Rok akademicki: 2017/2018

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

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. Wprowadzenie

Na przestrzeni ostatnich lat nie sposób nie zauważyć ogromnego postępu w przedstawianej dziedzinie tj. wirtualnej rzeczywistości (ang. VR - 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 było rozpoznanie możliwości dostępnych na rynku związanych z tematem, a następnie implementacja aplikacji do wirtualnego „spaceru" po terenie AGH na platformę Android oraz wersję dostępną w przeglądarkach internetowych. Przeanalizowano różnice pomiędzy różnymi typami wirtualnej rzeczywistości i przedstawiono jak działa omawiana technologia. Wspomniano, również o procesie pozyskiwania zdjęć sferycznych, następnie zabrano się na stworzenie dwóch aplikacji. Przedstawiono proces budowania „gry" w środowisku Unity jak i poza nim wykorzystując czysty JavaScript, wszystko to wspomagając się SDK (ang. software development kit) od Google. Zaprezentowano wyniki badania doświadczeń wynikających z przetestowania aplikacji na grupie 21 ludzi. Pod koniec zostały przedstawione perspektywy jakie daje owa technologii oraz kilka ciekawych projektów, które maja szansę zabłysnąć w niedalekiej przyszłości. Całość zakończyło podsumowanie, w którym autorzy postarali się wyciągnąć odpowiednie wnioski z realizowanej pracy.

1.1. Cele i zakres pracy

Głównym celem było 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. Umożliwiło to wybranie odpowiedniego sprzętu oraz oprogramowania do tworzenia wirtualnej rzeczywistości po czym, stworzono aplikacje.

Implementacja programu występuje w 2 wersjach. Pierwsza, zaimplementowana za pomocą silnika graficznego Unity3D, a druga przy pomocy języka skryptowego JavaScript, wszystko po to, aby dobrze zbadać możliwości, uzyskać jak najlepszy efekt końcowy i zdecydować, która technologia w naszej opinii jest lepsza.

1.2. Streszczenie kolejnych rozdziałów

Drugi rozdział jest swoistym wprowadzeniem w różne rodzaje wirtualnej rzeczywistości. Opisuje szczególne typy, które da się wyróżnić oraz przedstawia ciekawostki z branży.

Rozdział trzeci to wyjaśnienie na czym polega technologia VR, w jaki sposób możliwe jest doświadczanie złudzenia bytu w innej rzeczywistości.

Czwarty, dotyczy uzasadnienia wyboru określonego sprzętu do obsługi technologii VR. Zawiera specyfikacje techniczną urządzeń oraz prezentację wybranych gogli.

Esencją rozdziału piątego jest sposób w jaki uzyskiwano fotografie służące w dalszym etapie pracy do wygenerowania wirtualnej rzeczywistości. Przedstawiono aparat, którym robiono zdjęcia, a także zawiera dywagacje na temat ulepszenia procesu pozyskiwania zdjęć.

Rozdział szósty przedstawia elementy środowiska Unity, najważniejsze aspekty związane z oprogramowaniem, strukturę projektu, cały proces tworzenia aplikacji oraz jej końcowy efekt. Opisuje również użyte biblioteki, funkcjonalności na podstawie kodu oraz eksport na inne urządzenie.

Siódmy, to bardzo podobna aplikacja do napisanej wcześniej w Unity, lecz stworzona za pomocą innych bibliotek i nakierowana na wersję webową. Wykorzystuje te same zdjęcia sferyczne oraz bazuje na tym samym koncepcie tworzenia. Dodatkowo testowana na domowym serwerze, udostępniającym aplikację w sieci publicznej.

W rozdziale ósmym po krótce została opisana metodologia badań, przedstawienie dwóch części badawczych oraz zaprezentowane zostały wyniki tzw. „User Experience” za pomocą wykresów oraz chmur słów. Na koniec wyciągnięcie wniosków.

Dziewiąty daje pogląd na perspektywy rozwoju technologii wirtualnej rzeczywistości. Autor, na podstawie znalezionych artykułów w sieci, przedstawia nowe, ciekawe projekty, które maja szanse w przyszłości zrewolucjonizować świat.

Cały sens niniejszej pracy inżynierski zawiera się w rozdziale dziesiątym. Stanowi podsumowanie i wyciągnięte wnioski z osiągniętych rezultatów.

2. Rzeczywistość wirtualna, rozszerzona i mieszana na przykładzie dostępnych rozwiązań. (P. Brzoza)

2.1. Rzeczywistość wirtualna

Wszystkie technologie „przerabiające" rzeczywistość zmieniają w pewnym sensie sposób w jakim ją postrzegamy, jednakże rzeczywistość wirtualna (ang. VR – Virtual Reality) całkowicie zmienia środowisko, które nas otacza. Merriam-Webster definiuje rzeczywistość wirtualną[1] jako sztuczne środowisko, które jest odbierane za pomocą bodźców jak wzrok/słuch. Jest Dostarczone przez komputer w którym akcje użytkownika częściowo determinują co stanie się w tym otoczeniu.

To, w jaki sposób wchodzimy w interakcję z otoczeniem zależy jednak sporo od platformy której używamy. Niektóre zestawy VR zostały zaprojektowane do używania siedząc w jednym miejscu i ruszając się za pomocą kontrolera, tak jak np. w grach komputerowych. Różnicą jest tutaj jednak to, że ekran jest przyczepiony do głowy użytkownika i zasłania jego dużą część pola widzenia „zanurzając" go w wirtualnym 360 stopniowym świecie.

Oculus Rift jest dobrze znany jako jeden z najbardziej popularnych zestawów VR aktualnie dostępnych masowo, ale wymaga zarówno komputera (do którego podłączone są gogle) jak i osobnego kontrolera do działania. Efektywnie, jest to dający złudzenie zanurzenia się w rzeczywistości wirtualnej ekran, który zakłada się na głowę. Podobnie działa PlayStation VR oraz HTC Vive – są to zbliżone technologicznie rozwiązania.

Z drugiej strony mamy np. Google Cardboard i Samsung Gear VR (jakościowo solidniejszy), które oferują podobne doświadczenia – niższej jakości, ale również niższej ceny. Plusem tych urządzeń jest z pewnością to że nie muszą być okablowane i podłączone do komputera. Zamiast tego do stworzenia doświadczenia VR używają smartfona podłączonego do gogli, które zakładamy na głowę.

Ze względu na tą prostotę i niski koszt zdecydowaliśmy się wybrać tą technologię do stworzenia w nim naszego projektu.

Istnieją również bardziej rozbudowane technologie VR, które próbują stworzyć hiperrealistyczne środowisko, które pozwala na fizyczne poruszanie się i manipulację nim - korzystają z tego np. wirtualne parki rozrywki.

Rysunek 1 - Park rozrywki z VR

W takim parku ujrzymy wirtualny świat, ale ponieważ nie jesteśmy uwiązani do komputera możemy fizycznie chodzić po sferze wokół nas, która często jest po prostu pustą przestrzenią z atrapami pewnych obiektów, w które wdmuchiwane jest życie dopiero w VR. Użytkownik sam jest niejako kontrolerem i może zanurzyć się całkowicie w wyrenderowanym świecie, który jest mu przedstawiony. Często w takim miejscu można natknąć się również na takie efekty jak np. na sztuczny deszcz, ciepło/zimno, aby gra była jeszcze bardziej interaktywna i immersyjna. Jak nietrudno się domyśleć rozwój technologii VR przyczynił się do wzrostu popularności tego typu gier itp. Najprościej myśleć o VR jako o całkowicie odrębnym i sztucznym świecie zaprojektowanym by zmieniać rzeczywistość i pochłaniać w niej użytkownika. Nic nie jest prawdziwe, wszystko jest wirtualne.

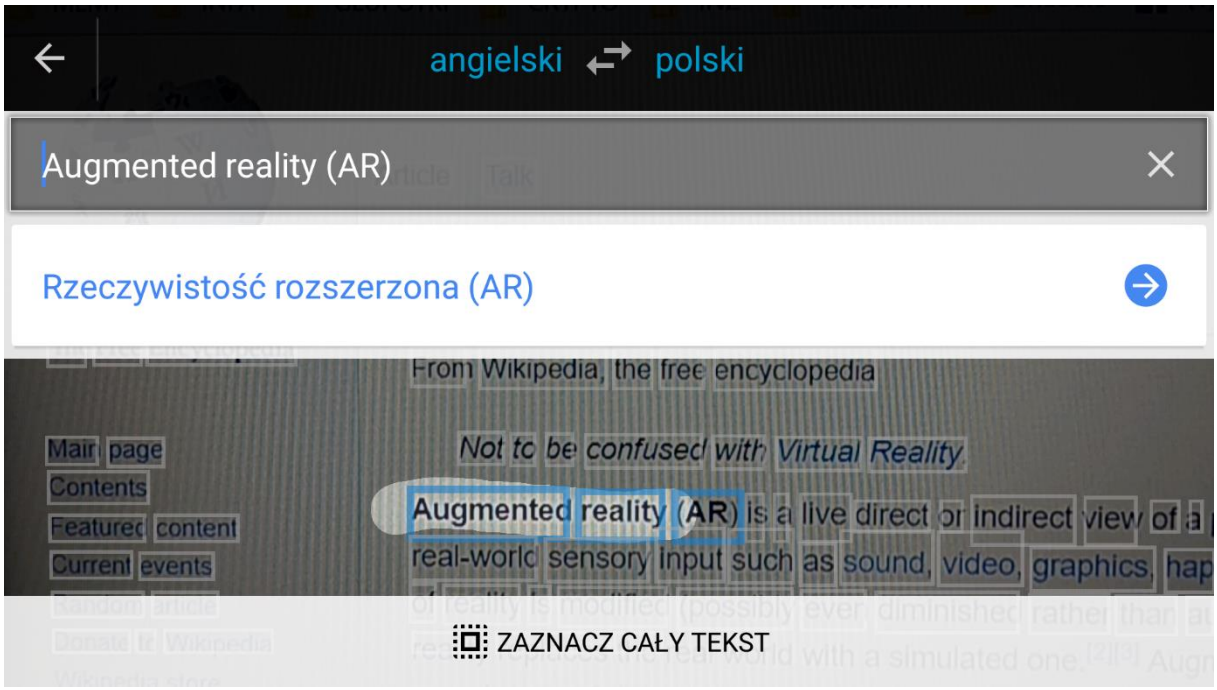
2.2. Rzeczywistość rozszerzona

Rzeczywistość rozszerzona[2] (ang. AR – Augmented Reality) czerpie z faktycznej, istniejącej rzeczywistości i zmienia pewne jej aspekty przez obiektyw smartfona, parę okularów czy wspomniane gogle. Z rzeczywistością rozszerzoną użytkownik zawsze widzi co jest przed nim, ale z dodaną na to wirtualną otoczką. Merriam-Webster definiuje rzeczywistość rozszerzoną jako uatrakcyjnioną wersję rzeczywistości stworzoną przy użyciu technologii by nakładać cyfrowe informacje na obraz czegoś obserwowanego przez urządzenie.

Rzeczywistość rozszerzona może przyjąć wiele form, ale jej najbardziej powszechne użycie jest w smartfonach. Najbardziej znaną obecnie aplikacją jest chyba gra Pokemon GO!, w której łapiemy wirtualne pokemony wesoło biegające sobie po naszym sąsiedztwie.

Rysunek 2 - Aplikacja na smartphone'y o nazwie Pokemon Go

Istnieją również aplikacje, które na bieżąco, po skierowaniu kamery w odpowiednie miejsce pokazują informacje o pobliskiej restauracji itp. Wszystkie te aplikacje używają aparatu smartfona, aby pokazywać wygląd na żywo tego co jest wokół nas, ale nakładają informacje na wyświetlacz. AR znajdziemy w wielu innych aplikacjach na smartfony, np. tłumacz gogle umożliwia bardzo praktyczną funkcjonalność. Pozwala po skierowaniu kamery na tekst w jednym języku przetłumaczyć i zamienić go na inny język.



Rysunek 3 - Screen z aplikacji Google Translate, przedstawiający możliwości rozszerzonej rzeczywistości

Jest ona dostępna również na Google Glass. Urządzenie było jednym z pierwszych, które korzystało z AR poza smartfonami, ale stało się źródłem kontrowersji z powodu obaw dotyczących prywatności, wysokiej ceny, małej ilości istotnych funkcjonalności. Jednakże porażka Google Glass nie przeszkadza innym firmom w próbach popularyzacji tej bardziej pasywnej metody interakcji (względem chociażby VR) i tchnięcia nowego życia w AR – są to np. LAFORGE Optical albo Optinvent. Istnieje wiele zwolenników, ale nie brakuje też przeciwników, którzy uważają, że urządzenia takie jak Google Glass nigdy nie osiągną sukcesu. Wszystko to działo dotyczyło wersji konsumenckiej, natomiast od roku 2015 na rynku pojawiła się nowa odsłona w wersji Enterprise Edition (Rys.x), którą zainteresowało się już ponad 50 firm. Google widzi w tym potencjał, a firmy zaczynają realizować zyski ze wzrostu efektywności pracowników czy ze wzrostu produkcji[4].

Rysunek 4 - Okulary Google Glass

Ogólnie rzecz ujmując, o rozszerzonej rzeczywistości można myśleć jak o nowej warstwie nałożonej na istniejącą rzeczywistość, ale nie przemieszanej z nią. Okulary być może podpadają pod tą kategorię, ale to smartfony znajdują w AR największe zastosowanie – jako nakładka na pole widzenia smartfona, bez interakcji jako część większego środowiska.

2.3. Rzeczywistość mieszana

Rzeczywistość mieszana (ang. MR – Mixed Reality) zabiera AR na nowy poziom i jest w zasadzie tym, czym wszyscy początkowo chcieli lub oczekiwali od AR. Zamiast będąc tylko pewną warstwą nałożoną na świat, rzeczywistość mieszana odnosi się do niejakiego połączenia cyfrowo renderowanych obiektów w świat rzeczywisty.

Wikipedia definiuje MR jako scalenie świata rzeczywistego z wirtualnym, aby utworzyć nowe środowiska i wizualizacje, gdzie fizyczne i cyfrowe obiekty koegzystują i wchodzą w interakcję w czasie rzeczywistym[3]. Rzeczywistość mieszana nie rozgrywa się tylko w świecie rzeczywistym lub tylko wirtualnym, ale jest swoistą mieszanką rzeczywistości oraz wirtualnej rzeczywistości obejmującą zarówno rzeczywistość rozszerzoną jak i wirtualność rozszerzoną. O ile omawiana definicja ma prawo bytu na smartfonach, to okulary do VR grają tutaj pierwsze skrzypce, ponieważ są zwyczajnie bardziej immersyjne. Jednymi z bardziej znanych gogli MR są, Microsoft HoloLens, Meta 2 oraz Magic

Leap One (choć nie są jeszcze szeroko dostępne na rynku, lub są jeszcze na etapie prac).

Rysunek 5 - Microsoft HoloLens

Rysunek 6 - Leap One

Rysunek 7 - Meta 2

Potrafią skanować pomieszczenie i przede wszystkim **rozumieć** przestrzeń, w której się znajdują dzięki czemu może precyzyjnie włączyć cyfrowe obiekty do istniejącego, rzeczywistego środowiska. Z wyświetlonymi obrazami przez okulary można wchodzić w interakcję, zupełnie tak, jakby były prawdziwe – ale w rzeczywistości nie są i o to właśnie w rzeczywistości mieszanej chodzi. Cyfrowy świat zostaje połączony z tym rzeczywistym, a przywdziewając gogle można z tym wszystkim wchodzić w interakcje, oddziaływać.

Technologie zmieniające rzeczywistość mogą być czasem dezorientujące, ponieważ bez wątpienia wszystkie się ze sobą zazębiają. Z pewnością rzeczywistość mieszana i rzeczywistość rozszerzona dzielą ze sobą bardzo wiele i mają podobne przypadki użycia. W pewnym sensie o MR można myśleć jako o podzbiorze bardzo zaawansowanej rzeczywistości rozszerzonej, gdyż MR udoskonala rzeczywistość, ale w bardziej gruntowny i integracyjny sposób. AR dodaje cyfrową warstwę na wierzch rzeczywistości, którą widzimy. MR łączy wszystko bardziej gładko i zapewnia więcej interakcji dla użytkownika.

