

Raport techniczny z realizacji projektu *Stilleben*

dr Jakub Woynarowski, dr Jan K. Argasiński

Grudzień 2019

UBU-19-01

Abstrakt. Termin „Stilleben”, zapożyczony z obszaru sztuk wizualnych, oznacza dosłownie „ciche życie”, ale również „martwą naturę”. Projekt pod tym tytułem, zrealizowany przez Jakuba Woynarowskiego i Jana K. Argasińskiego w ramach laboratorium twórczego programowania UBU Lab na Uniwersytecie Jagiellońskim w Krakowie, stanowi posthumanistyczną opowieść o domowym wnętrzu, traktowanym nie tyle jako funkcjonalna „maszyna do mieszkania”, co raczej niesamowity „gabinet osobliwości”. Przyswajając poszczególne warstwy wizualnego tekstu odbiorca pozostaje w stałym ruchu między dostrzegalną powłoką i niewidoczną podszewką świata przedstawionego, uczestnicząc tym samym w jego utajonym „cichym życiu”. Interaktywna, wirtualna wystawa „Stilleben” powstała zarówno w wersji stacjonarnej, jak i mobilnej; naturalnym dopełnieniem projektu VR zrealizowanego w laboratorium jest narracyjny artbook, wzbogacony o elementy „rzeczywistości rozszerzonej”.

Raport techniczny z

UBU lab

Uniwersytet Jagielloński

Wydział Zarządzania i Komunikacji Społecznej

Instytut Kultury

ul. Łojasiewicza 4, p.2.120, 30-348 Kraków

piotr.marecki@uj.edu.pl

www.ubulab.edu.pl

Praca naukowa finansowana
w ramach programu Ministra Nauki
i Szkolnictwa Wyższego pod nazwą
„Narodowy Program Rozwoju Humanistyki”
w latach 2016-2019

Copyright © by the Authors, 2019

Spis treści

1. Wirtualna wystawa jako gra eksploracyjna • 3
 - 1.1. Wprowadzenie do problematyki projektu • 3
 - 1.2. Inspiracje artystyczne • 4
 - 1.3. Konstrukcja utworu • 7
2. Scenorysy • 9
 - 2.1. Aplikacja VR - scenorys z omówieniem • 9
 - 2.2. Aplikacja VR - plan interakcji: komunikacja wizualna • 18
 - 2.3. Aplikacja AR - scenorys z omówieniem • 23
3. Implementacja • 31
 - 3.1. Aplikacja VR - Unity, VRTK, SteamVR, HTC Vive • 31
 - 3.1.1. Unity Engine • 31
 - 3.1.2. Virtual Reality Toolkit • 32
 - 3.1.3. Steam Virtual Reality • 32
 - 3.1.4. HTC Vive • 32
 - 3.1.5. Sceny • 33
 - 3.1.6. Obiekty • 34
 - 3.1.7. Skrypty/interakcje • 35
 - 3.2. Aplikacja AR - Unity, Vuforia AR, Android SDK • 36
 - 3.2.1. Unity Engine • 36
 - 3.2.2. Vuforia Augmented Reality SDK • 36
 - 3.2.3. Android SDK • 38
 - 3.2.4. Obiekty, skrypty i interakcje w podziale na sceny • 38
4. Podsumowanie • 44
 - 4.1. Testy i release AR • 44

1. Wirtualna wystawa jako gra eksploracyjna

1.1. Wprowadzenie do problematyki projektu

Termin „Stilleben” oznacza dosłownie „ciche życie”, a także - jako jego rewers - „martwą naturę”. Obecne we wczesnonowożytnym malarstwie alegoryczne przedstawienia przedmiotów, skomponowanych w swoiste eseje wizualne, znajdowały swoją paralelę również w postaci tak zwanych wunderkamer - artystycznych kompilacji „obiektów znalezionych” różnego pochodzenia, potraktowanych jako metaforyczne opisy „świata w miniaturze”. W ramach zmiennej i subiektywnie organizowanej struktury gabinetu osobliwości, poszczególne eksponaty stanowiły w istocie „ikonki”, odsyłające do innych elementów i tworzące w związku z tym rodzaj „interfejsu”. Nieraz gabinety tego rodzaju podlegały miniaturyzacji, stając się luksusowym meblem, zawierającym skomplikowany system szufladek i folderów z „ulubionymi” obiektami, a także zbiór gier, mechanizmów odtwarzających określone melodie oraz „aplikacje” biurowe, umożliwiające spisanie listu lub wykonanie rysunku.

Zdarzało się, że kolekcja podlegała „wirtualizacji”, a oryginalne obiekty zastępowano ich graficznymi wyobrażeniami, zgromadzonymi w opasłych katalogach, co potraktować można jako zapowiedź XIX-wiecznej idei Mundaneum - biblioteki mającej zgodnie z założeniem twórców (Paula Otleta i Henriego La Fontaine'a) gromadzić całą wiedzę świata. Zdaniem wielu badaczy koncepcja Mundaneum, uniwersalnej bazy danych, stanowi bezpośrednią zapowiedź powstania wirtualnej sieci internetowej. Jak widać, typowy dla wunderkamer sposób myślenia nie zanikł wraz z końcem epoki nowożytnej - współcześnie coraz częściej przyrównuje się praktyki internetowych blogerów do ekscesów nowożytnych zbieraczy kuriozów, a powszechne marzenia o „świecie w miniaturze” - jak się zdaje - z biegiem lat wcale nie wygasły.

Odwołując się do kulturowej tradycji, projekt „Stilleben” wykracza równocześnie w przyszłość, łącząc format wystawy z hipertekstową narracją, stanowiącą wizualną opowieść o domowym wnętrzu, traktowanym jednak nie tyle jako funkcjonalna, Corbusierowska „maszyna do mieszkania”, ile raczej „gabinet cudów” - źródło niesamowitości, która wdziera się w oswojoną przez człowieka przestrzeń. Warto w tym miejscu przywołać etymologię terminu „niesamowitość”, spopularyzowanego przez słynny esej Zygmunta Freuda „Das Unheimliche” z 1919 roku. Polskie „niesamowite” nie zawiera w sobie rdzenia, podstawowej figury architektonicznej, jaką jest w niemieckim odpowiedniku „das Heim” - dom, dom rodzinny, czy też miejsce, w którym czujemy się swojsko, wśród swoich, u siebie. Doznanie niesamowitości związane jest z zaburzeniem poczucia równowagi, jakie daje nam funkcjonowanie w oswojonym, racjonalnie zdefiniowanym środowisku. To właśnie dom lub mieszkanie - posłużmy się językiem biologii - stanowi nasz naturalny habitat (z łaciny: „habitat” = „mieszka”), siedlisko stwarzające najbardziej dogodne warunki dla życia.

Opowiadając o habitacie współczesnego człowieka, „Stilleben” przesuwa punkt ciężkości z osoby użytkownika na „użytkowane” przez niego środowisko. Pojawienie się niebezpieczeństwa w „oswojonej” przestrzeni, wskazuje na umowność wyznaczonego przez człowieka terytorium - stanowiącego zarówno azyl, jak i więzienie. Oto okazuje się, że zgeometryzowana, syntetyczna powłoka ukrywa wręcz barokowo rozbuchaną, organiczną strukturę; w ten sposób cywilizacja techniczna - wraz ze swymi solarnymi systemami lamp i roślinnymi instalacjami - staje się nową, sztuczną naturą (toczoną od środka przez pleśń i insekty, wiodące „ciche życie” w betonowej twierdzy). Eksploracja kolejnych warstw funkcjonalnej, minimalistycznej powłoki, maskującej dynamicznie zmienną, organiczną „podściółkę” staje się głównym tematem wirtualnej ekspozycji, której wieloetapowa percepcja znajduje analogię w strukturach narracyjnych gier interaktywnych.

Warto zauważyć, że formy symulacji, obecne w grach komputerowych i w elektronicznej sztuce interaktywnej, zmierzają do wykreowania środowiska immersyjnego, oferującego

użytkownikowi „doświadczenia graniczne”, które Janet Murray określa jako „przejście między światem codziennego doświadczenia, który traktujemy jako zewnętrzny wobec nas i prawdziwy, a fantazjami naszego umysłu. Kiedy jakaś historia pochłania naszą uwagę w takim stopniu, że powoduje głęboki stan zaangażowania odczuwamy prawdziwe emocje i wzruszenia, chociaż są one wywoływane przez fikcyjne obiekty” [zob. Katarzyna Prajzner: „Tekst jako świat i gra. Modele narracyjności w kulturze współczesnej”, Łódź, 2009, s. 35]. W myśl tej zasady, struktura rzeczywistości wirtualnej, przedstawionej w aplikacji „Stilleben”, odwołuje się zarówno do „świata codziennego doświadczenia”, jak i do deformujących go projekcji - „fantazji naszego umysłu”.

