Tomasz Dębiński*

Kierunek studiów: *Informatyka Stosowana*

*Kraków 2018*

Oświadczam, świadomy (-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem (-am) osobiście i samodzielnie i że nie korzystałem (-am) ze źródeł innych niż wymienione w pracy.”

Paweł Brzoza

Marcin Szumlański

***Spis treści:***

1. Wstęp (wspólnie)
   1. Cele i zakres pracy
   2. Streszczenie kolejnych rozdziałów
2. Świat wirtualny, a rzeczywisty (P. Brzoza)
   1. Niezbędne definicje
   2. Historia
   3. Obszary zastosowań
   4. Przedstawiany typ wirtualnej rzeczywistości
3. Przegląd dostępnych rozwiązań technologicznych (P. Brzoza)
   1. Przegląd rynku
   2. Możliwości zastosowania
4. Opis wybranych technologii (P. Brzoza)
   1. Niezbędne elementy instalacyjne
      1. Unity3D
      2. Android Studio SDK
      3. Java Development Kit
   2. Smartphone do obsługi aplikacji VR
      1. Xiaomi Redmi Note 4X
      2. Sony Xperia Z1
   3. Okulary VR
      1. Google Cardboard
      2. FiiT VR 2S
5. Praca z silnikiem graficznym Unity (P. Brzoza)
   1. Składniki projektu
   2. Struktura sceny
   3. Poszczególne komponenty środowiska
   4. Elementy biblioteki Google VR SDK
   5. Pozostałe elementy
   6. Implementacja mechaniki widoku
   7. Implementacja efektów specjalnych
   8. Porównanie różnych jakości teksturowania oraz kompresji tekstur.
6. Export aplikacji na urządzenia z systemem Android (P. Brzoza)
   1. Niezbędne ustawienia
   2. Budowanie aplikacji
7. Zdjęcia sferyczne – pozyskiwanie i właściwości (M. Szumlański)
8. Implementacja aplikacji przy użyciu języka skryptowego JavaScript (M. Szumlański)
9. Porównanie wydajności dwóch aplikacji. (M. Szumlański)
10. Badania „user experience” (M. Szumlański)
11. Perspektywy rozwoju i przyszłość wirtualnej/rozszerzonej rzeczywistości (M. Szumlański)
12. Podsumowanie (wspólnie)
    1. Rozwiązanie podziału pracy - prezentowana praca inżynierska była wykonywana przez dwie osoby, wobec tego zastosowany został podział pracy tak, aby zarówno jedna jak i druga osoba mogła w pełni zaprezentować swoje umiejętności inżynierskie.
13. Spis ilustracji
14. Bibliografia
15. **Wstęp**

Na przestrzeni ostatnich lat nie sposób nie zauważyć ogromnego postępu w przedstawianej dziedzinie tj. wirtualnej rzeczywistości (ang. virtual reality). Dzięki łatwości użycia i coraz większej dostępności sprzętu do obsługi VR, ludzie zaczynają eksperymentować z prostymi aplikacjami czy filmikami dającymi szansę przetestować wirtualny świat. Przedmiotem niniejszej pracy jest rozpoznanie możliwości dostępnych na rynku związanych z tematem, a następnie implementacja aplikacji do wirtualnego „spaceru” po terenie AGH na platformę Android oraz wersję przeglądarkową. Przedstawiony zostanie proces budowania aplikacji w środowisku Unity jak i poza nim wykorzystując czysty JavaScript, wszystko to wspomagając się API od Google. Przeanalizowane zostaną również różnice pomiędzy różnymi typami wirtualnej rzeczywistości, a na koniec prezentacja wyników, czyli porównanie możliwości pomiędzy dwoma rodzajami implementacjami, wyciągnięcie odpowiednich wniosków dotyczących m.in. wykorzystania mocy obliczeniowej sprzętu czy wygody użycia.

* 1. **Cele i zakres pracy**

Głównym celem pracy jest zapoznanie się z technologią VR, następnie rozpoznanie rozwiązań aktualnie dostępnych na rynku i wybranie możliwie niskobudżetowego zestawu, który w pełni umożliwi korzystanie z przedstawianej technologii. Umożliwi to wybranie odpowiedniej i stworzenie aplikacji.