3. Jak działają okulary VR? (M. Szumlański)

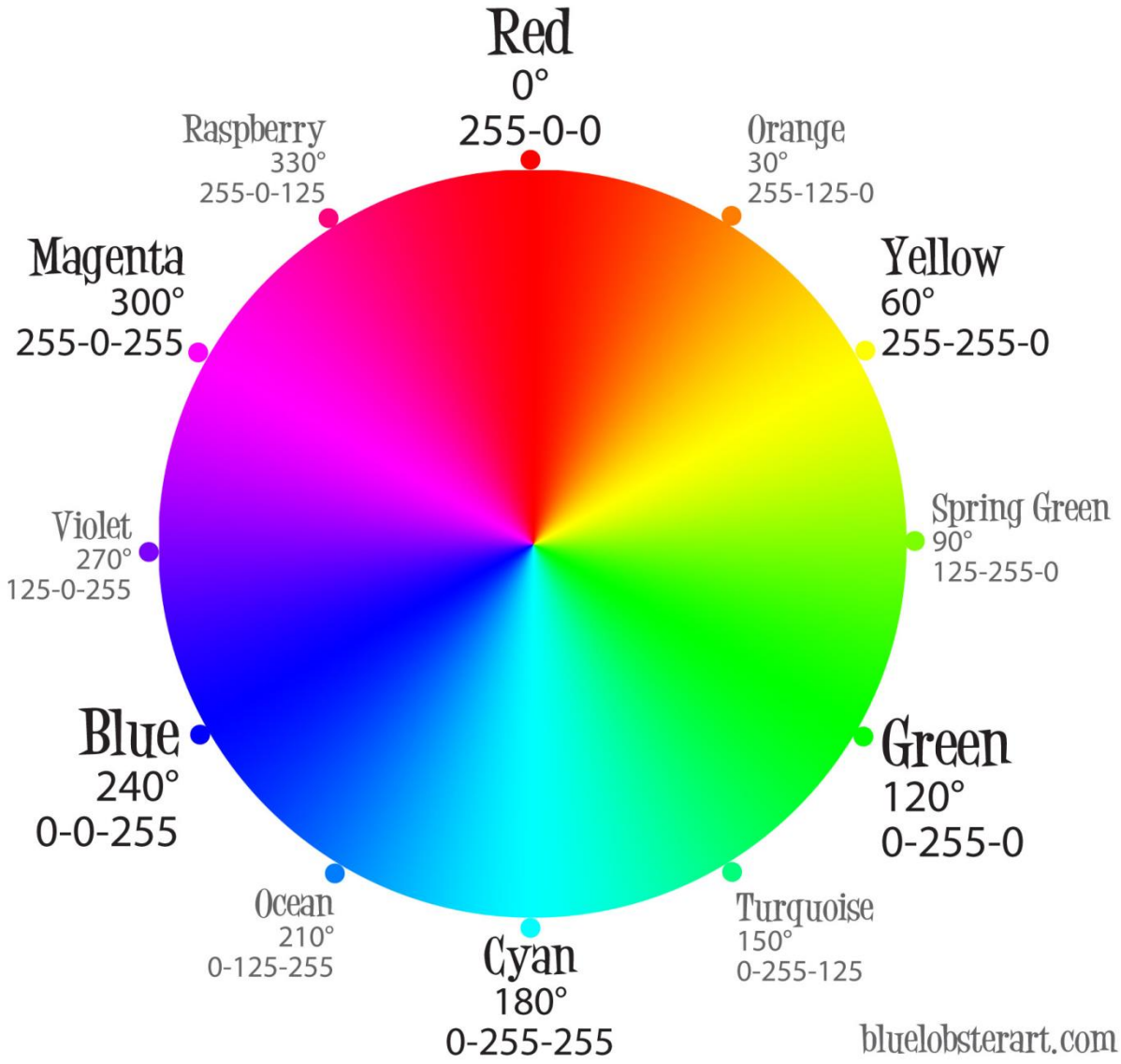
3.1. Stereoskopia

Aby rozpocząć omawianie mechanizmu działania nowoczesnych okularów, dobrze zacząć od stereoskopii. Stereoskopia[28] to mechanizm tworzenia złudzenia trójwymiaru na zdjęciach za pomocą dwóch zdjęć tego samego obiektu różniące się nieznacznie przesunięciem dając wrażenie widzenia stereoskopowego. Co to widzenie stereoskopowe?
Wikipedia mówi[22]:
„**Widzenie stereoskopowe** (widzenie binokularne) – dwuoczna percepcja głębi i odległości, rodzaj postrzegania wzrokowego umożliwiający ocenianie odległości do widzianych przedmiotów. „ W rzeczywistości nasze oczy nie widzą tego samego obrazu tylko dwa nieznacznie przesunięte i mózg jest w stanie stworzyć z nich jeden z efektem głębi – tzw. obraz cyklopowy.

O co chodzi: klasycznie obraz stereoskopowy, czyli stereogram[27] to para 2 zdjęć 2D (po 1 na każde oko) przy czym takie stereoskopowe zdjęcie nosi nazwę stereogramu i do patrzenia na nie wykorzystuje się okulary zwane stereoskopem. Zasadniczo obraz stereoskopowy to po para 2 zdjęć dwówymiarowych tej samej sceny (po jednym na każde oko) przy czym perspektywa jednego jest nieznacznie przesunięta względem drugiego celem uzyskania złudzenia ww. widzenia stereoskopowego.

3.2. Anaglify

Istnieją specjalne stereogramy tzw. anaglify[26] szczególnie popularne w przeszłości, których stworzenia polega na nałożeniu na siebie tego samego obrazu w różnych barwach z poziomym przesunięciem. Zazwyczaj jako 2 kolory wybiera się te, które leżą po przeciwnych stronach względem siebie na kole barw. Najbardziej typową para są szkła cyjan-czerwień, cyjanowe szkła nie przepuszczają prawie w ogóle czerwonego światła, a czerwone cyjanowego.



Rysunek 8 - Koło kolorów RGB

Każdy z dwóch obrazów dociera do odpowiedniego oka, ukazując stereograficzny obraz. Kora wzrokowa łączy dwa obrazy w postrzeganie obrazu jako trójwymiarową scenę. Zaletą tego rozwiązania jest cena, ale ma sporo wad: przede wszystkim obraz jest wyprany z oryginalnych kolorów.

Rysunek 9 - Obraz 3D

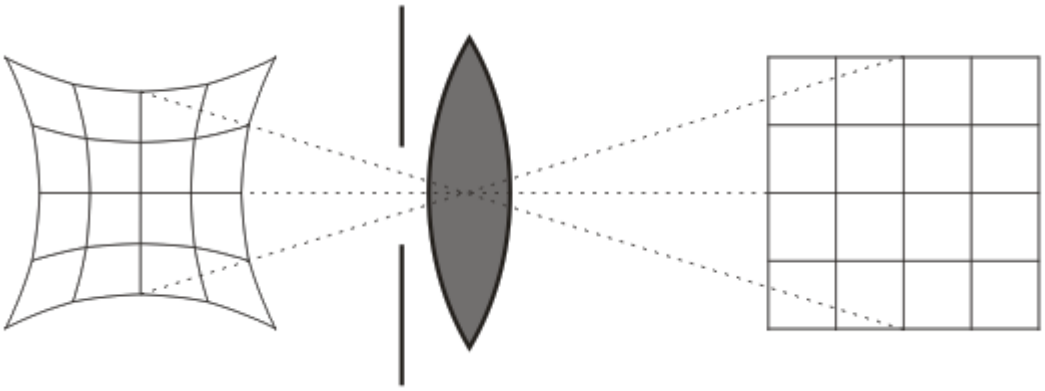
3.3. Optyka okularów VR

Okulary VR korzystają właśnie z podobnego mechanizmu. Składają się z 2 soczewek które odpowiednio zakrzywiają dwa lekko przesunięte względem siebie obrazy tej samej sceny. W taki sposób mózg można oszukać do widzenia obrazu cyklopowego, stworzenia z nich jednego obrazu z efektem głębi. Jednak jeśli obraz jest zbyt blisko oczu, wtedy soczewka oka nie może wystarczająco zagiąć światła i obraz na siatkówce powstaje niewyraźny – punkt ogniskowania przesuwa się za siatkówkę. Wyświetlacz/smartfon w goglach jest na odległości kilku cm od oczu, dlatego soczewki w goglach dodatkowo zakrzywiają światło które wpada do oka tak, aby obraz na siatkówce pozostał wyraźny i postrzegać go dalej niż jest w rzeczywistości - mapując zbliżony obraz na szersze pole widzenia.

3.4. Mankamenty i ich rozwiązania

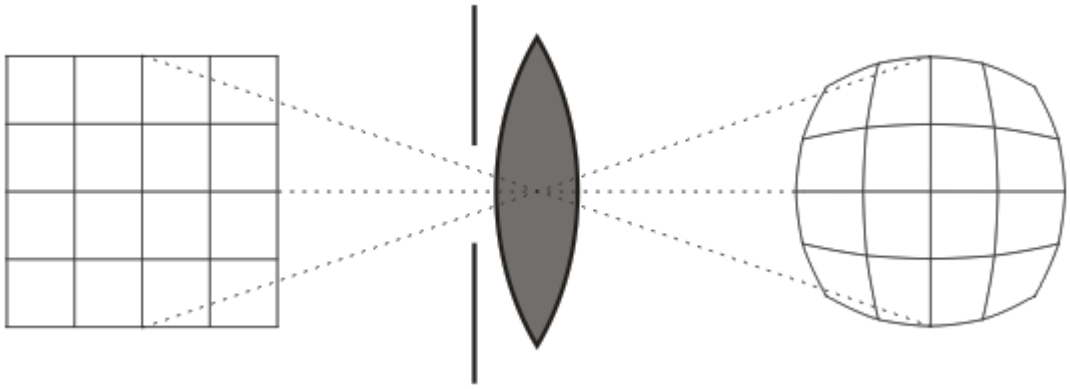
Nie jest to jednak rozwiązanie bez wad. Im większe pole widzenia tym powstaje większe zniekształcenia obrazu tzw pincushion effect[23].

Po prawej obraz z wyświetlacza, po lewej obraz widziany przez nas po przepuszczeniu go przez soczewkę.

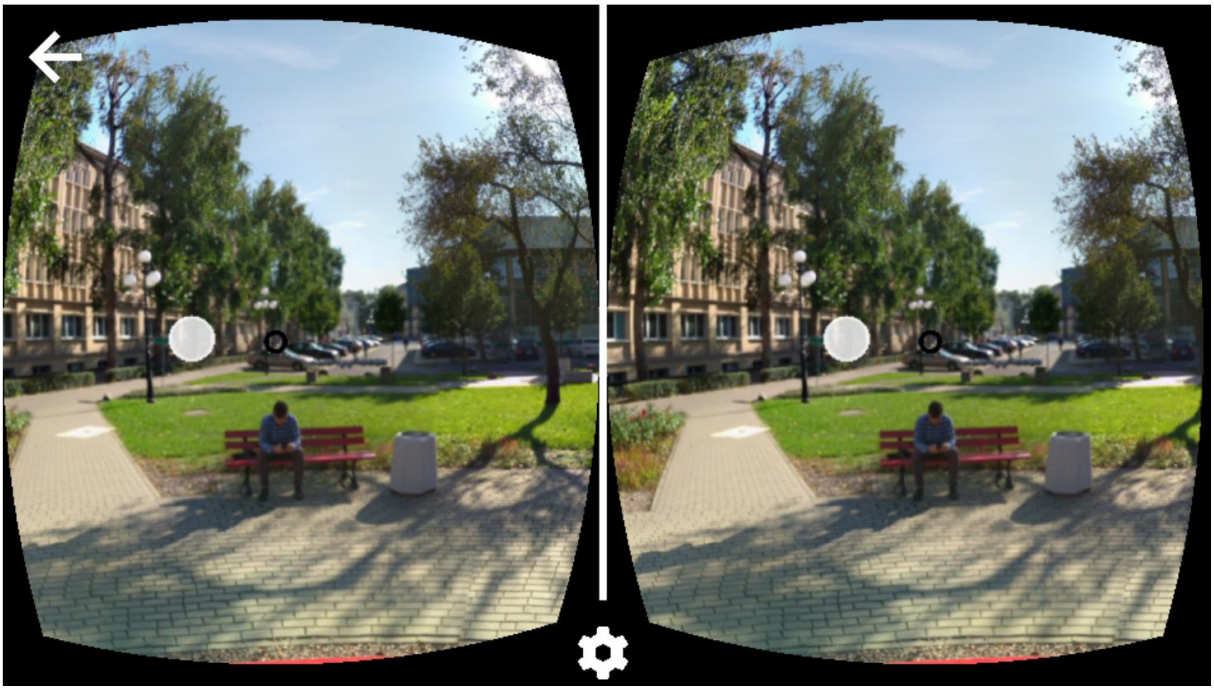


Rysunek 10 - Skupienie soczewek 1

Rozwiązuje się to programowo i na obraz na wyświetlaczu nakłada się tzw. barrel distortion, dzieki czemu finalny obraz wygląda neutralnie.



*Rysunek 11 - Skupienie soczewek 2 (*http://smus.com/vr-lens-distortion/)



Rysunek 12 - Barrel distortion na przykładzie zaimplementowanej aplikacji webowej

Wskutek używania zniekształcających soczewek występuje również tzw. aberracja chromatyczna. Wikipedia[24] definiuje ją jako:
„Cecha soczewki lub układu optycznego wynikająca z różnych odległości ogniskowania (ze względu na różną wartość współczynnika załamania) dla poszczególnych barw widmowych światła (różnych długości fali światła). W rezultacie występuje rozszczepienie światła, które widoczne jest na granicach kontrastowych obszarów pod postacią kolorowej obwódki”
Problem ten poza zmniejszaniem immersyjności doświadczenia, jest również główną przyczyną mdłości i bólów głowy podczas użytkowania aplikacji VR przez niektóre osoby. Chromatyczną aberracje również usuwa się sprzętowo (a raczej ogranicza).
Zdjęcie sprzed (góra) i po (dół) korekcji chromatycznej aberracji:

Rysunek 13 - Naprawa chromatycznej aberracji (https://cdn.photographylife.com/wp-content/uploads/2011/11/Uncorrected-and-Corrected-CA.jpg)

4. Wybór wykorzystywanego sprzętu (P. Brzoza)

4.1. Wyjaśnienie dotyczące wyboru

Podczas rozpoczęcia rozważań nad wyborem urządzeń do obsługi okularów oraz samych gogli, brano pod uwagę kilka czynników decydujących. W przypadku smartphone'ów, warto zaznaczyć, iż mówimy to o wyborze do samej obsługi technologii VR, a nie do robienia zdjęć (ten temat zarezerwowany został na osobny rozdział). Decydujące czynniki to: rozdzielczość i ilość kolorów wyświetlacza, przekątna ekranu, waga, system operacyjny, a dla okularów największe znaczenie miały: cena, jakość wykonania, optyka, wygoda, możliwości regulacji.

4.2. Zestawienie specyfikacji technicznej urządzeń

Analizując urządzenia odpowiedzialne za renderowanie obrazu, na wstępie zostały przekreślone wysokobudżetowe rozwiązania takie jak Oculus Rift czy HTC Vive, czyli technologie zintegrowane. Jakość doświadczeń z pewnością byłaby większa, natomiast nie każdy może sobie na takie rozwiązania pozwolić. Dlatego zdecydowano, że skupimy się na wyborze zalewających masowo rynek rozwiązaniach, okularach, do których możliwe jest umieszczenie smartphone'a i uruchamianie aplikacji właśnie z tego sprzętu będącego jednostką obliczeniową i wyświetlającą.

Mając do dyspozycji trzy modele telefonów sprawdzono ich charakterystykę sprzętową.

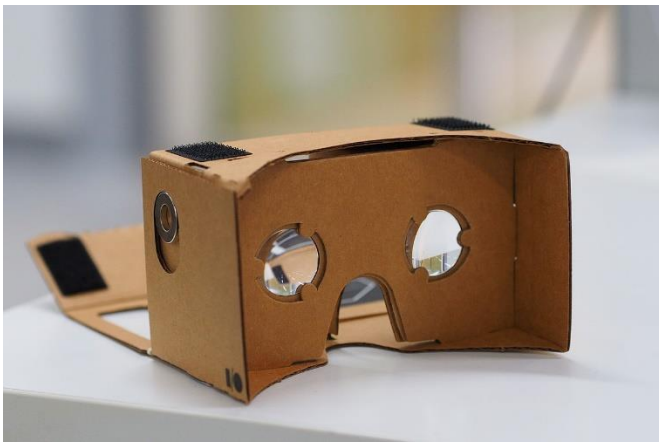
Tabela 1 - przedstawia porównanie istotnych parametrów sprzętowych

SMARTPHONE	XIAOMI REDMI NOTE 4X	SONY XPERIA Z1	HTC ONE M7
ROZDZIELCZOŚĆ WYŚWIETLACZA	1080 x 1920 [px]	1080 x 1920 [px]	1080 x 1920 [px]
RODZAJ WYŚWIETLACZA	Kolorowy / IPS TFT 16M kolorów	Kolorowy / IPS TFT 16M kolorów	Kolorowy / Super LCD 16M kolorów
PRZEKĄTNA EKRANU	5,50”	5,00”	4,70”
SYSTEM OPERACYJNY	Android 7.0	Android 5.1.1	Android 4.1.2
WAGA	165 [g]	170 [g]	143 [g]
PROCESOR GRAFICZNY	Adreno 506	Adreno 330	Adreno 320

Na podstawie danych przedstawionych w tabeli zaobserwowano, że najlepszym z dostępnych nam urządzeń okazał się Xiaomi Redmi Note 4X. Wygrał ze swoimi rywalami przede wszystkim przekątną ekranu oraz procesorem graficznym. Od Sony Xperia Z1 lepszy miał również system operacyjny i był lżejszy (co przy zestawie VR jest wyjątkowo ważne z uwagi na wygodę późniejszego użytkowania). Warto zauważyć, że HTC ONE M7 nie miałby nawet możliwości obsługi aplikacji Cardboard przez wersję androida niższą od 4.4, dlatego został od razu wyeliminowany.

4.3. Opis wybranych okularów VR

Na potrzeby późniejszych badań wybrano dwa modele gogli.
Pierwszy rodzaj to najprostsza możliwa forma prezentowanych okularów – Google Cardboard. Charakteryzują się prostotą swojej budowy, dlatego uzyskanie tego modelu jest niezwykle łatwe, ponieważ można je zbudować samemu. Na stronie producenta możemy znaleźć instrukcję złożenia, a wszystkie części, potrzebne do konstrukcji można kupić w Internecie lub w sklepie z narzędziami i są to kolejno: (1) tektura, (2) soczewki, (3) magnesy, (4) rzepy, (5) gumka recepturka, (6) chip NFC (ang. Near Field Communication).