1.2. Inspiracje artystyczne

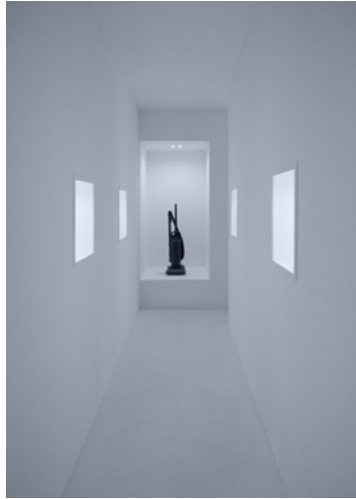
Projekt „Stilleben” - realizowany przez Jakuba Woynarowskiego we współpracy z Janem K. Argasińskim - kontynuuje wątki podejmowane przez Woynarowskiego w poprzednich pracach, takich jak eksperymentalny komiks „Hikikomori” (rozwinięty później w formie instalacji, prezentowanych m.in. w 2012 roku w krakowskim MOCAK-u i w 2015 roku podczas Think Tank lab Triennale we Wrocławiu), czy powieść graficzna „Martwy Sezon” (inspirowana twórczością Brunona Schulza).



Rysunek 1. Jakub Woynarowski, „Hikikomori”, komiks, 2007



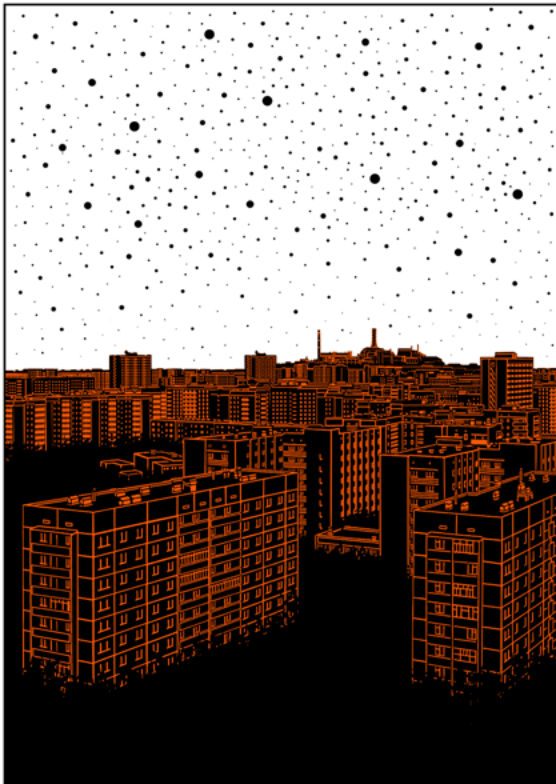
Rysunek 2. Jakub Woynarowski, „Hikikomori”, instalacja, MOCAK, Kraków, 2012



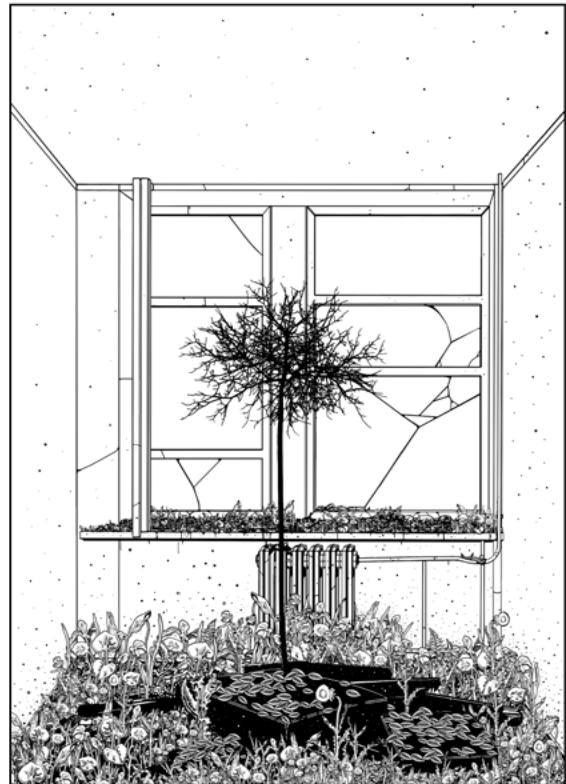
Rysunek 3. Jakub Woynarowski, „Hikikomori”,
instalacja, Think Tank lab Triennale, Wrocław, 2015



Rysunek 4. Jakub Woynarowski, „Hikikomori”,
instalacja, Think Tank lab Triennale, Wrocław, 2015



Rysunek 5. Jakub Woynarowski,
„Martwy sezon”, 2014



Rysunek 6. Jakub Woynarowski,
„Martwy sezon”, 2014

Podobnie jak w przypadku wymienionych utworów, punkt wyjścia dla rozbudowanej wizualnej narracji stanowi obserwacja najbliższego otoczenia człowieka, postrzeganego w optyce posthumanistycznej. Zbliżoną metaforyką posługuje się Stanisław Lem w swoim opowiadaniu „Ciemność i pleśń” [4], w którym niewielka niewielka przestrzeń mieszkalna ulega aneksji przez obcą formę życia o nieokreślonym statusie.

„Narracja jest przede wszystkim sprawą kosmologiczną. Opowiadając o czymś, zaczynasz niczym demiurg, który stwarza świat - świat, który musi być tak precyzyjny, jak to tylko możliwe, żebyś mógł się po nim zupełnie sprawnie i pewnie poruszać” - stwierdza Umberto Eco [U. Eco, „Wyznania młodego pisarza”, Warszawa 2011, s. 18-19.]. Wielopoziomowy, „kosmologiczny” sposób konstruowania narracji można odnaleźć między innymi w dwóch nowożytnych przykładach syntetycznych wizualnych „heksameronów”, opisujących pracę Boga podczas sześciu dni Stworzenia: w sekwencji genezyjskiej w „Kronice Norymberskiej” Hartmanna Schedla oraz w cyklu ilustracji do traktatu Roberta Fludda „Utriusque cosmi... historia” [2]. W obu przypadkach przedstawienie kreacyjnego mitu przybiera formę półabstrakcyjnych opowieści graficznych, skonstruowanych na podstawie geometrycznego modułu koła.

Grafika otwierająca dzieło Fludda przedstawia pierwotny chaos w formie czarnego kwadratu, opisanego formułą „Et sici in infinitum” („I tak dalej w nieskończoność”). Obraz ten do złudzenia przypomina późniejszy o trzy stulecia „Czarny kwadrat na białym tle”, określony przez jego twórcę, Kazimierza Malewicza, mianem „embrionu wszystkich możliwości”.

Można pokusić się o zestawienie przywołanych dzieł z „kosmologiczną” narracją, stworzoną w XX wieku przez El Lissitzky’ego [5], który korzystając z architektonicznych metafor, wykreował własny „heksameron”, zatytułowany „O dwóch kwadratach. Suprematystyczna bajka w sześciu kompozycjach”. Jak zauważa Odile Belkeddar we wstępie do brytyjskiego wydania książki: „Przeobrażona w obiekt wizualny, książka konstruowana jest jak budynek, odzwierciedlając zarówno ruch, jak i dźwięk. Jako projektant wystaw, El Lissitzky użyłby nawet ścian, podłogi i sufitu, jak sześciu powierzchni sześcianu” [O. Belkeddar, „Afterword”, w: El Lissitzky, „A Suprematist Tale of Two Squares in Six Constructions”, London 2014, s. 22.].

Podobną intuicję wyraża Paul Gravett opisując brytyjski projekt „PoCom-UK-001” - „brakujące ogniwo” między „hiperkomiksem” a „komiksem galeryjnym” (*gallery comics*): „Ambicją pomysłodawców - wciąż czekającą na realizację - jest to, by PoCom rozrósł się na trzy pozostałe ściany galerii, a następnie by panele połączone zostały kolejnymi historiami umieszczonymi na podłodze i suficie oraz przecinającymi diagonalnie przestrzeń galerii, by wreszcie wybuchnąć poza sześcian, tworząc rozkwitający organicznie zbiór opowieści, dostępnych wyłącznie w przestrzeni wirtualnej” [P. Gravett, „Comics Art” [3], s. 132].

Zbliżone strategie można odnaleźć w obrębie amerykańskiej sceny komiksu alternatywnego, przed wszystkim w kręgu magazynu „RAW”, wydawanym przez Arta Spiegelmana, autora takich powieści graficznych jak „Maus” i „W cieniu nieistniejących wież”. Warto porównać dwóch artystów wywodzących się z tego środowiska, stosujących podobną, „paradygmatyczną” metodę opowiadania ujętą w nielinearne struktury: Richarda McGuire’a [6] i Chrisa Ware’a. Zarówno „Tutaj” McGuire’a, jak i „Building Stories” Ware’a [11] stanowią hipertekstowe wizualne opowieści, obierające formę architektoniczną jako punkt wyjścia do narracyjnych eksperymentów.

Podobny, „architektoniczny” i „kosmologiczny” model opowieści graficznej odnajdziemy w „redukcji” cyklu powieściowego Marcela Prousta „W poszukiwaniu straconego czasu”, dokonanej przez François Ayrolesa, członka OuBaPo [7] (Ouvroir de bande dessinée potentielle - Warsztatu Komiksu Potencjalnego), francuskiego eksperymentalnego ruchu, wzorowanego na literackiej awangardzie OuLiPo. Ayroles, podobnie jak oulipijczycy w polu literatury, konsekwentnie poszerza granice komiksowego medium, opierając swoją strategię na wyrafinowanych „przymusach”, stymulujących kreatywność.