Implementacja programu będzie występować w 2 wersjach. Pierwsza odbędzie się za pomocą silnika graficznego Unity, a druga przy pomocy języka skryptowego JavaScript, wszystko po to aby dobrze zbadać możliwości i uzyskać jak najlepszy efekt końcowy.

* 1. **Streszczenie kolejnych rozdziałów**

1. **Świat wirtualny, a rzeczywisty** (P. Brzoza)
2. **Przegląd dostępnych rozwiązań** (P. Brzoza)
3. **Opis wybranych technologii** (P. Brzoza)
   1. **Niezbędne elementy instalacyjne**
      1. **Unity3D**

Zintegrowane środowisko programistyczne (ang. IDE, Integrated, Development Environment), jakim jest Unity, pozwala tworzyć nie tylko gry komputerowe (z czym może być kojarzone), ale również pozwala na kreowanie różnych wizualizacji, animacji czy materiałów interaktywnych. Wszystko to może zostać stworzone na płaszczyźnie dwuwymiarowej lub trójwymiarowej. W przedstawianym środowisku, jak w każdym innym profesjonalnym programie tego typu, można tworzyć, edytować, testować, bądź „konserwować” kod programu. W tym wypadku są to głównie skrypty pisane w języku programowania C#.

Jedną z wielu zalet silnika Unity jest jego elastyczność. Odpowiednio skonfigurowany projekt pozwala na jego eksport na ponad 10 różnych platform:

- urządzenia mobilne – Android, IOS, Windows Phone

- technologie webowe – WebGL, Facebook

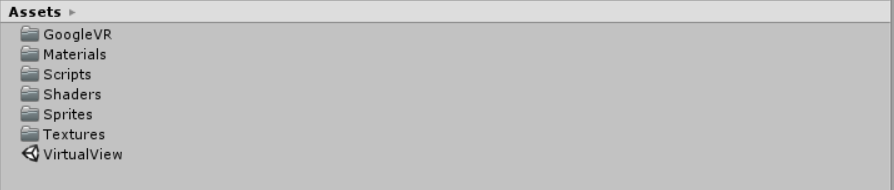
- komputery stacjonarne – Windows, Mac, Linux

- konsole do gier – PlayStation 4, PS Vita, Xbox One

1. **Praca z silnikiem graficznym Unity** (P. Brzoza)

Powstanie projektu rozpoczyna się od stworzenia odpowiedniej struktury.

* 1. **Struktura projektu**



Strukturę projektu przedstawia folder Assets (co w dosłownym tłumaczeniu w języku polskim może oznaczać wartości użyteczne, dla wygody stosowania w dalszej części pracy używana będzie angielska wersja), w którego skład wchodzą poszczególne pod foldery zawierające:

- GoogleVR – zaimportowane elementy z biblioteki GoogleVR SDK, będące dodatkowym wsparciem dla projektu, ważne skrypty to: GvrEditorEmulator, GvrControllerMain, GvrEventSystem i GvrReticlePointer, które dokładniej zostaną opisane niżej,

- Materials – stworzone „materiały”, czyli elementy graficzne kontrolujące modele 3D, zawierające w sobie niżej opisane Shadery

- Scripts – skrypty, odpowiadające głównie za mechanikę programu

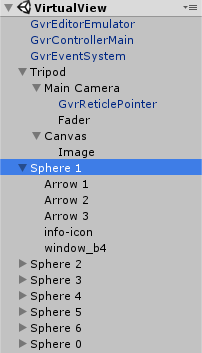
- Shaders – folder zawierający tzw. Shadery, czyli kod napisany w języku HLSL (High-Level Shading Language) lub CG (C for Graphics), które są do siebie bardzo zbliżone. Służą między innymi do opisu sposobu teksturowania modeli.