Rysunek 14 - Google Cardboard (https://vr.google.com/intl/pl_pl/cardboard/get-cardboard/)

Drugie, to gogle chińskiej firmy FiiT VR 2S. Przy wyborze tychże okularów, kierowano się pozytywnymi opiniami oraz kilkoma recenzjami znalezionymi w Internecie. Przedstawiony zestaw składa się z gogli VR, wykręcanych(regulowanych) soczewek, pasków na głowę oraz szmatki do czyszczenia soczewek. Dodatkowo w soczewkach regulowany jest ich rozstaw, co pozwala na dokładne dopasowanie do wyświetlanego obrazu oraz wyregulowanie ostrości poprzez wkręcanie bądź wykręcanie soczewek. Plastikowa część jest bardzo solidna i lekka. Coś co z pewnością odróżnia FiiT VR od wersji kartonowej to wygoda i możliwość przymocowania headset'u do głowy za pomocą elastycznych pasków. Wyściółka, która znajduje się pomiędzy głową, a plastikiem jest bardzo miękka i przyjemna w dotyku, co ważne przepuszcza powietrze dlatego nie ma problemu z parowaniem soczewek. Montaż telefonu jest bardzo szybki za sprawą prostoty skonstruowanego zamocowania.

Rysunek 15 - Okulary FiiT VR 2S

5. Zdjęcia sferyczne – pozyskiwanie i właściwości (M. Szumlański)

5.1. Zdjęcie sferyczne, media 360

Zdjęcie sferyczne (inaczej fotosfera, zdjęcie 360) – zdjęcie uchwytujące kompletną scenę jako jeden obraz, która to scena oglądana jest obracając się wokół jednego centralnego punktu. Zazwyczaj tworzona na zasadzie łączenia ze sobą wielu zdjęć wykonanych w rotacji 360 stopni lub używając specjalnych kamer 360. Zdjęcie może być również generowane całkowicie komputerowo, a także łączyć ze sobą komputerowe efekty i fotografię. Media 360 (zdjęcia i filmy) można tworzyć w formacie mono lub stereo. Muszą być one przechowywane w odpowiednim formacie np. cubic lub equirectangular panorama. Format equirectangular jest obecnie najszerzej wspierany - w surowej formie zdjęcie jest oczywiście płaskie, a efekt wirtualnej rzeczywistości uzyskuje się odpowiednio je przekształcając. Przykładowe wykonane zdjęcie przez nas (format equirectangular) wygląda tak:

Rysunek 16 - Zdjęcie sferyczne w formacie equirectangular

Istnieją również inne formaty tzw. niepełne które nie pokrywają całej sfery, jednak w naszej pracy nie są nam potrzebne, wręcz niepożądane.

5.2. Wykorzystany sprzęt

Do zrobienia zdjęć wykorzystano aplikację Google Street View[5], statyw FIRST C-3570 oraz telefon Xiaomi Redmi Note X4.
Aplikacja Google Street View to aplikacja od firmy Google która m.in. pozwala na dość łatwe robienie zdjęć sferycznych oraz udostępnianie ich. Cały proces jest dość prosty, aplikacja przy użyciu żyroskopu w telefonie i po określeniu lokalizacji naznacza nam punkt na kamerze w rzeczywistości rozszerzonej. Po skupieniu się na nim w czasie 2-3 sekund w bezruchu automatycznie robi zdjęcie, które następnie nakładane jest na wygenerowaną grafikę 3D nałożoną na obraz z kamery. Następnie zdjęcie to staje się punktem odniesienia i wokół niego pojawiają się kolejne punkty, które po skupieniu się na nich robią kolejne zdjęcia. W efekcie, jeżeli poruszamy się po osiach mając kamerę w jednym miejscu możemy zapełnić naszą wirtualną grafikę 3D 360 stopni mozaiką zdjęć. Kiedy wykonamy wszystkie zdjęcia (w przypadku naszego aparatu kilkadziesiąt) poddawane one są scaleniu do zdjęcia sferycznego.
W naszym przypadku z racji niskiego budżetu, wykorzystano telefon z niższej półki Xiaomi Redmi Note X4 z aparatem o specyfikacji technicznej:

- matryca 13 Mpix
- ogniskowa 4mm oraz z przysłoną f/2.0.

Rysunek 17 - Statyw stabilizujący aparat

Rysunek 18 - Smartphone wykorzystywany to robienia zdjęć 360

Statyw bardzo ułatwił umiejscowienie punktu orientacji w jednym miejscu dzięki czemu zdjęcia wyszły stabilniej i aplikacja lepiej je scaliła. Porównanie gotowej zrobionej „z ręki” oraz przy użyciu statywu poniżej.
Gdyby wybrać potencjalne ulepszenie całego procesu bez wątpienia telefon można by wyposażyć np. w obiektyw szerokokątny (charakteryzujący się dużym kątem widzenia przy jak najmniejszym zniekształceniu geometrycznym w całym kadrze) który wpłynąłby na mniejszą liczbę zdjęć potrzebnych do wykonania całej sfery, a co za tym idzie przyspieszyłby proces jak również poprawiłby jakość – mniej zdjęć do scalania to mniej potencjalnych zniekształceń.

6. Praca z silnikiem graficznym Unity (P. Brzoza)

Stworzenie aplikacji wymagało przejścia przez określone zadania, rozpoczynając od najważniejszych to:

1) Stworzenie projektu
2) Dodanie wymaganych skryptów od GoogleVR SDK
3) Utworzenie wirtualnego statywu z kamerą
4) Utworzenie 7 sfer oraz teksturowanie ich
5) Wprowadzenie odpowiednich obiektów w postaci strzałek umożliwiające przejścia do innych sfer
6) Umieszczenie ikon z informacją o obiektach
7) Utworzenie skryptu odpowiadającego za mechanikę przejść

6.1. Niezbędne elementy instalacyjne

Unity3D

Zintegrowane środowisko programistyczne (ang. IDE, Integrated, Development Environment), jakim jest Unity[6], 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

Zainstalowano najnowszą dostępną bezpłatną wersję personalną tj. 2017.3.0f3 z zaznaczonym komponentem „Android Build Support".

Android Studio SDK

W celu prawidłowego działania środowiska Unity, należało zainstalować również środowisko Android Studio. Umożliwia ono między innymi przetestowanie aplikacji w wirtualnym urządzeniu, natomiast w naszym przypadku zależało nam przede wszystkim na instalacji Android SDK w najniższej możliwej wersji Android 4.4 (KitKat) z API Level 19. Ta wersja i wyższe współpracują z GoogleVR SDK, stąd ten wybór. W łatwy sposób umożliwia to wbudowany w środowisko SDK Manager.

Java Development Kit

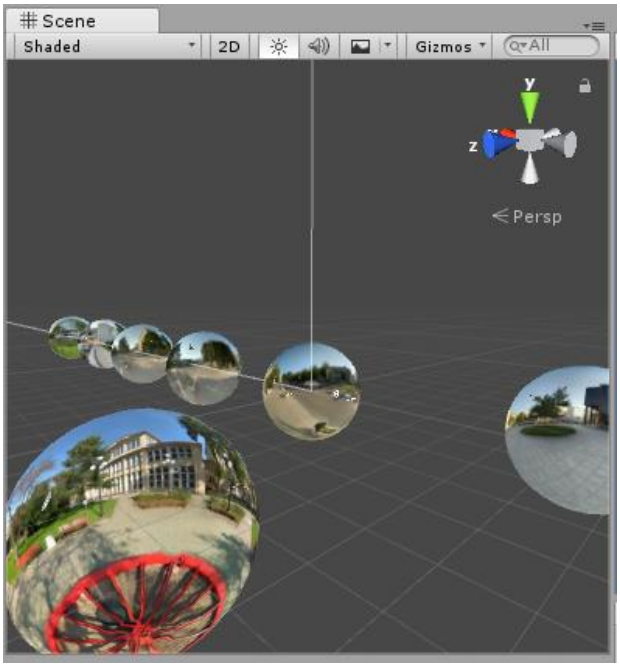
Zainstalowano również najnowsze biblioteki Java SE Development Kit w wersji 8u151.

Wszystko to dla możliwości skompilowania programu oraz jego późniejszego zbudowania/eksportu do pliku .apk w celu możliwości instalacji na smartphonach z systemem Android.

Zarówno Android SDK jak i JDK, należało podpiąć w ustawieniach Unity.

6.2. Poszczególne elementy środowiska

Scena (ang. scene)(Rys.x) – pojedyncza przestrzeń dla obiektów, w której mogą na siebie oddziaływać. Jedna aplikacja może zawierać wiele scen, a w momencie przełączania się między nimi wymaga ponownego wyrenderowania.



Rysunek 19 - Scena w środowisku Unity 3D

Obiekt gry (ang. game object) - każdy element, który występuje w scenie.

Komponenty (ang. components) – stanowią funkcjonalną część aplikacji tworzonych w Unity3D, zawiera je każdy obiekt gry, a te definiują np.

- pozycję i przekształcenie (ang. Transform)
- szczegóły wyświetlania takie jak tekstura (ang. Renderer)
- interakcję z innymi obiektami (ang. Collider)
- skrypty opisujące jego zachowania (ang. Script).

Ciekawą i wygodną rzeczą jest fakt, iż każdy obiekt jest równocześnie kontenerem i nic nie stoi na przeszkodzie aby zagnieżdżał kolejne.

Bardzo ważnym typem obiektu jest kamera (ang. camera). Określa jaka część sceny 3D ma być w obecnej chwili wyświetlana w okienku gry. Pozycja kamery może się zmieniać w trakcie działania aplikacji co może odpowiadać np. poruszaniu się gracza w przestrzeni sceny lub obracaniu przez niego głową.

Częścią aplikacji odpowiadającą za dynamikę oraz zachowania obiektów gry są skrypty. Unity3D pozwala uruchamiać skrypty napisane w językach C#, UnityScript (które jest składniowo podobne do JavaScriptu) oraz Boo. Zastosowanie języka C# jest bardzo praktyczne z uwagi na dwie istotne kwestie: powszechność i kompatybilność.

Za łatwiejsze wyszukiwania obiektów w programie odpowiedzialne są krótkie ciągi znaków zwane tagami.

6.3. Składniki projektu

Powstanie projektu rozpoczęło się od stworzenia odpowiedniej struktury, czyli podziału na foldery zawierające odpowiednie moduły, z których powstał projekt. Przedstawiono kawałki screenów ze środowiska Unity3D:

Rysunek 20 - Struktura folderów

Wszystkie składniki projektu znajdują się w folderze Assets (co w dosłownym tłumaczeniu w języku polskim może oznaczać wartości użyteczne, dla wygody stosowania w dalszej części pracy używana będzie angielska wersja). W jego skład wchodzą poszczególne pod foldery zawierające kolejno:

GoogleVR – zaimportowane elementy z biblioteki: GoogleVRForUnity_1.110.0.unitypackage[11], 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 specjalne Shadery.

Scripts – skrypty, odpowiadające głównie za mechanikę programu

Shaders – folder zawierający tzw. Shadery[7], czyli kod napisany w języku HLSL (High-Level Shading Language)[8] lub CG (C for Graphics), które są do siebie bardzo zbliżone. Służą między innymi do opisu sposobu teksturowania modeli.

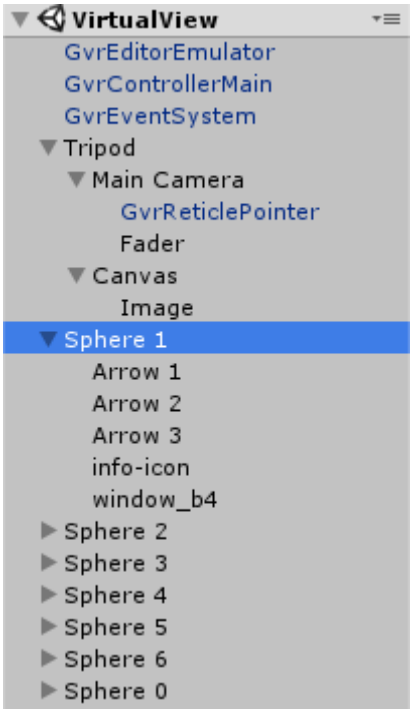
Sprites – elementy, rozszerzające rzeczywistość takie jak: strzałki wskazujące kierunek, w którą stronę się udamy, ikony informacyjne, okienka informacji o budynkach oraz wskaźniki w postaci okręgów ułatwiające określenie czasu w

momencie wykonywania tzw. GazeClick (dokładniej opisany w osobnym podrozdziale)

Tekturze – tekstury, czyli w naszym wypadku zdjęcia sferyczne

VirtualView – scena zawierająca obiekty

6.4. Struktura sceny:

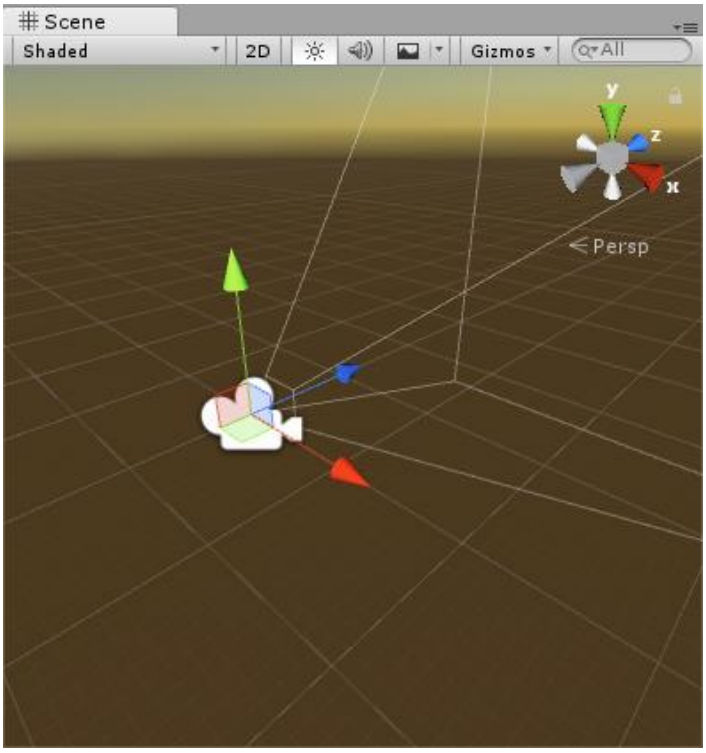


Rysunek 21 - Struktura sceny

Na strukturę sceny składają się kolejno:

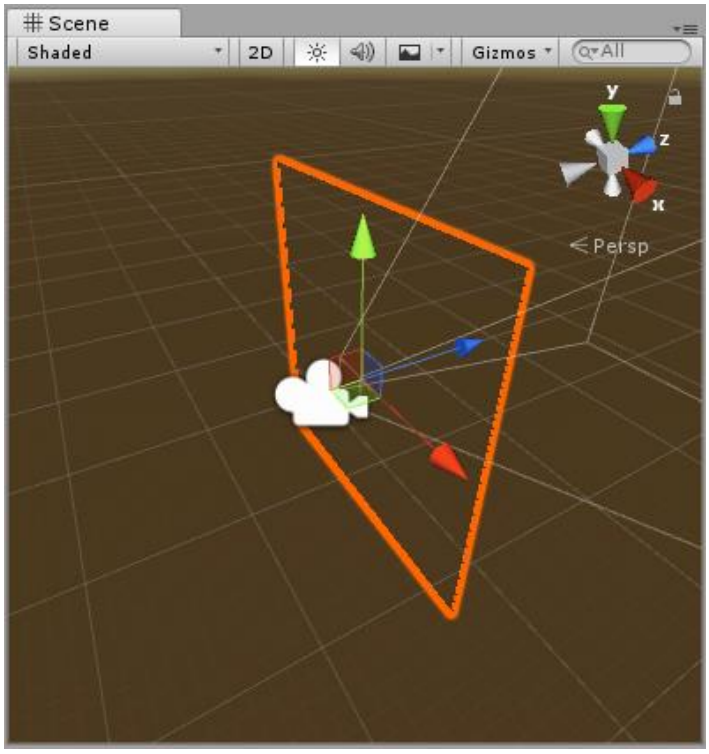
Cztery skrypty od GoogoleVR SDK (3 bezpośrednio w elementach sceny + 1 w kamerze głównej, dokładniej zostaną opisane w podrozdziale „Elementy biblioteki GoogleVR SDK”)

Statyw (ang. Tripod) – pusty obiekt, zawierający w sobie kamerę główną (ang. Main Camera).