Ten niekonwencjonalny projekt wpisuje się niewątpliwie w szerszą tradycję „opowieści bez bohaterów”, wśród których można wymienić między innymi „Klatkę” („The Cage”) Martin Vaughn-Jamesa [10], „SPUK” Niklause Rügga [9] oraz „Rumble Strip: If You Want to Get Away with Murder, Buy a Car” Woodrow Phoenixa [8].

Wpisując się w nurt opisanych eksperymentów narracyjnych, „Stilleben” odsyła do tradycji wizualnych „heksameronów” opisujących kolejne fazy procesu genezyjskiego. Formę białego prostopadłościanu można w tym przypadku odczytywać również jako metaforę bliską „Sześcianowi Przestrzeni” - ezoterycznej idei spopularyzowanej przez Paula Fostera Case’a. W myśl koncepcji Case’a elementy składowe sześcianu - trzy osie, punkt centralny, sześć ścian i dwanaście krawędzi - miały odpowiadać dwudziestu dwóm literom hebrajskiego alfabetu. Zaczątków „Sześcianu Przestrzeni” można bowiem doszukiwać się w protokabalistycznym tekście Sefer Jeciry, opisującym proces stworzenia świata. Zaskakująco podobną myśl wyraża Jay David Bolter w swojej książce „Writing Space: Computers, Hypertext, and the Remediation of Print”, w której porównał alchemiczne i kabalistyczne kody ze współczesnymi interfejsami komputerowymi.

Tego rodzaju procedura odpowiada w pewnym sensie działaniom podejmowanym przez użytkownika „Stilleben”, otwierającego nowe rozdziały opowieści, podążając za półabstrakcyjnymi ikonami umieszczonymi we wnętrzu sterylnej „białego sześcianu”. Proces komunikacji jest jednak nieustannie przerywany za sprawą „buntowniczej” czarnej materii. Zbliżone strategie odnajdziemy we wczesnych przykładach literatury „interaktywnej”: „Życie i myśli JW Tristrama Shandy” Laurence’a Sterne’a (pełne narracyjnych „blackoutów”) oraz „Dzieje Świętej Rusi” Gustave’a Doré’s [1] (niektóre strony tej graficznej powieści zostają zalane przez abstrakcyjne plamy tuszu).

W klaustrofobicznej przestrzeni „Stilleben” tajemnicza „pramateria” wprawia w ruch obiekty codziennego użytku, w miarę upływu czasu powodując ich spektakularną dekonstrukcję. Biały kolor ścian, tkanin i elektrycznych urządzeń może budzić skojarzenie z iluzorycznym „domowym” spokojem, podszytym jednak lękiem o utratę „oswojonego” terytorium, funkcjonalnej „maszyny mieszkalnej”, zgodnej z modernistycznymi ideałami Bauhausu.

W ten sposób opozycje zdają się doskonale uzupełniać, a aspekty genezyjski i apokaliptyczny pozostają w ścisłym związku. Ostatecznie, w wyniku dynamicznej metamorfozy pusta biała przestrzeń staje się wybujałym ogrodem. W tym kontekście „Stilleben” można odczytywać jako uniwersalną opowieść o nieustannej cyrkulacji materii, przybierającej wciąż inne kształty.

1.3. Konstrukcja utworu

Symboliczna „podróż” użytkownika aplikacji „Stilleben” przebiega z zachowaniem jedności czasu, miejsca i akcji, a jej niezmienną ramą scenograficzną pozostaje naturalnych rozmiarów symulacja standardowej, jednoosobowej przestrzeni mieszkalnej. Podstawowym modułem architektonicznym jest tu minimalistyczna bryła „białego sześcianu”, nawiązującego do modernistycznej koncepcji wystawienniczej tzw. *white cube*’a (neutralnej przestrzeni ograniczającej bodźce zaburzające percepcję eksponatów), której swoją pracę „Le Vide” poświęcił w 1958 roku Yves Klein.

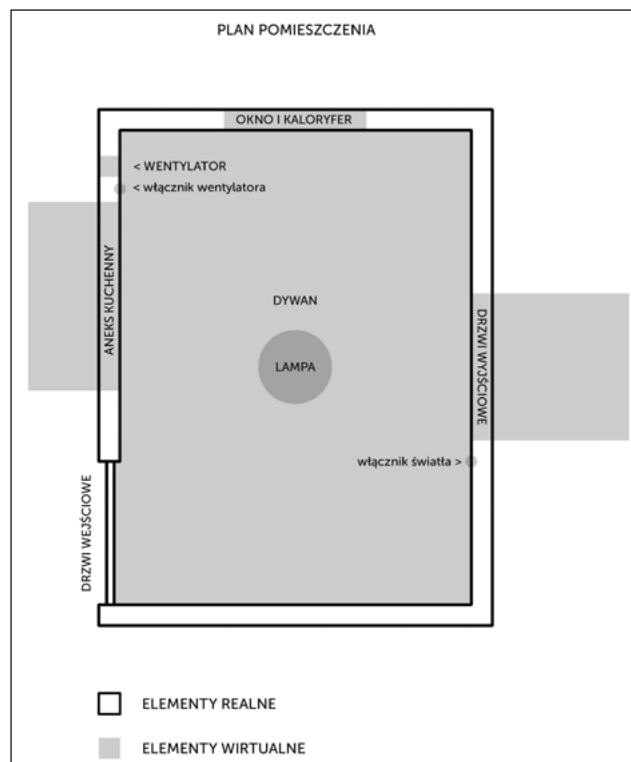
Eksplokacja kompaktowego mieszkania, stanowiącego model „świata w miniaturze”, przebiegałaby analogicznie do procesu genezyjskiego: sześć kolejnych etapów, związanych z aktywizacją sześciu powierzchni kubicznej przestrzeni (cztery ściany, sufit i podłoga), prowadziłyby do etapu siódmego, pozwalającego na bardziej pasywną kontemplację wykreowanego przez użytkownika efektu. W ten sposób odbiorca - realizując przewidzianą dla niego procedurę działania - każdorazowo bezpośrednio wpływałby na docelową formułę wirtualnej wystawy.

Fabularna oś projektu oscyluje wokół próby opanowania przez użytkownika „obcego żywiołu” - organicznej struktury, złożonej z drobnych czarnych cząstek, anektującej sterylność

białe pomieszczenie. Każde kolejne „zadanie”, wykonane przez odbiorcę, inicjuje następny etap eksploracyjnej gry i wiąże się z aktywizacją wybranych elementów architektury wnętrza, między innymi okna, przewodów wentylacyjnych, instalacji hydraulicznej, aneksu kuchennego, źródła światła, dywanu i drzwi wyjściowych. Wskazówki dotyczące kolejnych możliwych interakcji podejmowanych przez użytkownika ujawniają się w formie wizualnych eksplikacji, opartych na języku uproszczonych ikonek. Kod tej swoistej „literatury obrazkowej” odsyła do estetyki typowej dla skonwencjonalizowanych form grafiki informacyjnej. „Użytkowa” estetyka, znana ze świata grafiki projektowej, wbrew obowiązującym konwencjom zostaje jednak wykorzystana do snucia opowieści ocierającej się o abstrakcję i surrealizm. Oswojony przez nas język wizualny zaczyna ujawniać wewnętrzne pęknięcia, wynikające z nadmiernych uproszczeń lub za daleko idącej komplikacji. Tego rodzaju zabiegi sprawiają, że pozornie „przezroczysty” komunikat na temat funkcjonalnego środowiska, nabiera cech „niesamowitości”, wymykającej się czysto rozumowemu poznaniu.

Kolorystyka scenografii pozostaje od początku do końca utrzymana w skali szarości. Wszystkie „skokowe” zmiany w scenografii - związane z wprowadzeniem dodatkowych elementów i przejściem użytkownika do nowych etapów gry - poprzedzone są wyciemnieniem (efekt „migotania” prowadzącego do awarii elektryczności) oraz ponownym uruchomieniem oświetlenia wirtualnej przestrzeni.

„Stilleben” to stacjonarna wirtualna wystawa, zrealizowana w sterylnej prostopadłościenną przestrzeni, przygotowanej na terenie laboratorium UBU lab. Eksplorację świata przedstawionego przez użytkownika umożliwiał zaawansowany hełm wirtualnej rzeczywistości HTC Vive, wspomagany przez duet kontrolerów ruchowych i sensorów określających nie tylko kąt nachylenia głowy, ale również umiejscowienie swobodnie poruszającego się w trójwymiarowej przestrzeni użytkownika. Kształt wirtualnego pomieszczenia (realistycznie odtwarzającego standardowe elementy wyposażenia wnętrz) pokrywa się z rzeczywistym modułem architektonicznym. W związku z tym aplikację „Stilleben” można uznać za przykład realizacji koncepcji „rzeczywistości rozszerzonej” (AR) z wykorzystaniem narzędzi, zacierających granicę między realną architekturą a jej wirtualnym „aneksem”, wygenerowanym cyfrowo.