- Sprites – elementy, które rozszerzające rzeczywistość takie jak: strzałki wskazujące kierunek, w którą stronę się udamy, ikony o informacjach, okienka informacji o budynkach oraz wskaźniki w postaci okręgów ułatwiające określenie czasu w momencie wykonywania tzw. GazeClicku

- Textures – tekstury, czyli w naszym wypadku zdjęcia sferyczne

- VirtualView – scena zawierająca obiekty „gry”

**5.2. Struktura sceny:**



Na strukturę sceny składają się kolejno:

- Cztery skrypty od GoogoleVR SDK:

* GvrEditorEmulator –
* GvrControllerMain –
* GvrEventSystem –
* GvrReticlePointer –

- Statyw (ang. Tripod) – pusty objekt, zawierający w sobie kamerę główną (ang. Main Camera, Rys.x).

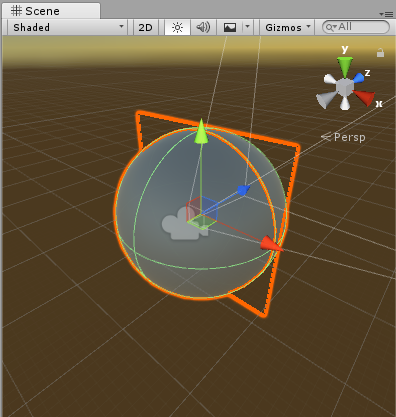
Ta z kolei zawiera w sobie ostatni z kolei opisany wyżej skrypt oraz obiekt 3D typu czworokątnego (ang. Quad, Rys.x) o nazwie Fader, który posłuży do ciekawszych przejść widoków kamery (dokładniej opisane zostanie niżej, przy przeglądzie skryptów).

Statyw zawiera, również strefę Canvas, gdzie powinny znajdować się wszystkie elementy związane z interfejsem użytkownika (ang. UI). W naszym przypadku jest to komponent typu Image odpowiadający za GazeClick (dokładniej opisany niżej)

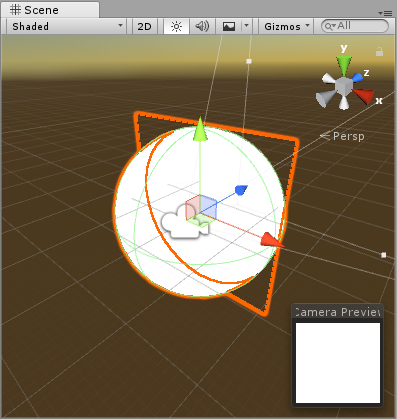
- Siedem sfer – z czego każda, zawiera strzałki, oznaczające kierunki przemieszczania oraz ikony, dzięki którym, po najechaniu na nie wyświetlana zostaje informacja o danym obiekcie.

Wszystkie sfery tworzone są następujący sposób:

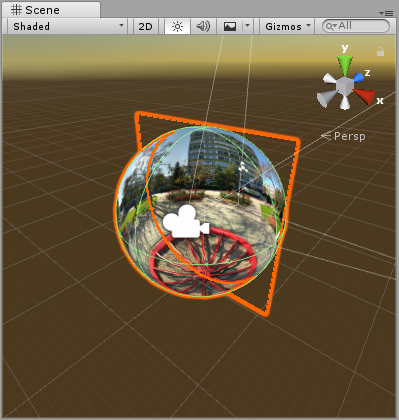
- do naszego statywu z kamerą dodatkowo utworzono obiekt 3D typu sfera (ang. Sphere, Rys.x)



- po utworzeniu sfery dodano odpowiedni shader o nazwie Insideout, który umożliwi teksturowanie sfery oraz nie będzie wymagał dodatkowego oświetlenia, domyślne renderujący siatkę na biało (Rys.x)



- stworzono odpowiedni materiał za pomocą, wcześniej tekstur czyli zdjęć w formie 360 i dodano go do sfery co daje następujący efekt (Rys.x)



**Bibliografia**

Paul Neumann, [dostęp online: 19.09.2017], <http://paulneumann.blox.pl/2011/09/Swiat-realny-a-swiat-rzeczywisty.html>

<https://unity3d.com/unity>

<https://unity3d.com/learn/tutorials/topics/graphics/gentle-introduction-shaders>

<https://docs.unity3d.com/Manual/SL-ShadingLanguage.html>

https://docs.unity3d.com/Manual/UICanvas.html