Rysunek 22 - wirtualny statyw w Unity 3D

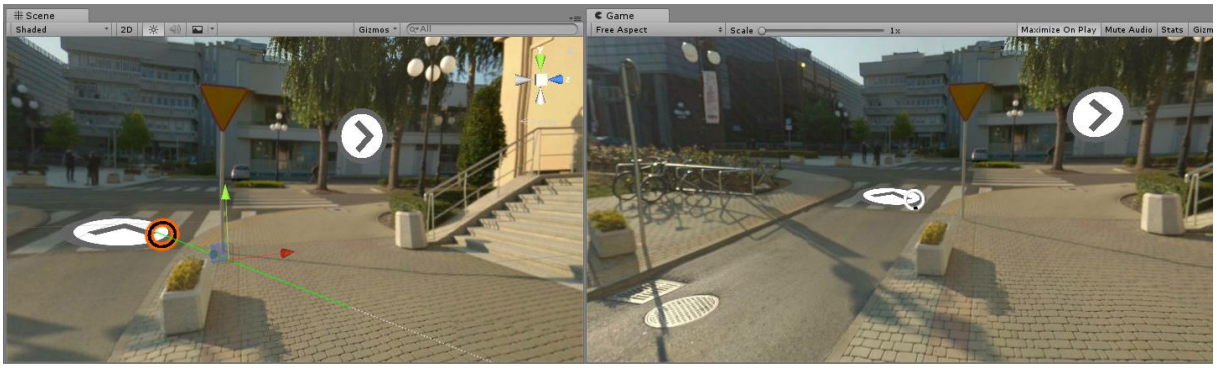
Ta z kolei zawiera w sobie skrypt służący do wskazywanie na inne obiekty oraz obiekt 3D typu czworokątnego (ang. Quad) o nazwie Fader, który posłuży do ciekawszych przejść widoków kamery.



Rysunek 23 - Fader

Statyw zawiera, również strefę Canvas[9], gdzie powinny znajdować się wszystkie elementy związane z interfejsem użytkownika (ang. UI). W naszym przypadku jest to komponent typu Image odpowiadający za GazeClick.

Siedem sfer 3D z czego każda, zawiera strzałki, oznaczające kierunki przemieszczania oraz ikony, dzięki którym, po najechaniu na nie, wyświetlana zostaje informacja o danym obiekcie.

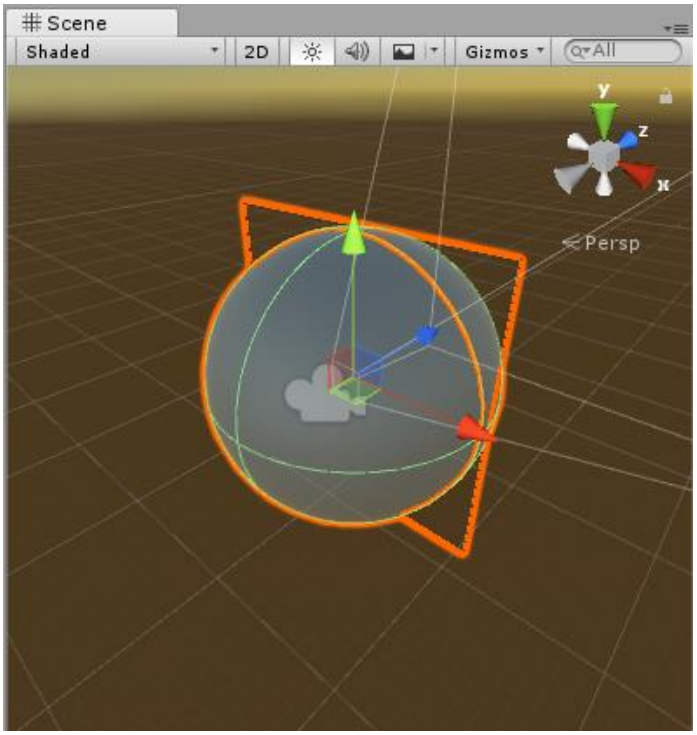


Rysunek 24 - przedstawienie występujących w projekcie strzałek do zmiany położenia; porównanie widoku między sceną a grą

Rysunek 25 - ukazanie funkcjonalności aplikacji

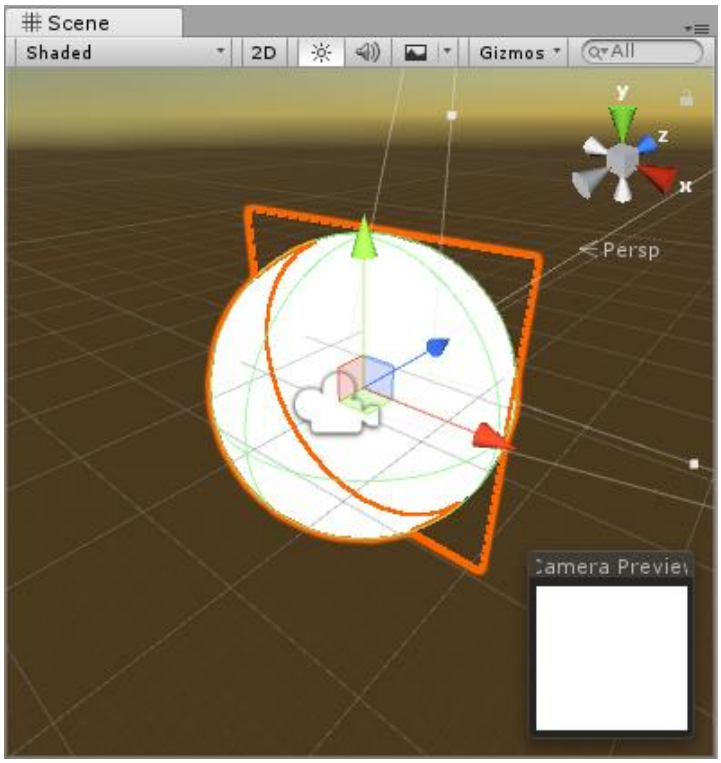
Wszystkie sfery tworzone są analogicznie do opisanej w następujący sposób:

Do naszego statywu z kamerą dodatkowo utworzono obiekt 3D typu sfera (ang. Sphere)



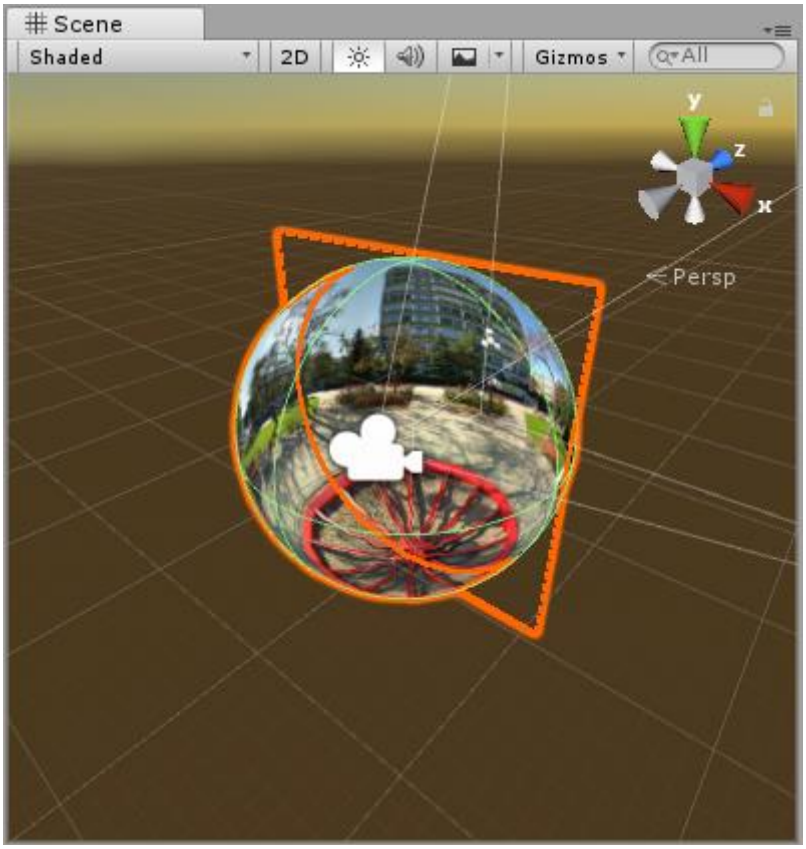
Rysunek 26 - pusta sfera 3D

Po utworzeniu sfery dodano Shader o nazwie Insideout, który umożliwił odpowiednie teksturowanie sfery, obiekt nie będzie wymagał dodatkowego oświetlenia, domyślne renderujący siatkę na biało.



Rysunek 27 - sfera wypełniona shaderem

Na koniec stworzono odpowiedni materiał za pomocą, wcześniej uzyskanych tekstur czyli zdjęć w formie 360 i dodano go do sfery co daje następujący efekt.



Rysunek 28 - Gotowa sfera, która przeszła przez wszystkie etapy budowania

6.5. Elementy biblioteki Google VR SDK

Przedstawiane skrypty to tzw. prefabrykaty (ang. prefabs) wprowadzające do projektu wsparcie w postaci kontrolerów [10]. Istnieją po to, aby łatwiej duplikować i zarządzać obiektami.

- GvrEditorEmulator – wszechobecny na scenie, pozwala na stymulowanie ruch kamery poprzez ruch głowy lub myszki
- GvrControllerMain – wszechobecny na scenie, główny kontroler, odpowiedzialny za zarządzanie stanami innymi kontrolerami
- GvrEventSystem – wszechobecny na scenie, zastępuje domyślny Event System Unity oraz osobny skrypt GvrPointerInputModule zastępuje domyślny StandaloneInputModule, umożliwia zarządzanie eventami
- GvrReticlePointer – zawierający się kamerze głównej, umożliwia wskazywanie na obiekty

6.6. Implementacja mechaniki przejść i funkcji GazeClick

Umożliwienie przechodzenia między sferami oraz uzyskanie dodatkowych efektów specjalnych umożliwia niżej opisany skrypt, który pozwala w projekcie na ciekawsze przejścia, czy też na wygodny „przycisk" ich zmiany. Na potrzeby implementacji stworzono skrypt obsługę tych możliwości. W dalszej części pracy zaprezentowane zostały kawałki kodu skryptu o nazwie „SphereChangerWithGazeClick.cs" wraz z ich opisem.

Zastosowana mechanika przejść, polega na płynnym zakryciu pola widzenia kamery przy wyjściu ze sfery (ang. fade out) oraz odkryciu przy wejściu w nową (ang. fade in).

Wszystko zaczyna się od zdefiniowania odpowiednich pól.

```
//This object should be called 'Fader' and placed over the camera
GameObject m_Fader;
float MyTime = 0f;
public Transform RadialProgress;
public Transform nextSphere{ get; set; }
```

- m_Fader - obiekt klasy GameObject, obsługuje wcześniej już stworzony obiekt gry o nazwie Fader, który zostanie przypisany w konstruktorze
- MyTime - pole typu zmiennoprzecinkowego float, odpowiada za odmierzanie czasu, po którym następuje przejście
- RadialProgress - obiekt klasy Transform, obsługuje komponent Image będący zmienny w czasie
- nextSphere - obiekt typu Transform, zawiera charakterystyczny dla języka C# getter i seter, odpowiada bezpośrednio za zmianę sfery

Następnie metoda typu void - Awake(), tworzona w momencie tworzenia obiektu na scenie, wykonuje się gdy obiekt zostaje aktywowany. Traktowana jest jako konstruktor.

```
void Awake()
{
    //Find the fader object
    m_Fader = GameObject.Find("Fader");

    //Check if we found something
    if (m_Fader == null)
        Debug.LogWarning("No Fader object found on camera.");
}
```

Zawiera w sobie przypisanie znalezionego obiektu Fader oraz instrukcję warunkowa, która dla przypadku nie znalezienia obiektu wypisuje w konsoli Unity ostrzeżenie.

```
public void ChangeSphere(Transform nextSphere)
{
    StartCoroutine(FadeCamera(nextSphere));
}
```

ChangeSphere - metoda typu void, pobierająca jeden argument o nazwie nextShpere klasy Transform. Jej wywołanie odbywa się w metodzie Update, która zostanie dokładniej wyjaśniona przy opisie funkcji GazeClick. Zostaje wywołana po przekroczenia czasu 3 sekund od nakierowania kursora na odpowiedni obiekt w tym przypadku są to obiekty obrazujące kierunki. ChangeSphere wywołuje metodę StartCoroutine[12] z klasy MonoBehaviour, która zaś przyjmuje wywołanie innej metody o nazwie FadeCamera powodująca oczekiwany efekt zakrycia i odsłonięcia kamery. Między tymi zdarzeniami występuje zmiana pozycji kamery na nową nextSphere wskazaną w środowisku Unity.

```
IEnumerator FadeCamera(Transform nextSphere)
{

    //Ensure we have a fader object
    if (m_Fader != null)
    {
        //Fade the Quad object in and wait 0.75 seconds
        StartCoroutine(FadeIn(0.75f, m_Fader.GetComponent<Renderer>().material));
        yield return new WaitForSeconds(0.75f);

        //Change the camera position
        Camera.main.transform.parent.position = nextSphere.position;

        //Fade the Quad object out
        StartCoroutine(FadeOut(0.75f, m_Fader.GetComponent<Renderer>().material));
        yield return new WaitForSeconds(0.75f);
    }
    else
    {
        //No fader, so just swap the camera position
        Camera.main.transform.parent.position = nextSphere.position;
    }
}
```

FadeCamera, metoda typu IEnumerator sprawdza, czy pobranie obiektu typu Fader powiodło się, jeśli tak to wywołuje kolejne współprogramy, które odpowiadają za zmianę, w przypadku nie powodzenia zwyczajnie zmienia sferą bez specjalnego przejścia.

```
IEnumerator FadeOut(float time, Material mat)
{
    //While we are still visible, remove some of the alpha colour
    while (mat.color.a > 0.0f)
    {
        mat.color = new Color(mat.color.r, mat.color.g, mat.color.b, mat.color.a
            - (Time.deltaTime / time));
        yield return null;
    }
}

IEnumerator FadeIn(float time, Material mat)
{
    //While we aren't fully visible, add some of the alpha colour
    while (mat.color.a < 1.0f)
    {
        mat.color = new Color(mat.color.r, mat.color.g, mat.color.b, mat.color.a
            + (Time.deltaTime / time));
        yield return null;
    }
}
```

Przedstawiony kod skryptu zawiera dwie metody typu IEnumerator o nazwach FadeOut oraz FadeIn pobierające dwa argumenty, pierwszy typu float i obiekt klasy Material.
Dane metody odpowiadają za usuwanie w przypadku FadeOut i dodawanie w przypadku FadeIn, koloru alfa w obiekcie Fader, co prowadzi do całkowitej przeźroczystości obiektu bądź całkowitego ukazania się go. Kolor jest stopniowo zmieniany w czasie.

GazeClick

Drugą wprowadzonym udogodnieniem jest tzw. GazeClick, czyli funkcjonalność która umożliwia „kliknięcie” obiektu na scenie po nakierowaniu na niego. Najechanie kursorem na dany obiekt włącza skrypt.

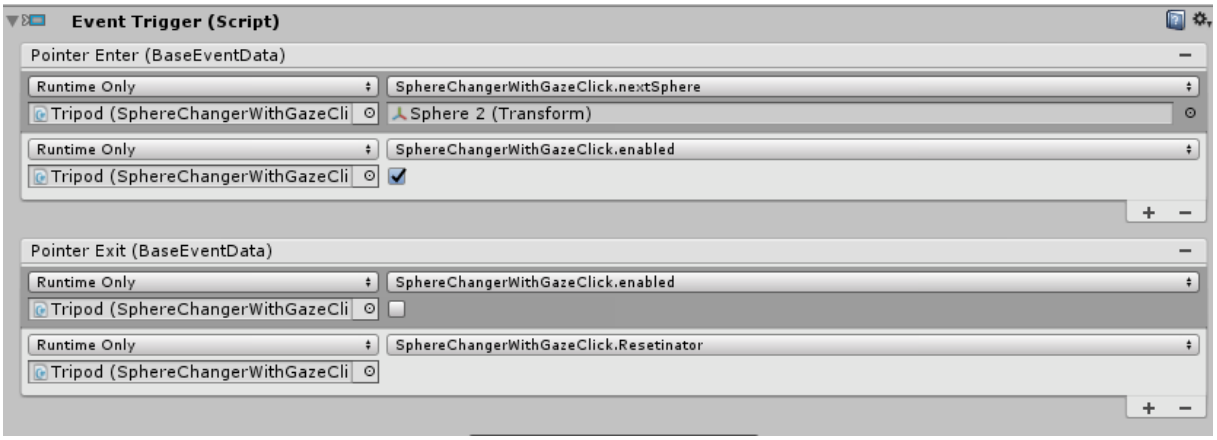
```
void Update()
{
    MyTime += Time.deltaTime;
    RadialProgress.GetComponent<Image>().fillAmount = MyTime/3;

    if (MyTime >= 3f) {
        ChangeSphere (nextSphere);
    }
}
```

Update () – metoda wywoływana w każdej klatce, jeśli MonoBehaviour jest włączone. Najczęściej używana metoda służąca do obsługi jakichkolwiek zachowań w grze: przesunięcia, obracanie, itd. Dla potrzeb uzyskania

określonego zachowania, z każdym wykonaniem funkcji Update inkrementowana jest zmienna MyTime o dany czas uzyskany z metody deltaTime klasy Time. Następnie dopełniany jest promień naszego okręgu i gdy przekroczy czas 3 sekund, bądź będzie mu równy, nastąpi wywołanie metody ChangeSphere.

Rysunek 29 - przedstawienie GazeClick

Na koniec ustawienie skryptu Event Trigger obsługującego zdarzenia.