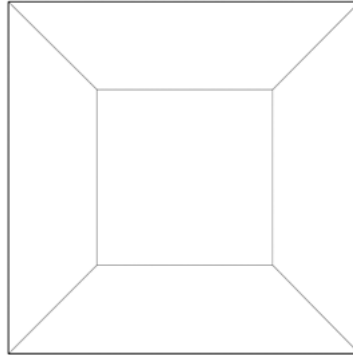


Rysunek 7. „Stilleben” w UBU Lab - plan pomieszczenia

2. Scenorysy

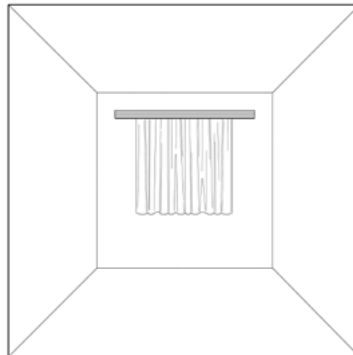
2.1. Aplikacja VR - scenorys z omówieniem

1. Z oślepiającej bieli wyłania się sterylna, prostokątna przestrzeń. Gładkie ściany emanują światłem. Blask jednak przygasa i w pewnym momencie pomieszczenie pogrąża się w półmroku.



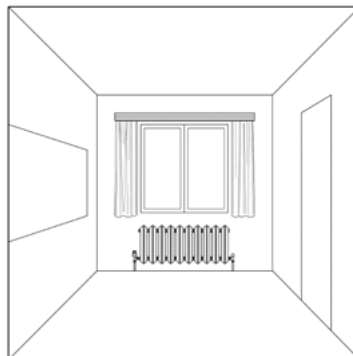
Rysunek 8. VR: scena nr 1

2. W półmroku stopniowo ujawnia się świetlista szczelina, szczelina, która po chwili okazuje się prześwitem między białymi zasłonami, zakrywającymi okno. Pojawiają się IKONKI (IKONKA nr 1, IKONKA nr 2), zachęcające użytkownika do odsłonięcia kotary (INTERAKCJA: odsłonięcie zasłon).



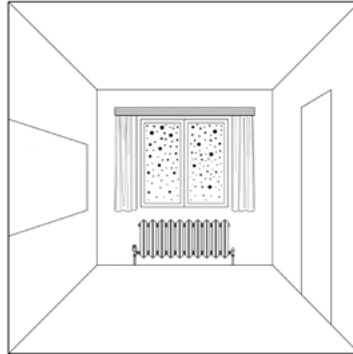
Rysunek 9. VR: scena nr 2

3. Po odsłonięciu zasłon w ramie okna ujawnia się nieograniczona biała przestrzeń. We wnętrzu pojawiają się wcześniej niewidoczne obiekty: kaloryfer we wnęce pod oknem, zarys prostokątnej wnęki w ścianie (po lewej stronie) i rama drzwi wyjściowych (po stronie prawej).



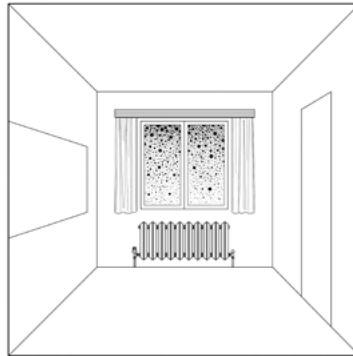
Rysunek 10. VR: scena nr 3

4. Szyba staje się coraz bardziej wilgotna i zaparowana. W białej przestrzeni na zewnątrz pojawiają się drobne ciemne punkciki, unoszące się w powietrzu. Pojawia się IKONKA (IKONKA nr 3) alarmująca użytkownika, że okno nie jest szczelnie zamknięte (INTERAKCJA: zamknięcie okna).



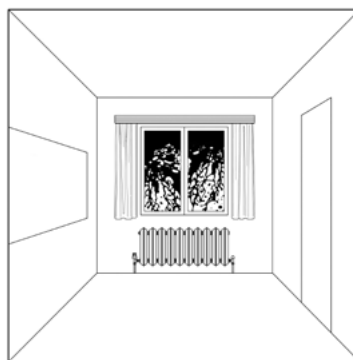
Rysunek 11. VR: scena nr 4

5. Czarne cząsteczki zagęszczają się, tworząc gęstą chmurę.



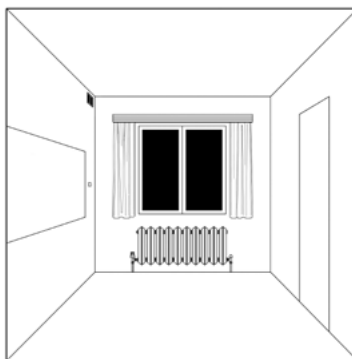
Rysunek 12. VR: scena nr 5

6. Ciemne drobinki o coraz większych rozmiarach zaczynają osiadać na szybie. Kropelki czarnej cieczy spływają po szkle.



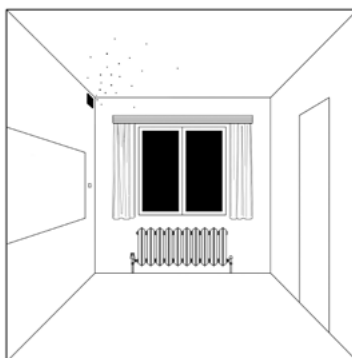
Rysunek 13. VR: scena nr 6

7. Czerń za oknem gęstnieje, całkowicie pochłaniając wcześniej widoczną biel. Szyby zaczynają drgać, na ich powierzchni pojawiają się drobne pęknięcia. Światło chwilowo gaśnie w wyniku awarii elektryczności. Na ścianie po lewej stronie od okna pojawiają się IKONKI (IKONKA nr 3, IKONKA nr 4), wskazujące włącznik wentylacji - kratka wywietrznika znajduje się powyżej (INTERAKCJA: uruchomienie wentylacji).



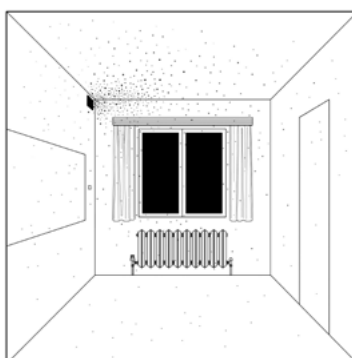
Rysunek 14. VR: scena nr 7

8. Czarne cząstki zaczynają przedostawać się przez kratkę do wnętrza pomieszczenia.



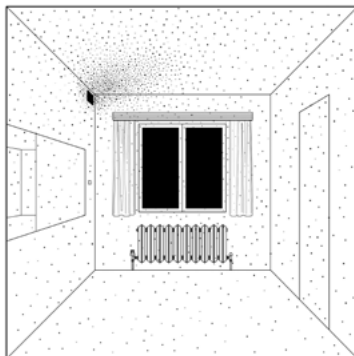
Rysunek 15. VR: scena nr 8

9. Gęste chmury cząsteczek wirują w powietrzu. Ruchy cząsteczek mają organiczny charakter, drobinki zachowują się w sposób zbliżony do mikroorganizmów.



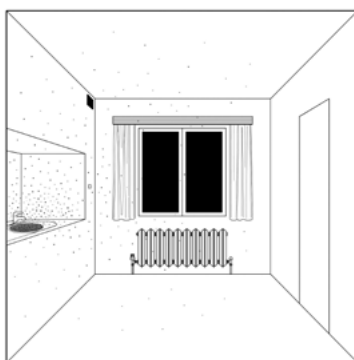
Rysunek 16. VR: scena nr 9

10. W prostokątnej wnęce po lewej stronie pojawia się IKONKA (IKONKA nr 5), wskazująca ruchomą ściankę i zachęcająca do jej odsunięcia (INTERAKCJA: odsunięcie ścianki). Za nią znajduje się mleczna szyba, która stopniowo staje się transparentna. Szyby nie można trwale odsunąć - stawia wyraźny opór i przy każdej próbie jej poruszenia wraca do wyjściowej pozycji. We wnęce widać coraz wyraźniej standardowy aneks kuchenny z białym blatem, płytą indukcyjną i zlewozmywakiem.



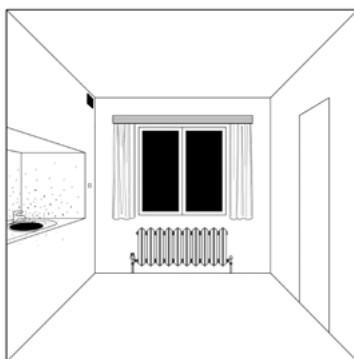
Rysunek 17. VR: scena nr 10

11. Nad zlewem pojawiają się IKONKI (IKONKA nr 6, IKONKA nr 7), zachęcające do wyjęcia (zdalnie, za pomocą manipulatorów) korka tkwiącego w odpływie (INTERAKCJA: wyjęcie korka). Czarne cząsteczki natychmiast reagują na ten ruch, kumulując się przy szybie aneksu kuchennego. Kolejna IKONKA (IKONKA nr 5) wskazuje na konieczność odsunięcia szyby (INTERAKCJA: częściowe odsunięcie szyby). Drobinki „zassane” przez wąską szczelinę powstałą po odsunięciu szyby wnikają w odpływ zlewu.