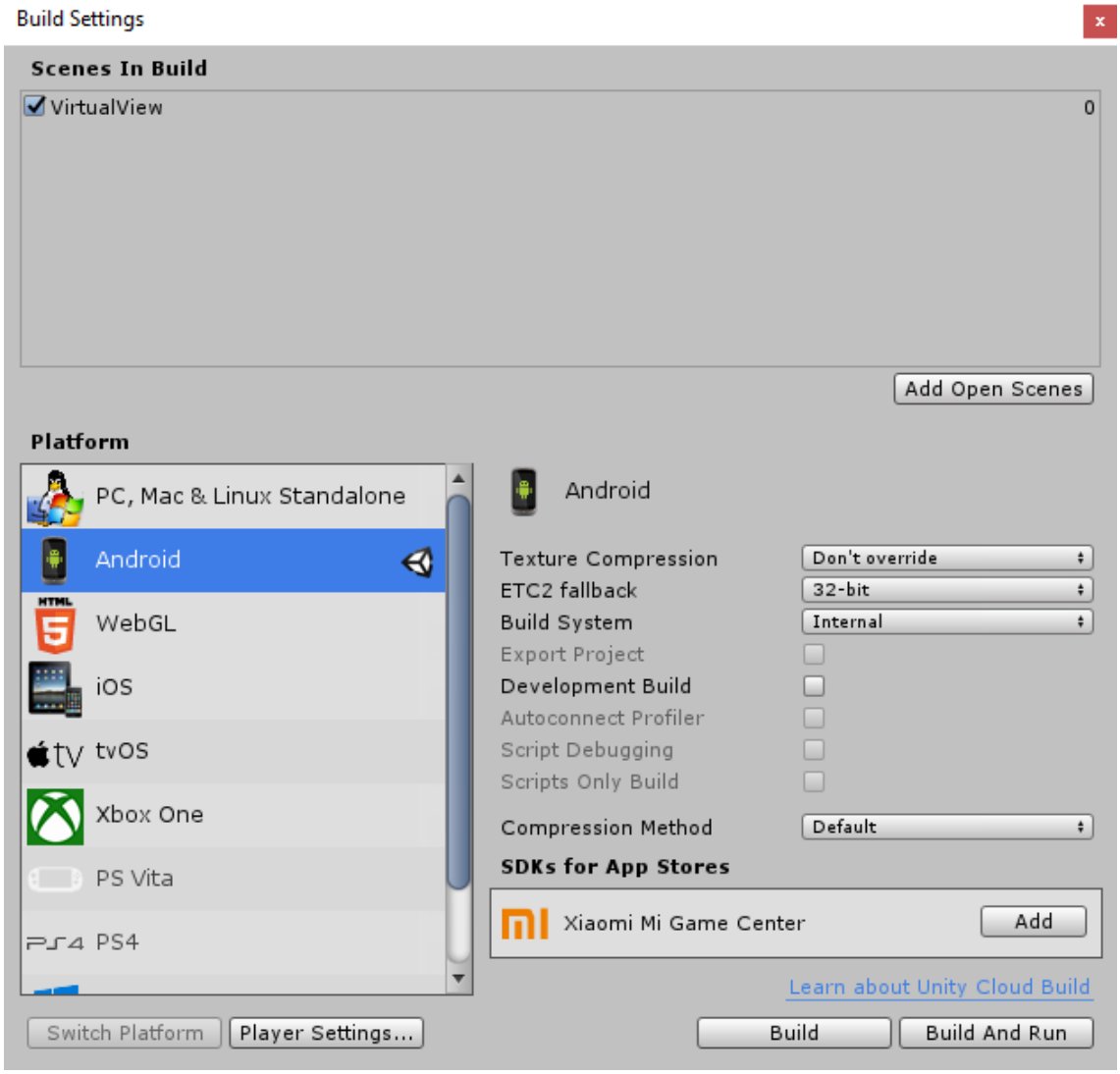


Rysunek 30 - okno przedstawiające ustawienia Event Trigerra

Przedstawiona ilustracja obrazuje w jaki sposób obsługiwane są przejścia i GazeClick. Zdarzenie Pointer Enter odpowiada na uruchomienie skryptu, natomiast Pointer Exit za jego wyłączenie. Dodatkowo w pierwszym, jako parametr przyjmowana jest sfera, na którą ma nastąpić zmiana, a w drugim przed wyłączeniem skryptu uruchamiana jest bardzo prosta metoda Resetinator, która zwyczajnie zeruje czas oraz postęp wypełnienia okręgu.

6.7. Export aplikacji na urządzenia z systemem Android

Za pomocą ukazanego na okienka można w łatwy sposób wyeksportować naszą aplikacje na różne platformy. Autorzy zdecydowali się na zaznaczoną opcję.



Rysunek 31 - okno eksportu, budowania .app

Dodatkowo wymagane było wskazanie Virtual Reality SDK, w naszym przypadku to Cardboard oraz Minimum API Level – zainstalowane wcześniej Android 4.4. Po zbudowaniu i zainstalowaniu aplikacji na telefonie uzyskaliśmy widoczny niżej efekt.

Rysunek 32 - widok z aplikacji zainstalowanej na smartphone'ie z systemem operacyjnym Android

Jak widać przedstawiony proces budowania aplikacji przebiegł pomyślnie, czego dowodem jest powyższa grafika.

7. Aplikacja wykonana przy użyciu VR View w JavaScript (M. Szumlański)

7.1. Opis

Aplikacja wykonana na wzór aplikacji Unity aby można było je jakoś sensownie porównać i dojść do jakichś wniosków w kwestii rozwoju aplikacji.
Google udostępnia JavaScript API, które pozwala na wstawienie zdjęć oraz filmów 360 na stronę internetową tworząc i kontrolując zawartość elementu iframe lub jawnie deklarując element iframe. Zaletą tego rozwiązania niewątpliwie jest jego prostota i elegancja.

7.2. Implementacja:

Aby w ogóle używać klasy VR View należy dołączono źródło skryptu w kodzie HTML oraz stworzono element div o id „vrview”, który po załadowaniu VR View zostanie zastąpiony elementem iframe o klasie „vrview”.

```
<!DOCTYPE html>
<html>
    <head>
        <title>VirtualView</title>
    </head>

    <body>
        <script src="http://storage.googleapis.com/vrview/2.0/build/vrview.min.js"></script>
        <script src="main.js"></script>

        <div id="vrview"></div>
    </body>
</html>
```

7.3. VR View

Kiedy strona się załaduje, chcieliśmy zarejestrować to, że się załadowała i wywołać funkcję, która stworzy nową instancję VR View.

Instancję VR View stworzono wołając konstruktor, VRView.Player i przekazując mu selektor, który determinuje gdzie „wrzucić” VR View oraz przekazując mu kolekcję parametrów. Selektor powinien wskazywać na id elementu diva, którego zawarliśmy w HTML-u. Parametrami moga być m.in. źródło np. filmu

lub zdjęcia (video/image), wymiary iframe (width/height), flagi np. is_debug do włączenia funkcji debugowania zawierającego m.in. licznik FPS-ów.

```
window.addEventListener('load', onVrViewLoad)

function onVrViewLoad() {
    vrView = new VRView.Player('#vrview', {
    width: '100%',
    height: 480,
    image: 'img/B5.jpg',
    is_stereo: false

  });
```

Ważne dla kolejnych części instrukcji - zdefiniowano obiekt scenes, którego pola do poszczególne sceny o kluczu takim samym jak nazwa zdjęcia, które z kolei mają zdefiniowane swoje parametry (m.in. ścieżka do pliku zdjęcia) i pola hotspots przypisane dla każdej sąsiedniej sceny, które zawierają informację o położeniu i rozmiarze hotspota na danej scenie (na zdjęciu poniżej zdefiniowana scena B5 i hotspot do sceny B4).

```
var scenes = {
    B5: {
        image: 'img/B5.jpg',
        is_stereo: false,
        hotspots: {
            B4: {
                pitch: 0,
                yaw: -120,
                radius: 0.05,
                distance: 1
            }
        }
    },
```

VR View rozgłasza różne rodzaje zdarzeń, które zostały być zarejestrowane używając funkcji VRView.on(String event, function handleEvent). Funkcja on() przyjmuje dwa argumenty: nazwa wydarzenia w postaci Stringa oraz funkcja do obsługiwania go.

```
vrView.on('ready', onVRViewReady);
vrView.on('modechange', onModeChange);
vrView.on('click', onHotspotClick);
vrView.on('error', onVRViewError);
vrView.on('getposition', onGetPosition);
```

Typy wydarzeń to:

- ready : VR View załadował początkową zawartość i jest gotowy przyjmować komendy.
- error : VR View zgłasza błąd, np. kiedy zawartość nie może być załadowana.
- click : VR View zarejestrował „klik” (tap na telefonie lub gaze w trybie VR). Może być użyte do przejścia do następnego slajdu w carousel lub rejestrowania, gdy użytkownik klika na hotspot.
- modechange : VR View zmienia tryb, np. kiedy użytkownik wchodzi w tryb VR lub pełny ekran. Można używać tego, aby śledzić zachowanie użytkownika rejestrując wywołania zwrotne kiedykolwiek użytkownik zmienia tryby.
- getposition: VR View zarejestrował zmianę pozycji (atrybuty Yaw oraz Pitch wskazującego na kolejno pozycję w osi poziomej i pionowej)

```
function onVRViewReady(e) {
  console.log('onVRViewReady');
  loadScene('B5');
}

function onModeChange(e) {
  console.log('onModeChange', e.mode);
}

function onVRViewError(e) {
  console.log('Error! %s', e.message);
}

function onGetPosition(e) {
    console.log(e)
}
```

7.4. Ładowanie nowej zawartości

Nową zawartość można załadowano do iframe bez jego przeładowywania, używając funkcji setContentInfo()/setContent (). Funkcja ta przyjmuje URL do zdjęcia 360 lub filmu jako string i kolekcję innych parametrów. Zasadniczo może używać tych samych parametrów co konstruktor VR View, poza width i height.

```
function loadScene(id) {
  console.log('loadScene', id);

  vrView.setContent({
    image: scenes[id].image,
    preview: scenes[id].preview
  });

}
```

7.5. Hotspoty

To miejsca na fotosferze, z którymi użytkownicy mogą wejść w interakcję. Hotspoty mogą być wyświetlane na wszystkich platformach, ale to w jaki sposób użytkownicy wchodzą w nimi z interakcję jest różny i zależny od platformy.
- Na desktopach, najechanie na hotspot zmienia je wizualnie, a kliknięcie aktywuje.
- W trybie mobile tap na hotspoty aktywuje go
- W trybie VR, wpatrywanie się w fotosferę zawierająca hotspoty powoduje pojawienie się celownika, który zmienia stan gdy wcelujemy go bezpośrednio w hotspot, wtedy tap gdziekolwiek aktywuje hotspot.

Hotspoty to okrągłe znaczniki na sferze z nadanym ID (String), koordynatami środka, promieniem oraz odległością od kamery. Zostały dodane poprzez funkcję vrView.addHotspot(). Koordynaty dzielą się na pitch i yaw i są one sferyczne. Domyślnie widok jest skierowany na (0,0). Zakres pitch to [-90,90] gdzie dodatnie wartości wskazują górę. Yaw to zakres [-180, 180] a dodatnie wartości wskazują na prawo.

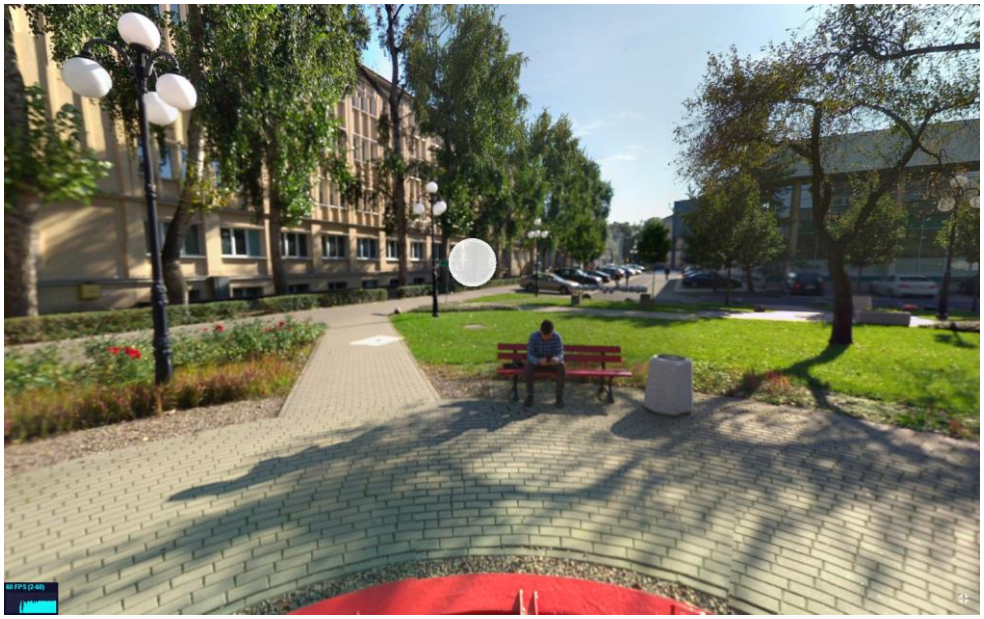
```
// Dodaje wszystkie hotspoty do sceny
var newScene = scenes[id];
var sceneHotspots = Object.keys(newScene.hotspots);
for (var i = 0; i < sceneHotspots.length; i++) {
  var hotspotKey = sceneHotspots[i];
  var hotspot = newScene.hotspots[hotspotKey];

  vrView.addHotspot(hotspotKey, {
    pitch: hotspot.pitch,
    yaw: hotspot.yaw,
    radius: hotspot.radius,
    distance: hotspot.distance
  });
}
```

Aby dodać wydarzenie, kiedy hotspoty zostanie aktywowany używam funkcji on().

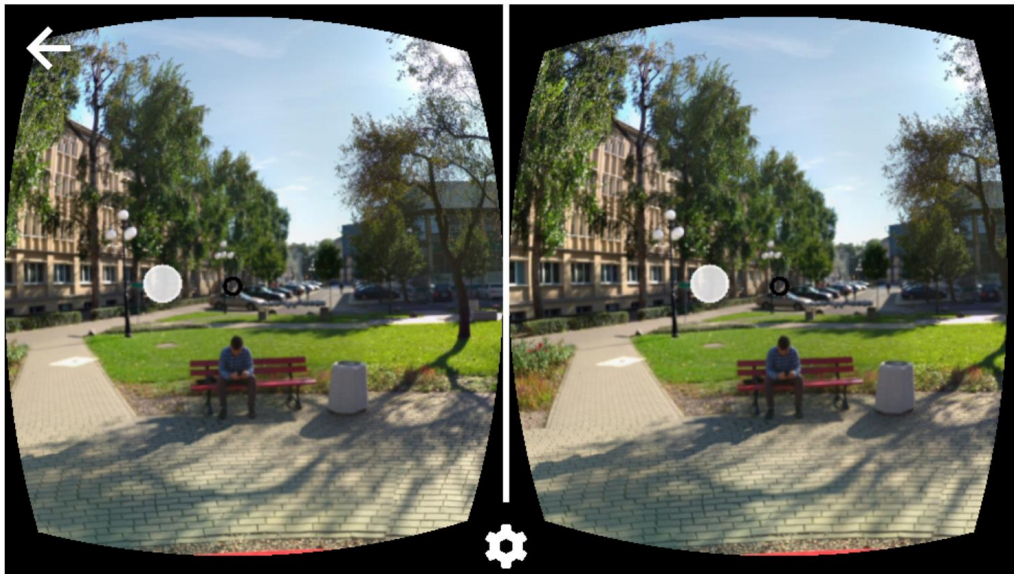
```
function onHotspotClick(e) {
 vrView.getPosition();
  console.log('onHotspotClick', e.id);
  if (e.id) {
    loadScene(e.id);
  }
}
```

7.6. Hosting

Aplikacja będąc aplikacją webową została udostępniona w sieci przy użyciu narzędzia „Chrome Web Server” które pozwala na hosting stron internetowych i dostęp do nich z dowolnego urządzenia posiadającego przeglądarkę Google Chrome w sieci lokalnej jak i publicznej. Dzięki temu można było połączyć się ze stroną web na którą wrzuciliśmy kod do aplikacji przez smartfon, który podłączyliśmy do okularów co pozwoliło na oglądanie zdjęć w trybie VR przez sieć.