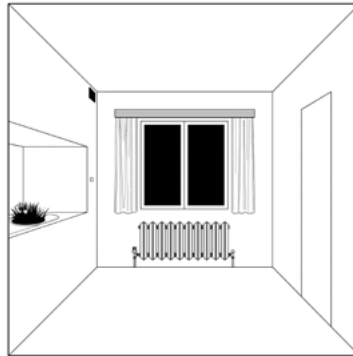
Rysunek 18. VR: scena nr 11

12. W momencie gdy wszystkie drobinki znajdą się we wnęce, pojawia się IKONKA (IKONKA nr 8), informująca o konieczności zasunięcia szyby (INTERAKCJA: zasunięcie szyby). Chmura czarnych cząsteczek jest w tej chwili uwięziona za szybą.



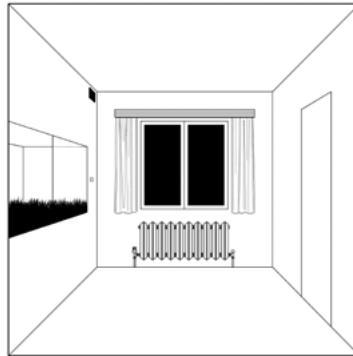
Rysunek 19. VR: scena nr 12

13. Wszystkie drobinki osiadają i skraplają się w odpływie zlewu, z którego zaczyna wydobywać się warstwa ciemnej pleśni, szybko przeradzająca się w bujną roślinność. Kropla czarnej cieczy wydobywa się z kranu.



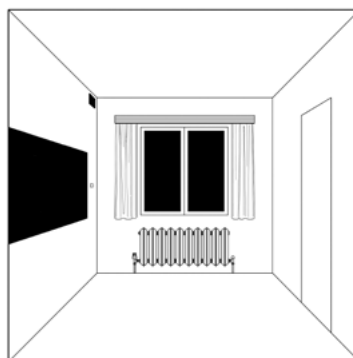
Rysunek 20. VR: scena nr 13

14. Czarny gąszcz rozrasta się w zlewie i na powierzchni blatu, wypełniając całą dostępną przestrzeń.



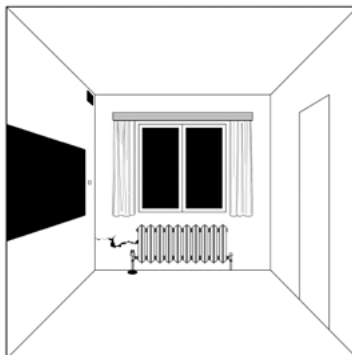
Rysunek 21. VR: scena nr 14

15. Czerni wypełnia już całą wnękę aneksu. Szyba zaczyna drgać, a na jej powierzchni pojawiają się drobne pęknięcia.



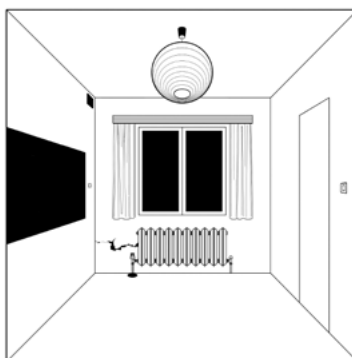
Rysunek 22. VR: scena nr 15

16. Na ścianie w pobliżu wnęki pojawiają się pęknięcia - ślady sprawiają wrażenie jakby odpowiadały linii wodociągów. Rysa biegnie aż do kaloryfera, z którego zaczyna sączyć się ciemna strużka. Pojawia się IKONKA (IKONKA nr 9), sugerująca zakręcenie zaworu kaloryfera (INTERAKCJA: zakręcenie kaloryfera). Wyciek zostaje powstrzymany.



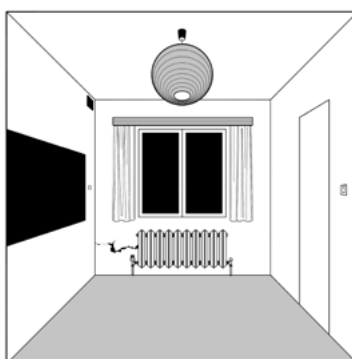
Rysunek 23. VR: scena nr 16

17. Następuje chwilowa awaria elektryczności. Gdy na krótko rozbłyska i gaśnie światło, po raz pierwszy widoczne staje się jego źródło: to zwisająca z sufitu lampa, otoczona białym kulistym abażurem. Pojawiają się IKONKI (IKONKA nr 10, IKONKA nr 11), wskazujące włącznik światła przy ramie drzwi wyjściowych (INTERAKCJA: włączenie światła).



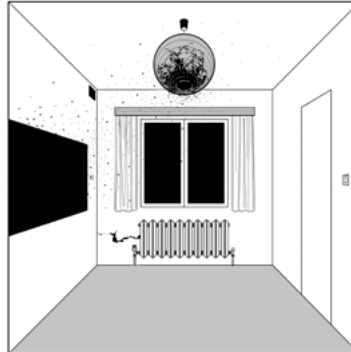
Rysunek 24. VR: scena nr 17

18. Po włączeniu światła na podłodze pojawia się miękki, jasnoszary dywan. Szyba aneksu kuchennego pęka, przepuszczając stłoczone za nią drobinki.



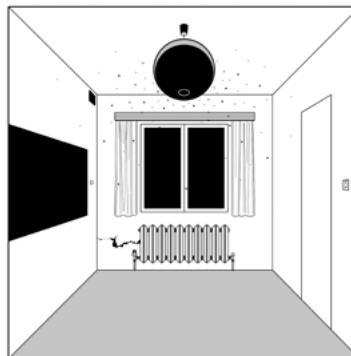
Rysunek 25. VR: scena nr 18

19. Uwolnione cząsteczki szybko przemieszczają się z wnęki aneksu kuchennego w kierunku lampy, osiadając i skraplając się we wnętrzu jej abażuru. Pojawia się IKONKA (IKONKA nr 12), zachęcająca do zdalnego zamknięcia klosza lampy w celu ponownego uwięzienia kotłujących się w nim czarnych drobinek (INTERAKCJA: zamknięcie klosza lampy).



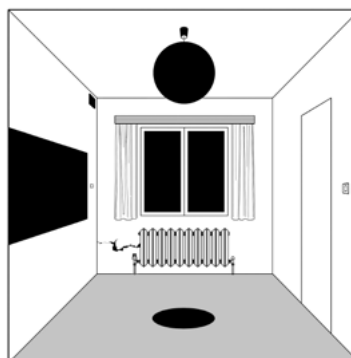
Rysunek 26. VR: scena nr 19

20. Abażur lampy zamyka się szczelnie, nasiąkając powoli gęstą czarną substancją.



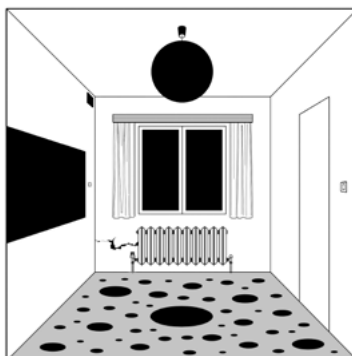
Rysunek 27. VR: scena nr 20

21. Na dywanie pod lampą pojawia się cień, przeradzający się w rosnącą ciemną plamę.



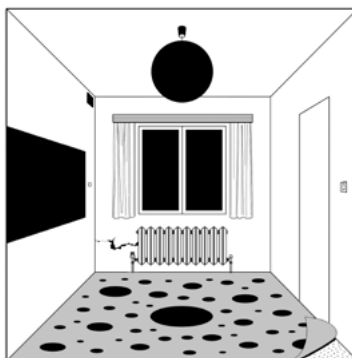
Rysunek 28. VR: scena nr 21

22. Stopniowo dywan nasiąka od spodu czarną cieczą.



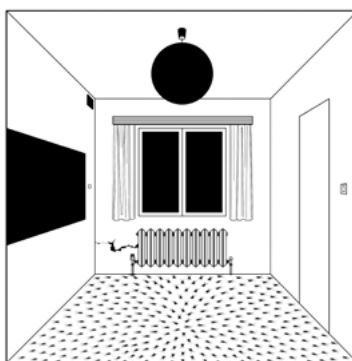
Rysunek 29. VR: scena nr 22

23. W narożniku dywanu pojawia się IKONKA (IKONKA nr 13), zachęcająca do podniesienia jego narożnika (INTERAKCJA: odsłonięcie narożnika dywanu).