Rysunek 33 - Tryb desktopowy (poruszanie za pomocą kursora) – aplikacja webowa



Rysunek 34 - Tryb VR – aplikacja webowa

8. Badania „user experience” (wspólnie)

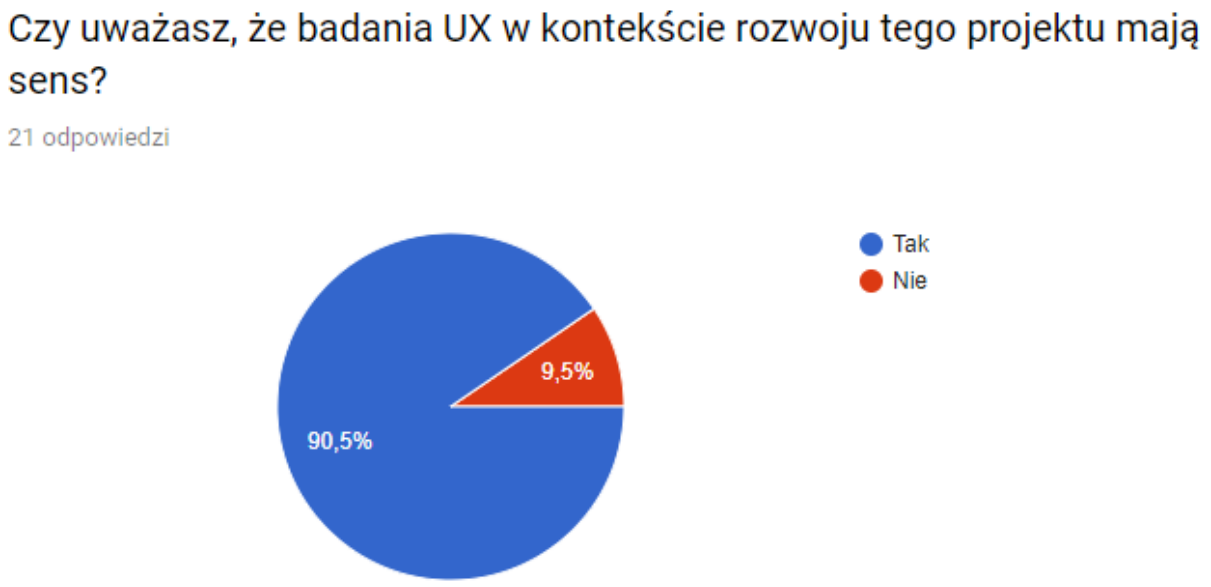
User experience odnosi się do całości wrażeń i nastawienia użytkownika co danego produktu, systemu lub usługi, wynikające z jego użytkowania. UX zawiera w sobie wszystkie preferencje, przekonania, wyobrażenia, emocje psychiczne oraz fizyczne reakcje, zachowania oraz dokonania, które następują przed, w trakcie i po używaniu produktu. Międzynarodowa organizacja normalizacyjna (ISO) wymienia 3 czynniki wpływające na doświadczenie użytkownika: system, użytkownika oraz kontekst użycia. Badanie nad UX polega na zaprojektowanie systemu, z którym interakcja będzie dostarczała jak najbardziej pozytywnych doświadczeń. Wikipedia mówi w tej kwestii: „Produkt powinien: prezentować się w sposób atrakcyjny dla użytkownika, być funkcjonalny, ergonomiczny, użyteczny, korzystanie z niego powinno sprawiać przyjemność i dawać satysfakcję.”

Wychodzimy z założenia, że nasz projekt jest zorientowany pod użytkownika i jego jak najbardziej pozytywne doświadczenia. W związku z tym zorganizowano badania mające na celu ulepszenie obecnego projektu i ew. ukierunkowanie jego dalszego rozwoju. Badania odbyły się w prywatnym mieszkaniu, przy wykorzystaniu obu aplikacji oraz 2 rodzajów gogli - Google Cardboard i FiiT VR 2S.

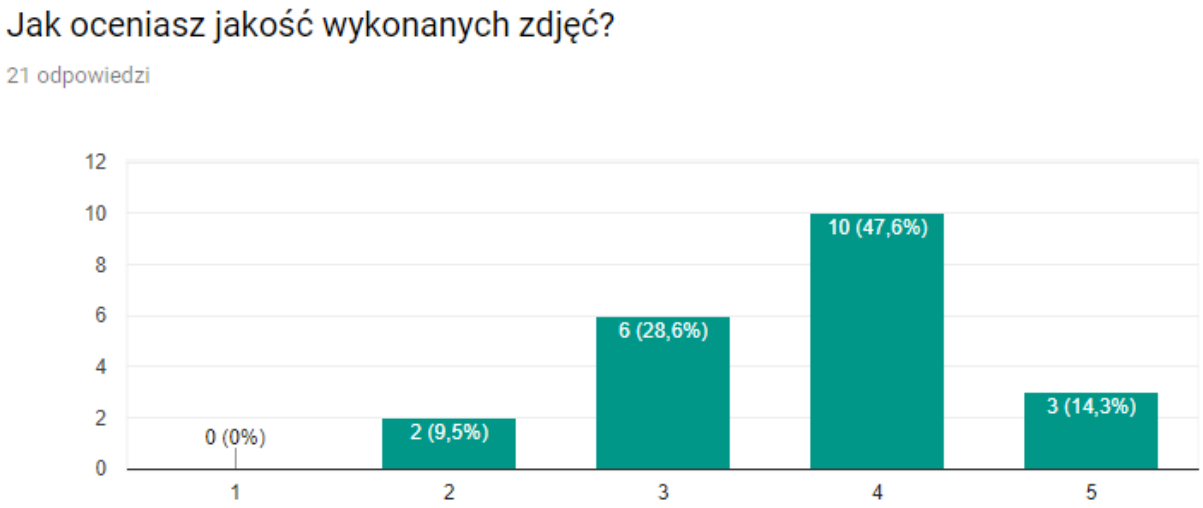
W badaniu wzięło łącznie udział 21 osób, w tym 76% mężczyzn (16) w przedziale wiekowym 19-24 lata, byli to głównie studenci lub absolwenci studiów technicznych, którzy są znajomymi autorów. Zaznaczam tutaj, że fakt ten mógł mieć pewien wpływ na subiektywność wydanych opinii. Mała próba ze specyficznego środowiska ankietowanych może też nieprecyzyjnie oddawać obraz odczuć przeciętnego użytkownika.

Badania podzielono na 2 części: w pierwszej została przedstawiona ankieta zawierająca kilkanaście pytań zamkniętych, zaś w drugiej poprosiliśmy o komentarz do 3 pytań, jakie chcieliśmy zadać potencjalnym przyszłym użytkownikom, które pomogłyby nam zaprojektować jeszcze lepsze doświadczenie w przyszłości. Przeprowadziliśmy również krótki wywiad związany z ww. pytaniami.

8.1. Część 1 - pytania zamknięte

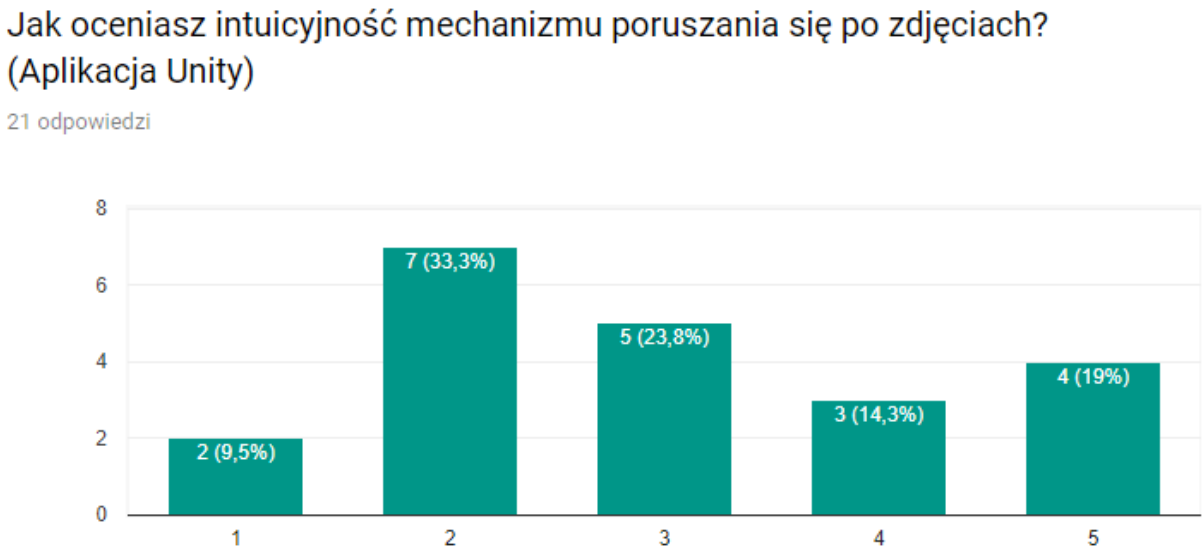


Pierwszym pytaniem chcieliśmy sprawdzić jaką wagę ludzie przywiązują do projektowania lepszego doświadczenia w kontekście takich aplikacji, odsetek odpowiedzi twierdzących mówi sam za siebie – ludzie są świadomi tego, że projektowanie UX to bardzo ważna część przy tworzeniu systemów interaktywnych.

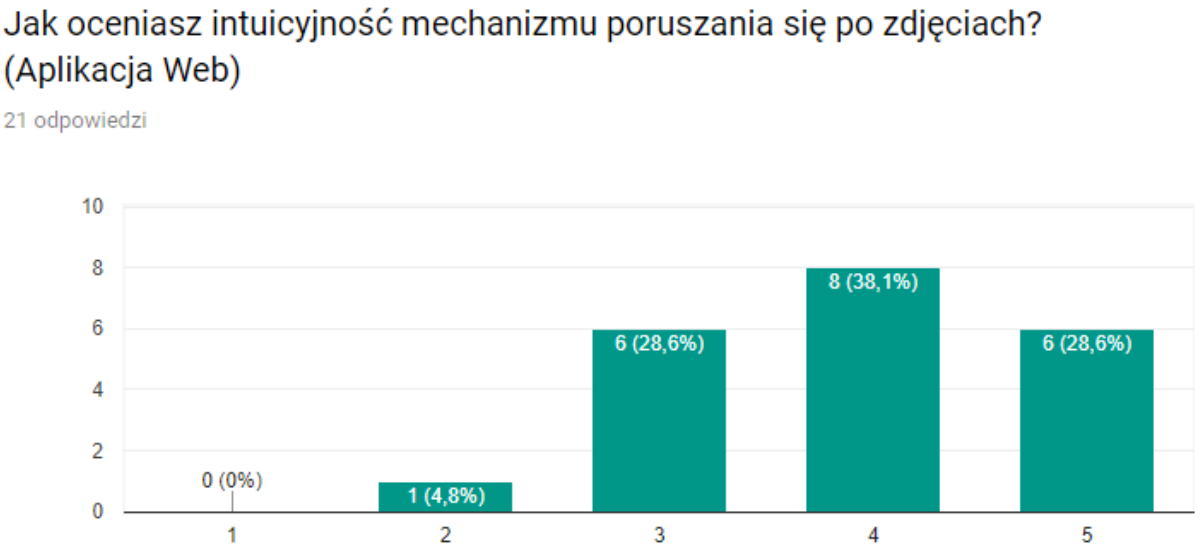


Zdjęcia zostały ocenione dobrze, jako niedociągnięcia zostały przedstawione niedokładne i pełne artefaktów bieguny sfery – jest to niestety bolączka amatorskiego robienia zdjęć sferycznych, bez profesjonalnego sprzętu opierając się na kilkudziesięciu zdjęciach bardzo ciężko jest je gładko skleić. Ten element można by poprawić używając wspomnianego wcześniej obiektywu

szerokokątnego w smartfonie lub specjalnych dedykowanych aparatów do tworzenia zdjęć i filmów sferycznych np. Samsung Gear360 lub Rico Theta.



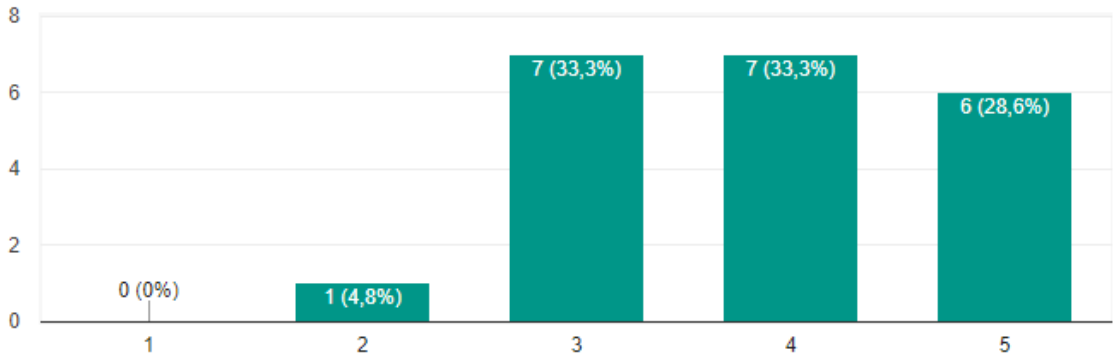
Tutaj wyniki były zdecydowanie niezadowalające, w aplikacji Unity jako znaczników do zmiany sceny użyto spriteów ptaków, które uznano za mocno nieintuicyjne i niewidoczne. W naszym odczuciu ptaki miały być oryginalne i artystyczne, jednak okazało się, że lwia część ankietowanych nie podobała się forma tej funkcjonalności.



W aplikacji Web sam mechanizm poruszania się i zmiany scen był bardzo podobny do aplikacji Unity, poza samym znacznikiem, który był bardziej przyjazną i znaną użytkownikowi białą obręczą. Ta jedna różnica między mechanizmem tak bardzo spolaryzowała opinie.

Jak oceniasz wygodę mechanizmu uzyskiwania informacji o obiektach? (Aplikacja Unity)

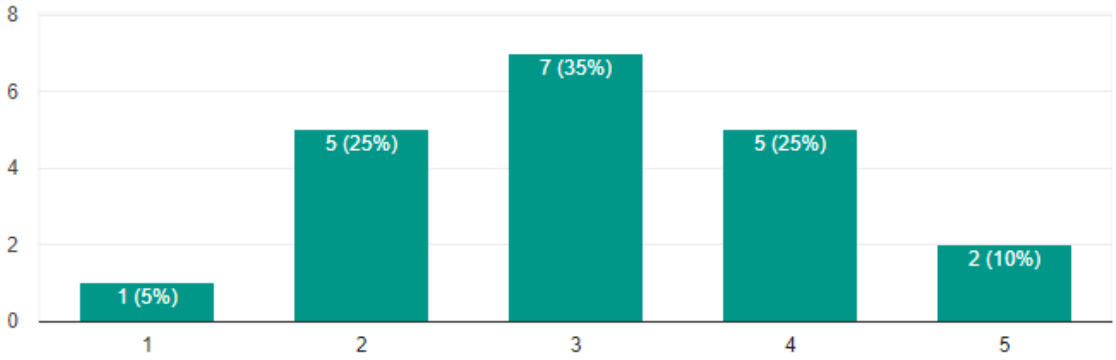
21 odpowiedzi



Uzyskiwanie informacji o obiekcie jako takie zostało ciepło przyjęte i uznane za bardzo ciekawy pomysł, jednak niektórzy mieli problem z odczytaniem liter z opisów, co niestety bierze się ze słabej rozdzielczości obrazu wyświetlanego przez smartfon, który dodatkowo jest poddawany obróbkom oraz aberracji chromatycznej opisanej wyżej.

Jak bardzo było to immersyjne doświadczenie?

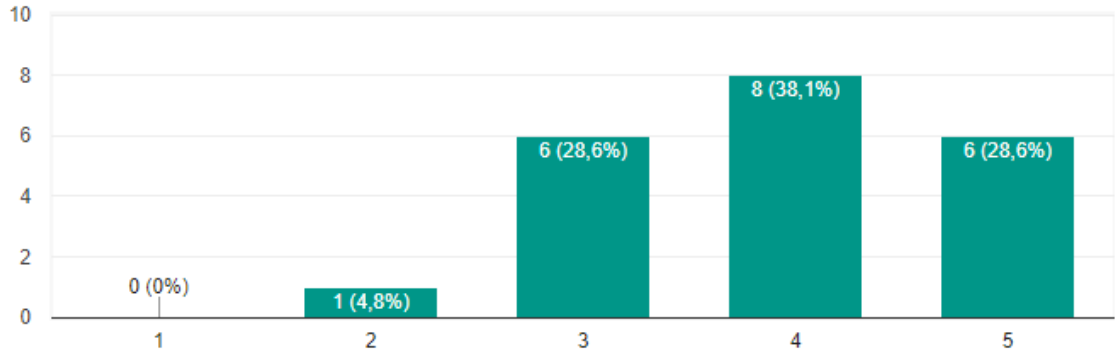
20 odpowiedzi



Immersyjność zaburzała przede wszystkim wspomniana słaba jakość obrazu spowodowana aberracją chromatyczną oraz zniekształceniami, które niestety w dzisiejszych czasach są bolączką układów VR.

Jak pozytywnie oceniasz całe doświadczenie?

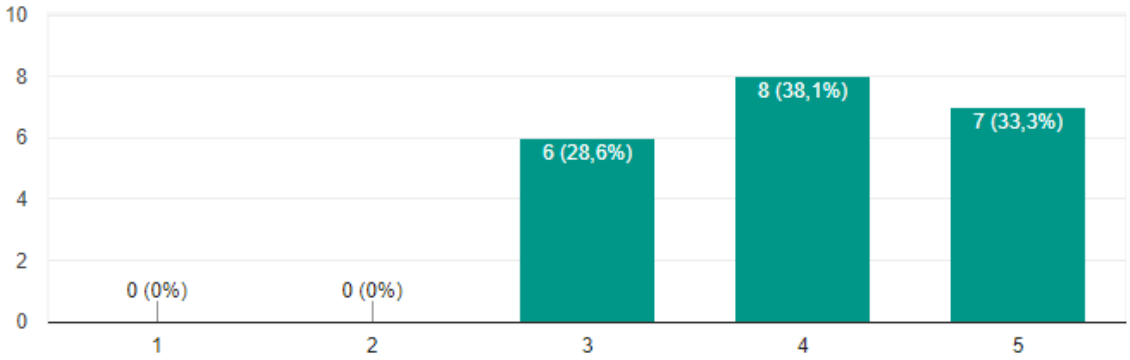
21 odpowiedzi



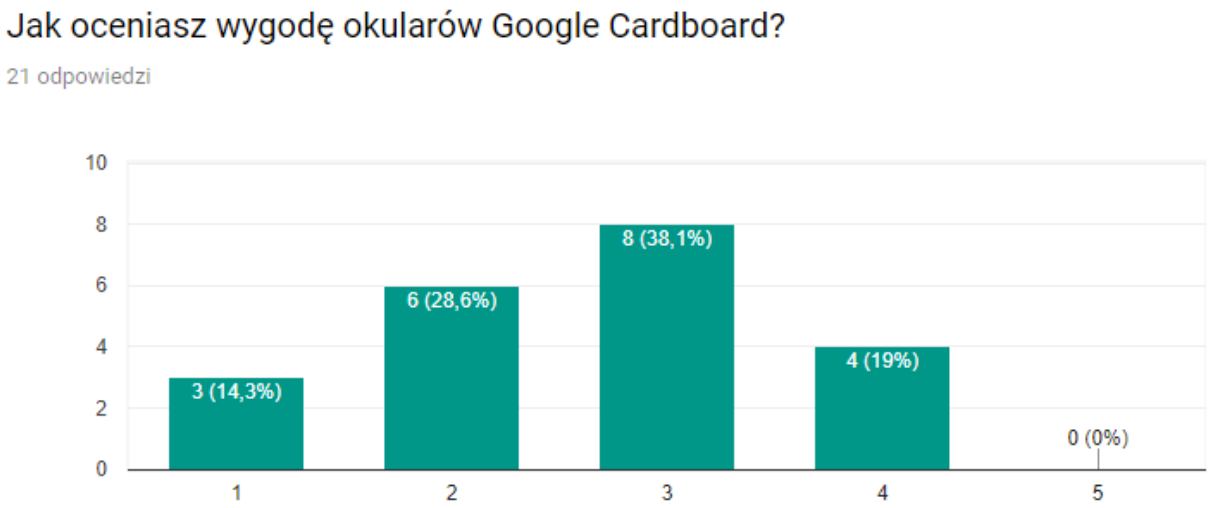
Doświadczenie odebrane pozytywne, część ludzi nie miała wcześniej bezpośredniej styczności z tą technologią, więc było to dla nich totalnie nowe i oryginalne doświadczenie.

Jak oceniasz wygodę okularów FiiT VR?

21 odpowiedzi



Okulary FiiT VR otrzymały wysoką ocenę. Są wygodne i mają różne udogodnienia w postaci rożnych regulatorów, które pozwalają je sobie dobrze spersonalizować.

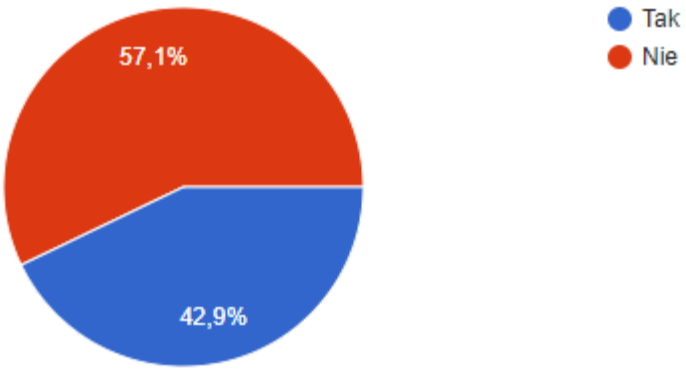


Nie było tutaj zaskoczenia, układ Google Cardboard jest z pewnością bardziej znaną opcją produkowaną i promowaną przez giganta tej branży jednakże w rzeczy samej pozostaje kartonem z soczewkami, co przekłada się na jego słabą ergonomię. Do jego plusów jednak na pewno należy cena.

Większość głosów na tak sugeruje, że takie projekty mogą mieć w przyszłości rację bytu i być wykorzystywane w celach komercyjnych lub edukacyjnych.

Czy uważasz, że wirtualne zwiedzanie tego typu mogłoby zastąpić prawdziwe?

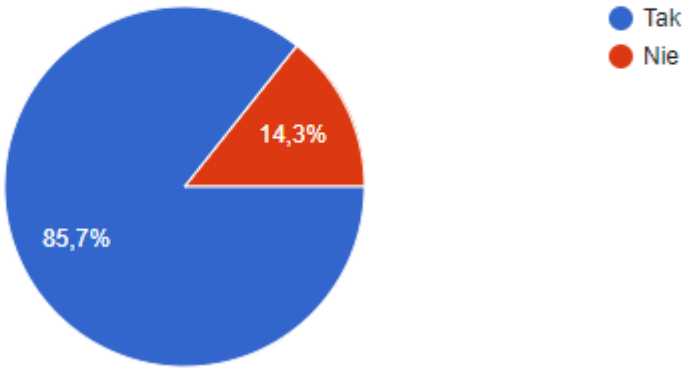
21 odpowiedzi



Na dzień dzisiejszy technologie wirtualnej rzeczywistości są na bardzo młodym etapie, można powiedzieć ta technologia dopiero raczkuje. Wynik nie dziwi – coś materialnego jest uznawane za bardziej autentyczne, a autentyczność to w dzisiejszych czasach to bardzo cenny towar. Istnieje w niej jednak wielki potencjał i jest możliwe, że w przyszłości technologia będzie w stanie bardzo wiernie odwzorowywać rzeczywistość – na tyle wiernie, że różnica pomiędzy nimi się zatrze.

Czy Twoim zdaniem jest to przyszłościowa aplikacja?

21 odpowiedzi



Tak jak zostało powiedziane wcześniej, technologia ta ma ogromny potencjał, a aplikacja pozwalająca wirtualnie zwiedzić jakiś (niedostępny) zakątek świata wydaję się być pod nią uszyta.

Czynniki od których zależy zostały przedstawione jako: jakość finalnego produktu, autentyczność wrażeń oraz cena.

8.2. Część 2 – pytania otwarte

Zostały zadane 3 pytania, a najbardziej powtarzające się wyrazy z odpowiedzi ankietowanych zostały zebrane w chmurę słów.

Jakie są Twoje odczucia po wirtualnym spacerze?

Rysunek 35 - chmura słów - pierwsze pytanie

Co wpłynęłoby na to, że byłbyś w stanie zapłacić za takie doświadczenie?

Rysunek 36 - chmura słów - drugie pytanie

Co Twoim zdaniem można by poprawić w prezentowanym rozwiązaniu?

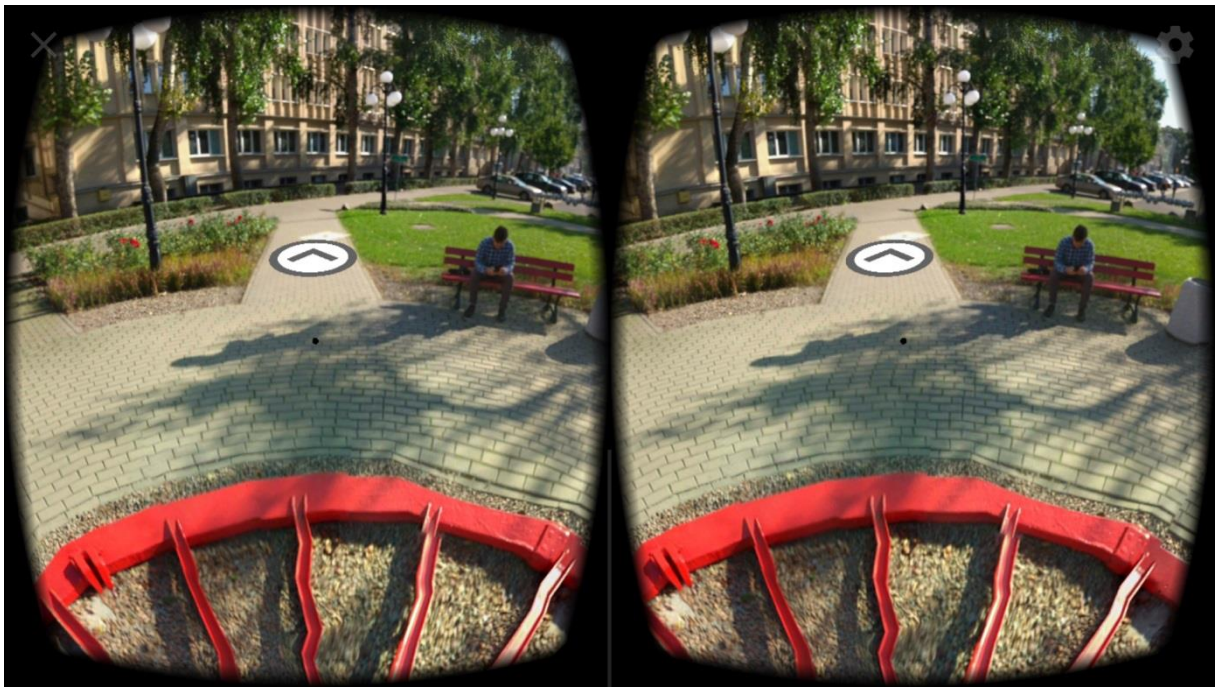
Rysunek 37 - chmura słów - trzecie pytanie

8.3. Wnioski wynikające bezpośrednio z badania

Przede wszystkim przyjrzeliśmy sprite'om, które służyły jako znaczniki do przechodzenia do następnego obszaru. Początkowo myśleliśmy że pierwotne rozwiązanie (w postaci ptaków, zdjęcie poniżej) będzie eleganckie i oryginalne. Po przeprowadzeniu badań jednak spotkaliśmy się z wielką dezaprobatą i niezadowoleniem większości badanych. W myśl User Experience, które mówi że doświadczenia powinny być jak najbardziej pozytywne, ptaki zostały zamienione na bardziej intuicyjne strzałki na ziemi, przypominające te z Google Street View. Zamienienie znaczników na formę znaną każdemu z podobnych aplikacji spotkało się z pozytywnym odzewem i oceną.

Jako że naszym zamiarem jest rozwijanie aplikacji w przyszłości zdecydowaliśmy sobie wziąć wszystkie opinie do serca, ale skupić się od tego momentu już tylko na jednej aplikacji. Padło na aplikację Unity ponieważ pozwala na lepsze dostosowywanie jej na potrzeby użytkowników, API od Google jest dość zamkniętym systemem i rozwój pewnych funkcjonalności mógłby być trudny, albo wręcz niemożliwy. Niestety, większość niedociągnieć jakie nie podobały się ankietowanym to kwestia hardware'u a nie software'u, co wymaga dużych nakładów finansowych i czekania na wyjście nowocześniejszych technologii.

Rysunek 38 - PRZED: Sprite'y służące do przejść jako ptaki na niebie



Rysunek 39 - PO: Sprite'y służące do przejść jako strzałki na ziemi

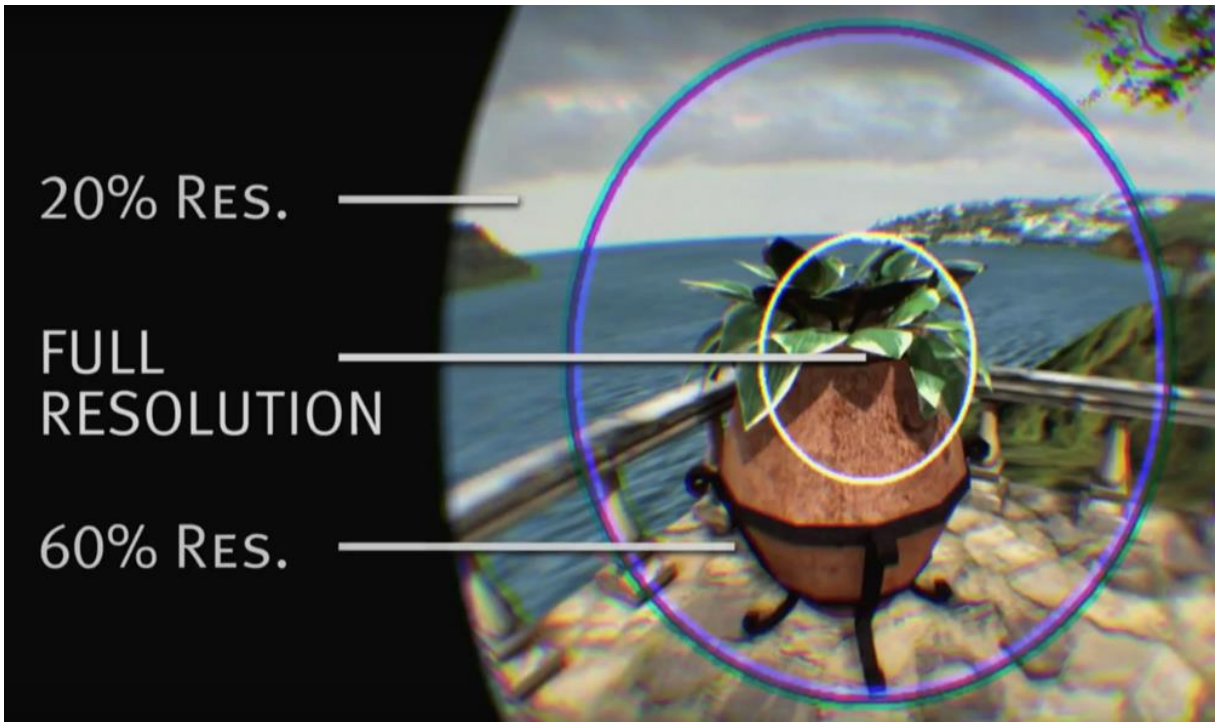
9. Perspektywy rozwoju i przyszłość wirtualnej/rozszerzonej rzeczywistości (M. Szumlański)

Aktualnie jesteśmy w środku, czym można określić jako „druga fala" wirtualnej rzeczywistości. W późnych latach 80 oraz wczesnych 90 była moda na gogle, ale były dość toporne i mało przekonujące. To była pierwsza fala. W dzisiejszych czasach postęp rozwoju układów GPU i podobnych jest nieuchronny, nie tylko z powodu apetytu na wydajność rynku gier/video, układy GPU oraz FPGA wykorzystuje się również w sieciach neuronowych oraz innych formach AI (ang. Artificial Intelligence). Giganty pokroju Google czy Facebook i ich centra danych nie są w stanie poradzić sobie ze wszystkimi zadaniami nie przenosząc większości obciążeń związanych z ich przetwarzaniem na wspomniane układy. Dodajmy do tego rosnący w sile rynek kryptowalut (pomijając dyskusję na temat tego czy jest bańką czy nie) i wiele wiele innych. Rynek chipów czeka (r)ewolucja, która przyniesie za sobą jeszcze lepszy VR niż myślimy.

9.1. Rozwój wyświetlaczy

Wirtualna rzeczywistość, aby wiernie symulować jakieś środowisko ma problem z rozdzielczością – jest ona aktualnie za mała do tego celu. Nawet przy najlepszych układach graficznych VR wciąż wygląda zbyt nieostro i zmatowiale niż ktokolwiek by chciał. Ale wydajność to jedno, dużą rolę odgrywają również same wyświetlacze, które mają pewne ograniczenia względem ilości pikseli, które fizycznie mogą reprezentować. Obecnie Google pracuje nad wyświetlaczem VR z 10x większą ilością pikseli niż dostępne na rynku gogle[15]. Współpracując z firmą Sharp chce stworzyć OLED-owy wyświetlacz VR o matrycy 20 Mpix. Nie jest to oczywiście jeszcze zagęszczenie pikseli, które pozwoli na symulowanie obrazu ludzkiego oka, ale jest to z pewnością krok w dobrą stronę i na pewno wniesie VR na wyższy poziom. 20 MP na oko przy 90-120 klatkach wyświetlanego obrazu niesie za sobą oczywiście ogromne wymogi szybkości przesyłania danych w granicach 50-100 Gb/sekundę. Jedna z najbardziej aktywnych korporacji w branży – NVIDIA estymuje, że jesteśmy 2 dekady od poziomu rozdzielczości, który oszukałby nasze oczy, że symulowany świat jest, w rzeczy samej, prawdziwy[13]. NVIDIA pracuje obecnie również nad dodatkowymi technologiami jak tzw. „foveated rendering" [14], które może drastycznie zmniejszyć narzut obliczeniowy do renderowania obrazu w

wirtualnej rzeczywistości. W skrócie jest to technika renderowania obrazu, która używa systemu śledzenia wzroku użytkownika, która renderuje tylko to, co koniecznie, zmniejszając jakość obrazu znajdującego się poza widzeniem peryferyjnym.