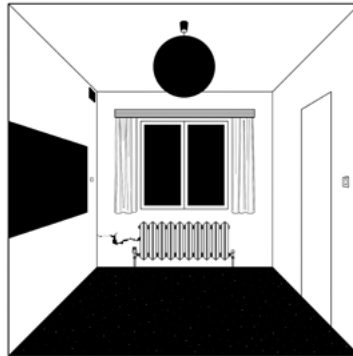
Rysunek 30. VR: scena nr 23

24. Na podłodze kłębią się czarne punkty, przypominające rój owadów.



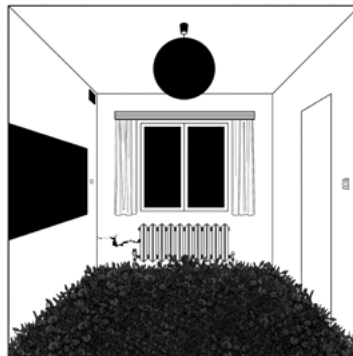
Rysunek 31. VR: scena nr 24

25. Rojowisko czarnych mikroorganizmów gęstnieje, a jego struktura zaczyna przypominać przesypującą się ziemię.



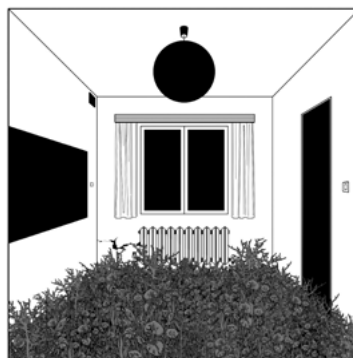
Rysunek 32. VR: scena nr 25

26. Na warstwie czarnej materii wyrasta ciemna roślinność, przypominająca wybujałą szarawą pleśń. W powietrzu unoszą się wydzielane przez nie ciemne pyłki.



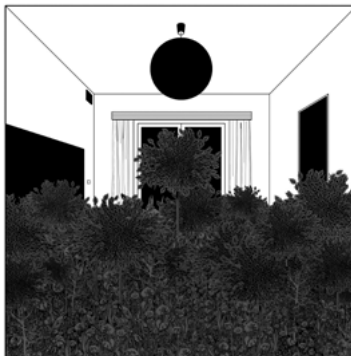
Rysunek 33. VR: scena nr 26

27. Przy drzwiach wyjściowych pojawia się IKONKA (IKONKA nr 14), zachęcająca do odsunięcia przykrywającej je płyty (INTERAKCJA: otwarcie drzwi wyjściowych). Za białą ścianką pojawia się czarna zwierciadlana tafla, w której użytkownik może zobaczyć niewyraźny zarys ludzkiej postaci - być może jest to jego własne, zdeformowane odbicie.



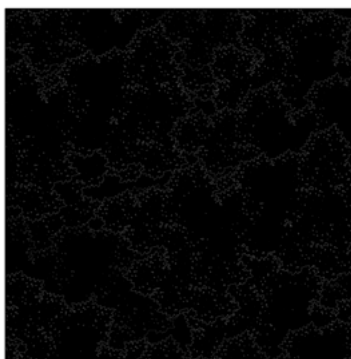
Rysunek 34. VR: scena nr 27

28. Roślinność formuje się w trawy, potem krzewy i małe drzewa, tworząc nieprzejrzysty gąszcz, utrudniający dostęp do drzwi.



Rysunek 35. VR: scena nr 28

29. Pomieszczenie jest już do samego sufitu wypełnione gęstniejącą czarnoszarą roślinnością. W plątaniu nawarstwiających się liści, kłaczy i pyłków zanikają ostatnie prześwity bieli.



Rysunek 36. VR: scena nr 29

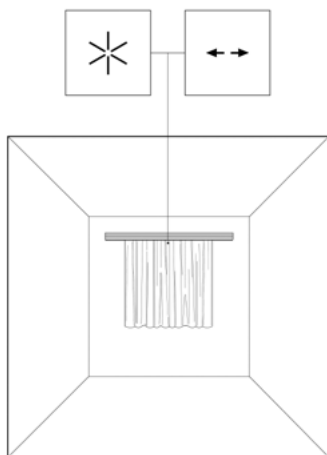
30. Światło ostatecznie gaśnie, wyznaczając naturalny koniec gry.



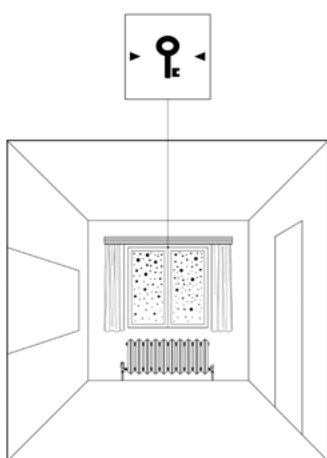
Rysunek 37. VR: scena nr 30

2.2. Aplikacja VR - plan interakcji: komunikacja wizualna

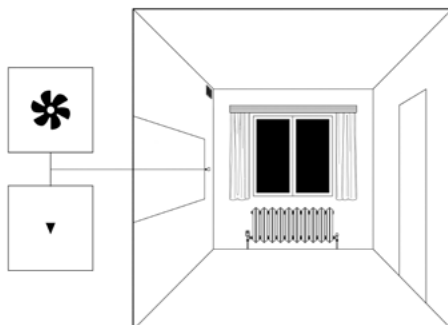
Aby umożliwić użytkownikowi aplikacji sprawną nawigację w przestrzeni rozszerzonej/ wirtualnej rzeczywistości, kolejnym interakcjom tworzącym logiczną sekwencję zdarzeń przyporządkowany został zestaw ikon, komunikujących poszczególne czynności. Zgodnie z konwencją estetyczną przyjętą w ramach projektu „Stilleben” warstwa graficzna wszystkich elementów interfejsu podporządkowana została zasadom minimalizmu oraz prostym strukturom geometrycznym. Zamiast poleceń w formie tekstowej, wskazówki dla uczestnika „gry” sformułowane zostały w uniwersalnym języku komunikacji wizualnej.



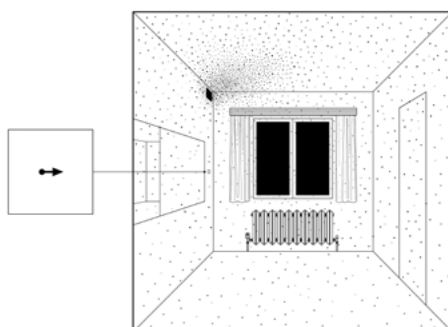
Rysunek 38. VR: Plan interakcji. Umieszczenie IKONKI nr 1 oraz IKONKI nr 2



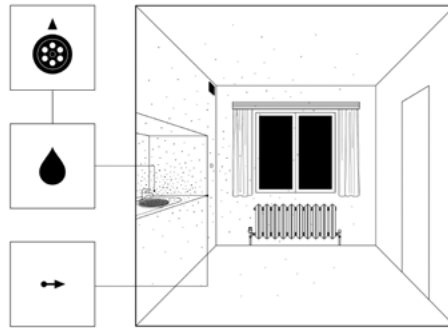
Rysunek 39. VR: Plan interakcji. Umieszczenie IKONKI nr 3



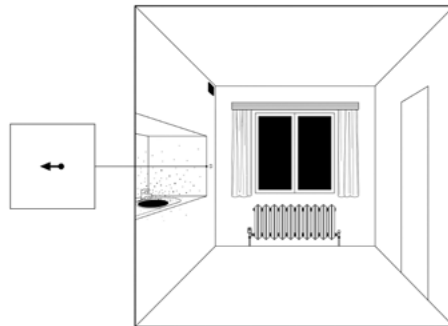
Rysunek 40. VR: Plan interakcji. Umieszczenie IKONKI nr 4 oraz IKONKI nr 5



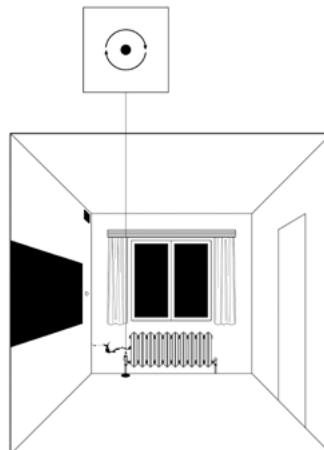
Rysunek 41. VR: Plan interakcji. Umieszczenie IKONKI nr 6



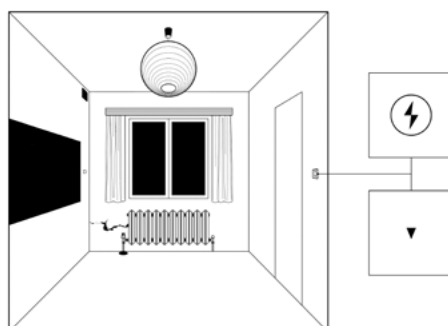
Rysunek 42. VR: Plan interakcji. Umieszczenie IKONKI nr 7, IKONKI nr 8 oraz IKONKI nr 6



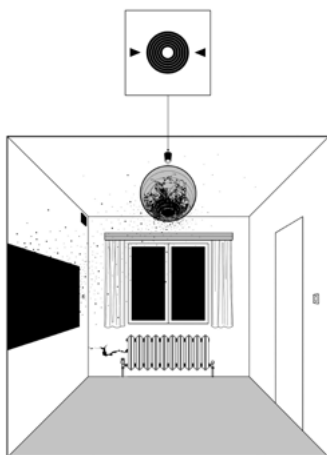
Rysunek 43. VR: Plan interakcji. Umieszczenie IKONKI nr 9