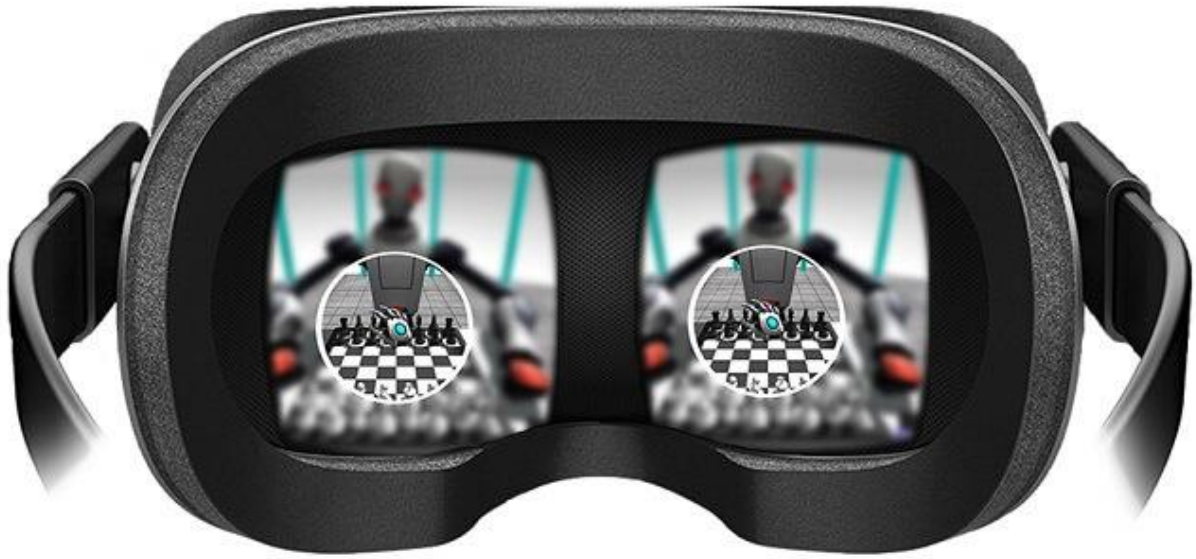
Rysunek 40 - Foveated rendering

9.2. Systemy śledzenia wzroku

Skoro już o systemach śledzenia wzroku użytkownika mowa – w ostatnim roku japońska firma FOVE wydała pierwsze na świecie okulary VR ze wspomnianym systemem wbudowanym w nie. Technologia ta zapowiedziała się na tyle obiecująco że wiele gigantów branży zakupiło nabyło start-upy zajmujące się rozwojem tej techonologii, aby wyposażyć w nią swoje urządzenia - Facebook/Oculus zakupił Eye Tribe pod koniec 2016, Google wykupiło firmę zwaną Eyefluence, Apple nabyło SMI, a z kolei Intel zainwestował 4,6 miliona dolarów w AdHawk[17].
Jakie to ma znaczenie[16]? Przede wszystkim, wydajność – wspomniany foveated rendering i inne technologię mogą delegować zasoby tam, gdzie jest to faktycznie potrzebne. Takie mądre używanie zasobów zmniejsza ogromną barierę wejścia (wydajności) w VR, a także daje twórcom większe pole do manewru i możliwość tworzenia jeszcze wiarygodniejszych efektów wizualnych. Inna sprawa – technologia taka może całkiem wiernie symulować skoncentrowanie wzroku, co bez wątpienia jest wizualnym usprawnieniem,

które do tej pory było tu nieobecne. System śledzenie wzroku daje również nowe możliwości co do projektowania interfejsów użytkownika oraz UX. Obecnie w urządzeniach, które bazują na wyświetlaczu, kiedy chcemy wykonać jakąś akcję musimy powiedzieć urządzeniu co chcemy aby zrobiło. Robimy to zazwyczaj dotykając jakiś konkretny obszar na ekranie albo wskazując jakieś rzeczy kursorem (używając np. myszki). Zanim to jednak zrobimy, zawsze przecież patrzymy na to z czym chcemy wejść w interakcję i w tym właśnie momencie do akcji wchodzi wspomniana technologia. Odcina ona pośrednika i pozwala bezpośrednio angażować się w zawartość zwyczajnie na nią patrząc. Da to nowe możliwości tworzenia interfejsów, które będą naturalne i niezwykle precyzyjne, efektywnie zastępując kursory i większość interakcji opartych na dotyku. Taka wzrokowa interakcja jest również dyskretna i może pozwolić nam na używanie immersyjnych urządzeń w ciasnych przestrzeniach publicznych itp. Kolejną zaletą systemu śledzenia wzroku jest to, że daje twórcom dostęp do niezrównanej ilości danych odnośnie użytkowania – będą wiedzieli na co użytkownicy patrzyli, co ignorowali w trakcie swojego doświadczenia, a także będą mogli zbadać zaangażowanie poprzez obserwację i śledzenie źrenic. Rozszerzone ludzkie źrenice zdradzają m.in. fizycznie pociąganie, psychiczne napięcie albo zaangażowanie emocjonalne. Idąc tą ścieżką może będzie możliwość przewidzenia akcji użytkownika **zanim ją w ogóle wykona.** Te wszystkie informacje mogą pomóc stwórcom na stworzenie immersyjnego oprogramowania które w 100% reaguje na emocje użytkownika i naprawdę rozumie co chodzi im po głowie w trakcie.

Taki system pozwoli również na tworzenie nowych rodzajów interakcji i mechanik rozgrywki, które nie były nigdy wcześniej możliwe – wirtualne postacie, będą teraz np. wiedziały kiedy użytkownik na nie spojrzy, a kiedyś może nawet przeanalizuje gdzie patrzy i dlaczego. Użytkownicy będą mogli celować za pomocą swoich oczu, podejmować decyzje w dialogach po prostu spoglądając na którąś opcję oczyma i znacznie wpływać na wirtualny świat wokół nich za pomocą praktycznie podświadomych gestów. Otwiera to wiele możliwości dla kreatywnej narracji i projektowania interakcji.



Rysunek 41 - Sterowanie za pomocą wzroku

Technologia ta już jest silnie rozwijana, do tej pory śledzenie wzroku polegało na robieniu setek zdjęć na sekundę oczom i używając algorytmów by je uporządkować i ustalić pozycję. Często taki sposób potrzebuje jasnych LED-ów do rozświetlania oczu dla aparatów. To wszystko powodowało opóźnienia i potrzebuje mocy – choć niewielkiej. Wspomniane wcześniej AdHawk zajmuje się rozwijaniem technologii o krok wprzód, która zastępują aparat ultra kompaktowymi elektromechanicznymi systemami zwanymi MEMS, a te zaś są tak małe, że aż niewidoczne przez ludzie oko. MEMS niweluje problem zasobożernego przetwarzania obrazów, co przekłada się na szybkość, większą kompaktowość oraz energooszczędność urządzeń VR które z nich korzystają. Tak samo jak inne technologie śledzenia wzroku AdHawk przewiduje gdzie użytkownik spojrzy do 50 milisekund wprzód. System MEMS posiada promień LED-owy, który skanuje oko i używając fotodiody wykrywa pozycję oka. Wszystko działa na CMOS-ie. Rozwiązanie pobiera bardzo niewiele energii, a projekt składa się z 2 czipów i szeregowej magistrali (trójprzewodowy interfejs). Jest również bardzo lekki, co sprawia że jest idealny dla każdego urządzenia – gogli czy smartfonu. Koszty mają wynosić od 10 dolarów na oko.

9.3. Inne projekty

AdHawk ma w planach również inny projekt – trójwymiarowy sensor gestów. Sensor ten miałby projektować sferę (10 cm objętości) naokoło np. tarczy zegarowej smartwatcha. Rzekomo miałby móc rozpoznawać gesty do 25 mikronów rozdzielczości na wszystkich osiach X,Y,Z. Byłoby możliwe np. pisanie

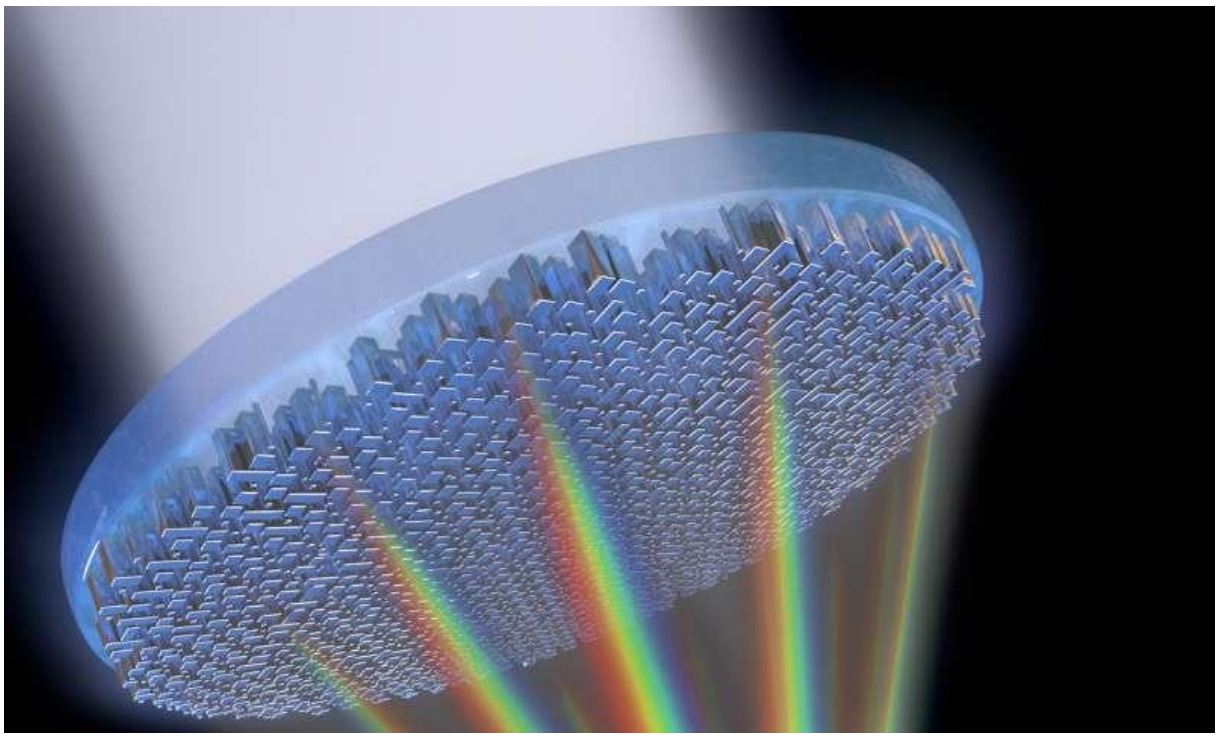
na wirtualnej klawiaturze na dowolnej powierzchni. Wiele rozwiązań zapożycza z ich technologii śledzenia oczu.
Obecnie również ciekawym rozwojem w jakim idzie branża są bieżnie, które zapewniają nieograniczoną płaszczyznę, po której użytkownik może się poruszać, zapewniając jeszcze większą immersję i dając złudzenie faktycznego poruszania się po symulacji. Przykładem jest np. VirtuSphere[18], która jest sferą położoną na specjalnej platformie która pozwala jej się obracać we wszystkie strony w zależności od ruchów użytkownika, który znajduje się w środku. Infinadeck to inny, trochę odmienny przykład. W tym przypadku mamy do ze zwykłą poziomią bieżnią ale można poprowadzić ją we wszystkie kierunki i dynamicznie wraca do środkowego położenia. Takie rozwiązania mogą również poszerzyć zakres wykorzystywania technologii VR do np. ćwiczeń czy treningów wojskowych.

Rysunek 42 - VirtuSphere

9.4. Metasoczewki

Ciekawym, i dość przełomowym ostatnim odkryciem w dziedzinie optyki było opracowanie przez grupę badaczy z Harvard John A. Paulson School od Engineering and Applied Science pierwszej pojedynczej soczewki, która potrafi skupić spektrum światła fal od 470nm do 650nm w tym samym miejscu i z

wysoką rozdzielczością[19]. W konwencjonalnych soczewkach zostaje to osiągnięte przez układanie kilku soczewek obok siebie. Metalenses – płaskie powierzchnie, które używają nanostruktur do skupiania światła zapowiadają się, że zrewolucjonizują optykę zastępując wielkie zakrzywione soczewki używane w optyce. Do tej pory takie płaskie soczewki nie potrafiły skupiać światła spektrum za dobrze, w efekcie występowało rozszczepienie światła. Nie jest to jeszcze produkt doskonały, gdyż widzialne spektrum mieści się w granicach 390 nm do 700nm (a soczewki skupiają zakres 470-650)[20], a do tego sprawność soczewek wynosi 20%, co efektywnie oznacza 80% utraty światła (soczewki fresnel w HTC Viveo charakteryzują się tą wartością na mniej niż 5%). Do tego dochodzi wysoki wariacja pomiędzy sprawnościami dla różnych długości fal (dla 405nm 86% dla 660nm 66%) co znaczy tyle, że niektóre kolory prezentują się jaśniej niż inne.
W następnym kroku badacze zamierzają zwiększyć ich średnicę do 1cm. Otworzyłoby to nowe możliwości zastosowania w aplikacjach zarówno VR jak i AR. Teoretycznie takie soczewki mogłyby pozbyć się efektu chromatycznej aberracji.



Rysunek 43 - Soczewka z nanostrukturami

10. Podsumowanie

Przechodząc do meritum wyciągnięto odpowiednie wnioski:

Po zbudowaniu dwóch aplikacji i ich porównaniu, stwierdzono, że zarówno zbudowanie aplikacji w postaci instalacyjnej na systemy Android jak i w wersji webowej, było łatwe, przyjemne i dające dużo satysfakcji.
Środowisko Unity zawiera szereg wygodnych narzędzi to tworzenia, co dość ułatwia cały proces wytwarzania oprogramowania, z drugiej strony biblioteki w języku JavaScrpit, również posiadają gotowe komponenty, które są eleganckie i standardowe dla aplikacji VR. Jednakże API dla drugiej aplikacji jest bardzo zamknięte i ma dość ograniczone pole manewru w stosunku do rozwiązania napisanego pod silnikiem graficznym Unity.
Myśląc przyszłościowo o wytwarzaniu aplikacji, które miałyby być bardziej elastyczne tzn. mające możliwość uruchomienia na różnych platformach systemowych uznano, że z pewnością lepsze byłoby środowisku Unity z racji opcji eksportu na pod 10 różnych platform.
Mimo wielu zalet podejście drugie czyli wersja w czystym JavaScript, okazała się lepsza w jakości stitchingu zdjęć, czyli łączenia ich w wirtualną sferę. Wynikało to prawdopodobnie ze złego doboru Shadera dla sfery, co spowodowało lekkie zniekształcenia budynków.

Badanie user experience wyraźnie pokazały, że jest zainteresowanie wśród ludzi wirtualną rzeczywistością, mimo pewnych niedoskonałości. Generalnie można zaobserwowano pozytywne nastawienie potencjalnych użytkowników do przedstawianego wirtualnego świata.

Podsumowując, dzięki realizowanej pracy autorzy pracy mieli okazje na dobre poznanie potężnego środowiska jakim jest Unity, które pozwala na bardzo dużo możliwości. Uznano je jako lepsze do tego typu aplikacji mimo drobnego potknięcia, nad którym autorzy zamierzają pracować.
W przyszłości planowany jest rozwój aplikacji, ponieważ uważa się, że w aplikacjach tego typu drzemie ogromny potencjał. Obecnie są na dość wczesnym etapie rozwoju i ich prawdziwe zapotrzebowanie nadejdzie w przyszłości wraz z rozwojem technologii opisanych w rozdziale 9. .
Autorzy nie chcą ograniczać się do spaceru po AGH, w przyszłości chcą stworzyć aplikacje pozwalającą każdej osobie na ziemi zwiedzić każde miejsce dotąd niezbadane lub nie dostępne

11. Spis ilustracji

12. Bibliografia

[1] https://www.merriam-webster.com/dictionary/virtual%20reality

[2] https://www.merriam-webster.com/dictionary/augmented%20reality

[3] https://en.wikipedia.org/wiki/Mixed_reality

[4] http://www.komputerswiat.pl/nowosci/sprzet/2017/29/google-glass-powracaja-jako-enterprise-edition.aspx

[5] https://play.google.com/store/apps/details?id=com.google.android.street

[6] https://unity3d.com/unity

[7] https://unity3d.com/learn/tutorials/topics/graphics/gentle-introduction-shaders

[8] https://docs.unity3d.com/Manual/SL-ShadingLanguage.html

[9] https://docs.unity3d.com/Manual/UICanvas.html

[10] https://developers.google.com/vr/develop/unity/controller-support

[11] https://github.com/googlevr/gvr-unity-sdk/releases/tag/v1.110.0

[12]https://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html

[13]https://uploadvr.com/nvidia-estimates-20-years-away-vr-eye-quality-resolution/

[14]https://en.wikipedia.org/wiki/Foveated_rendering

[15]https://www.roadtovr.com/google-developing-vr-display-10x-pixels-todays-headsets/

[16]https://medium.com/futurepi/why-eye-tracking-is-a-huge-deal-for-vr-ar-683e971652ee

[17]http://www.tomshardware.co.uk/intel-invests-4.6m-adhawk-eye-tracking,news-57098.html

[18]https://en.wikipedia.org/wiki/VirtuSphere

[19]https://www.seas.harvard.edu/news/2018/01/single-metalens-focuses-all-colors-of-rainbow-in-one-point

[20]https://www.ncbi.nlm.nih.gov/pubmed/27257251

[21]http://www.hypergridbusiness.com/2016/08/review-fiit-vr-2s-is-my-new-favorite/

[22]https://pl.wikipedia.org/wiki/Widzenie_stereoskopowe

[23]http://smus.com/vr-lens-distortion/

[24]https://pl.wikipedia.org/wiki/Aberracja_chromatyczna

[25]https://pl.wikipedia.org/wiki/Stereofotografia

[26]https://pl.wikipedia.org/wiki/Anaglif

[27]https://pl.wikipedia.org/wiki/Stereogram

[28] https://pl.wikipedia.org/wiki/Stereoskopia

[29] https://developers.google.com/vr/develop/web/vrview-web