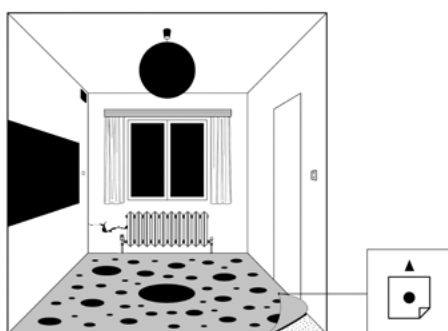
Rysunek 44. VR: Plan interakcji. Umieszczenie IKONKI nr 10



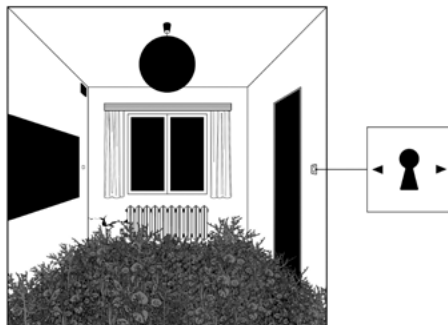
Rysunek 45. VR: Plan interakcji. Umieszczenie IKONKI nr 11 oraz IKONKI nr 5



Rysunek 46. VR: Plan interakcji. Umieszczenie IKONKI nr 12

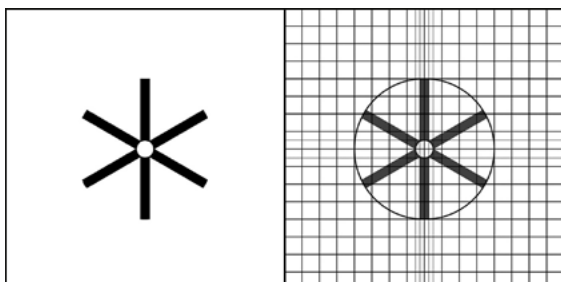


Rysunek 47. VR: Plan interakcji. Umieszczenie IKONKI nr 13

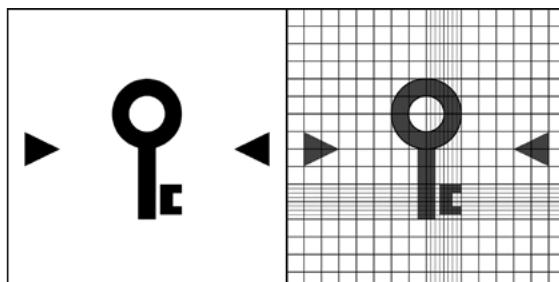


Rysunek 48. VR: Plan interakcji. Umieszczenie IKONKI nr 14

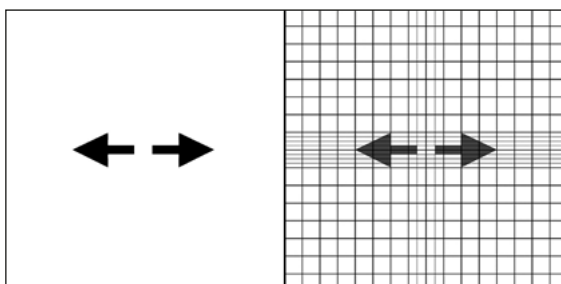
Jedenaście interakcji opisanych zostało za pomocą zestawu czternastu ikon, przygotowanych na podstawie siatki złożonej z kwadratowych modułów (16 x 16 pól). Poszczególne grafiki skonstruowano, opierając się na elementarnych figurach geometrycznych: kole, kwadracie i trójkącie. Uprozczone wizerunki obiektów występujących w przestrzeni wirtualnej uzupełnione zostały strzałkami, wskazującymi kierunek oczekiwanego ruchu (otwarcie, zamknięcie, obrót, itp.) oraz oznaczeniami odnoszącymi się do różnych rodzajów energii: przepływu powietrza, wody i elektryczności.



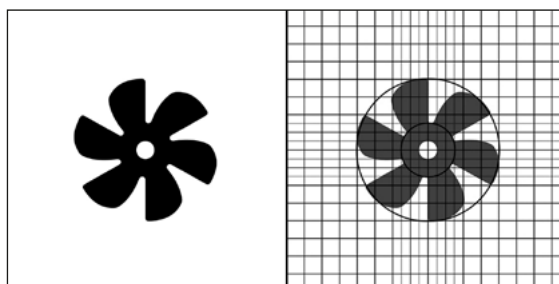
Rysunek 49. VR: IKONKA nr 1. Metoda konstruowania znaku w oparciu o siatkę modułową



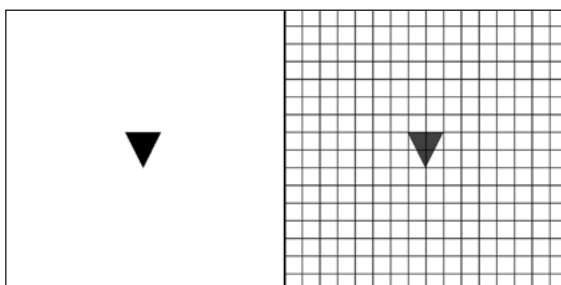
Rysunek 51. VR: Metoda konstruowania znaku w oparciu o siatkę modułową



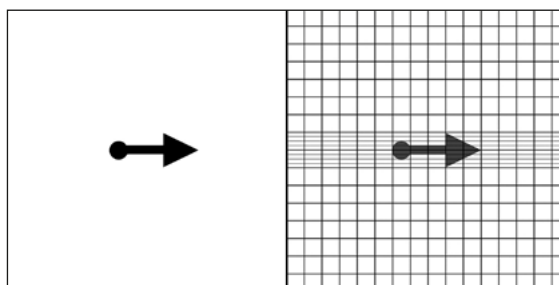
Rysunek 50. VR: IKONKA nr 2. Metoda konstruowania znaku w oparciu o siatkę modułową



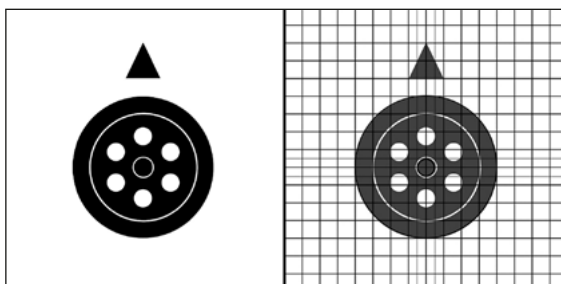
Rysunek 52. VR: IKONKA nr 4. Metoda konstruowania znaku w oparciu o siatkę modułową



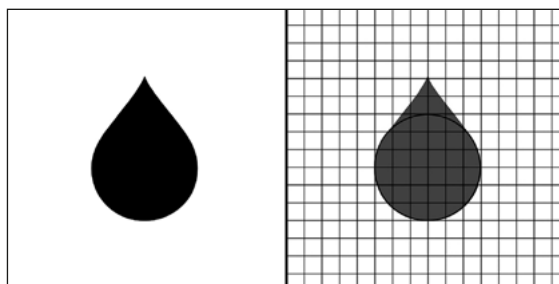
Rysunek 53. VR: IKONKA nr 5. Metoda konstruowania znaku w oparciu o siatkę modułową



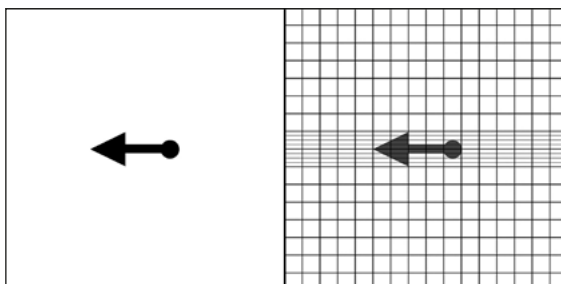
Rysunek 54. VR: IKONKA nr 6. Metoda konstruowania znaku w oparciu o siatkę modułową



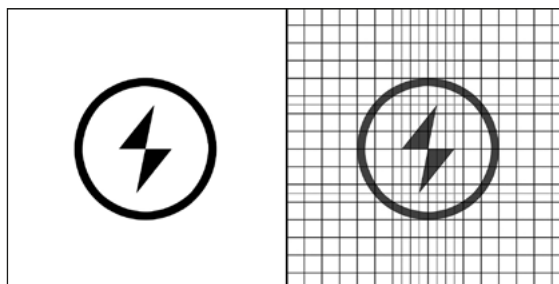
Rysunek 55. VR: IKONKA nr 7. Metoda konstruowania znaku w oparciu o siatkę modułową



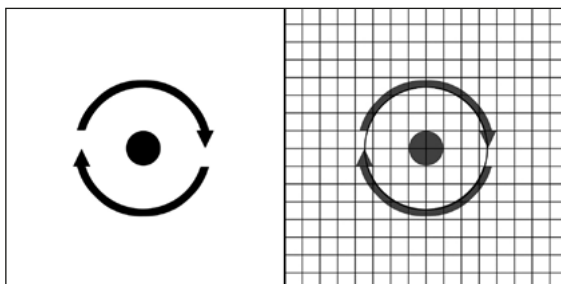
Rysunek 56. VR: IKONKA nr 8. Metoda konstruowania znaku w oparciu o siatkę modułową



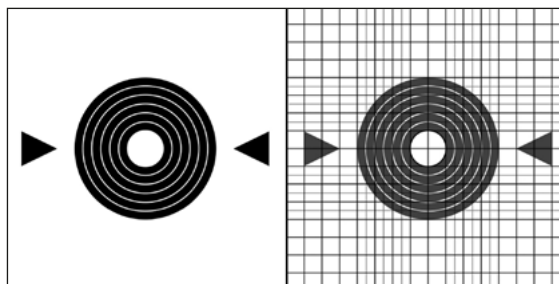
Rysunek 57. VR: IKONKA nr 9. Metoda konstruowania znaku w oparciu o siatkę modułową



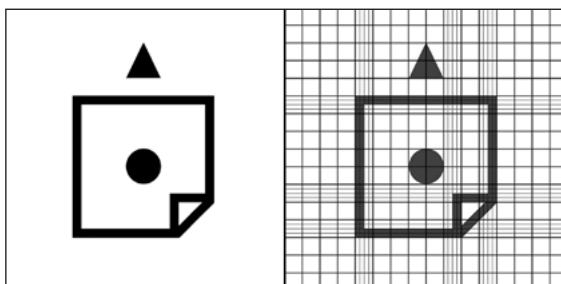
Rysunek 59. VR: IKONKA nr 11. Metoda konstruowania znaku w oparciu o siatkę modułową



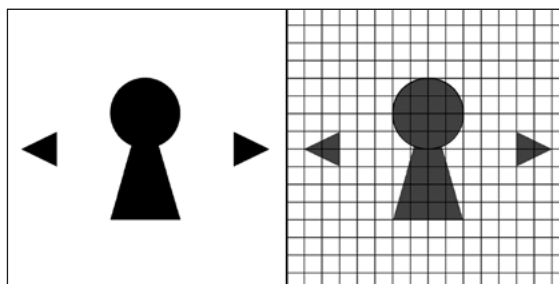
Rysunek 58. VR: IKONKA nr 10. Metoda konstruowania znaku w oparciu o siatkę modułową



Rysunek 60. VR: IKONKA nr 12. Metoda konstruowania znaku w oparciu o siatkę modułową



Rysunek 61. VR: IKONKA nr 13. Metoda konstruowania znaku w oparciu o siatkę modułową



Rysunek 62. VR: IKONKA nr 14. Metoda konstruowania znaku w oparciu o siatkę modułową

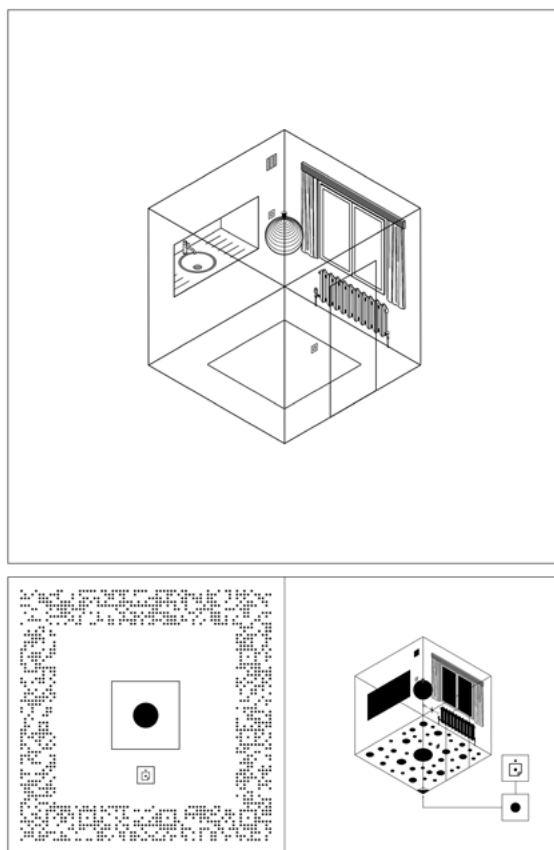
2.3. Aplikacja AR - scenorys z omówieniem

Aby rozszerzyć możliwości upowszechnienia aplikacji, pod względem strukturalnym projekt „Stilleben” został podzielony na dwie autonomiczne części. Pierwszą z nich stanowi omówiona wcześniej stacjonarna wirtualna wystawa VR/AR, której prezentacja możliwa jest w sterylnej przestrzeni na terenie laboratorium lub w specjalnie zaadaptowanej przestrzeni galeryjnej. Drugą odsłoną projektu stanowi „mobilną” adaptację zrealizowanej wcześniej wystawy w postaci narracyjnego artbooka, zawierającego zbiór wizualnych kodów - symboli i diagramów - uzupełnionych o wirtualne trójwymiarowe obiekty i animacje, tworzące interaktywną strukturę „przypisów” do syntetycznego modelu przestrzeni mieszkalnej. W tym wariancie hełm wirtualnej rzeczywistości zastąpiony został telefonem komórkowym umożliwiającym obsługę rozszerzonej rzeczywistości (AR).

System minimalistycznych oznaczeń graficznych obecnych w wersji książkowej stanowi w dużej mierze uproszczony wariant rozwiązań zastosowanych również w pierwszej wersji projektu. W związku z tym książka może być traktowana także jako dodatkowy komponent uzupełniający wcześniejszą aplikację, swoista instrukcja obsługi, ułatwiająca zrozumienie obowiązujących w obu wariantach „zasad gry”. Istotną cechą projektu jest jego dostępność, związana z możliwością szerokiej dystrybucji w formie darmowej aplikacji AR oraz publikacji w formacie PDF przeznaczonej do samodzielnego wydruku.

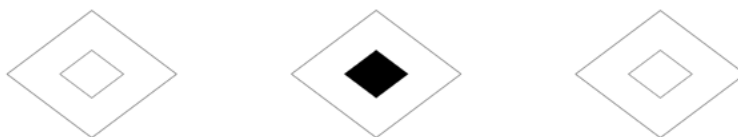
Struktura wizualnej opowieści przedstawionej w artbooku „Stilleben” odzwierciedla w uproszczonej formie schematy interakcji obecne w obrębie aplikacji stacjonarnej, wykorzystując również przygotowany na jej potrzeby system wewnętrznej identyfikacji wizualnej projektu - zestaw ikonki ułatwiających użytkownikowi eksplorację świata przedstawionego. Ten szczególny rodzaj graficznego języka zastępuje tradycyjne formy tekstu, niemal całkowicie - poza krótkim wprowadzeniem - nieobecne w książce. Narracyjna sekwencja obrazów zaprezentowana w książce stanowi swoisty „storyboard” historii opowiedzianej wcześniej w ramach „wystawy” VR/AR.

64-stronicowa graficzna historia podzielona została na trzydzieści dwie „jednostki” - dwustronicowe rozkładówki, ujednoczone pod względem kompozycyjnym, zawierające każdorazowo stały zestaw elementów. Prawa stronica każdej z rozkładówek przedstawia izometryczny rzut „miejsca akcji” - pomieszczenia mieszkalnego, wpisanego w moduł sześciokąta, w przestrzeni dwuwymiarowej odwzorowanego jako regularny heksagon. [ilustracja 00A] Każdy z architektonicznych „diagramów” zawiera odnośniki do konkretnych detali, powiązanych z kolejnymi etapami eksploracyjnej aktywności użytkownika. Ikonki wykorzystane w odnośnikach odsyłają zawsze bezpośrednio do lewej stronicy rozkładówki, przedstawiającej - w geometrycznym uproszczeniu - wybrany element, rozbudowany w rzeczywistości rozszerzonej o trójwymiarowe symulacje, często wzbogacone również prostymi animacjami. [ilustracja 00B] Niewielkie ikonki znajdujące się w dolnej części niektórych z tych stronic stanowią proste „butony”, których „naciśnięcie” palcem (czyli *de facto* ich zasłonięcie, rejestrowane przez aplikację) wywołuje interakcję wpływającą na modyfikację wirtualnej „nadbudowy” grafiki. Ponowne odsłonięcie każdej z ikonki powoduje „cofnięcie” wprowadzonych zmian i powrót do stanu wyjściowego. Otaczające grafikę wzory złożone z czarnych punktów spełniają funkcjonalną rolę „zmaczników” pozwalających na odwzorowanie za sprawą aplikacji właściwych współrzędnych przestrzennych w obrębie „rzeczywistości rozszerzonej”.



Rysunek 63. AR: ilustracje 00A, 00B

Ilustracje 01A, 01B i 01C przedstawiają biały kwadrat obwiedziony czarną ramką, który w rozszerzeniu AR w krótkich nieregularnych odcinkach czasu wypełnia się czernią. Wywołany w ten sposób efekt migotania wywołuje skojarzenie z awarią instalacji elektrycznej.



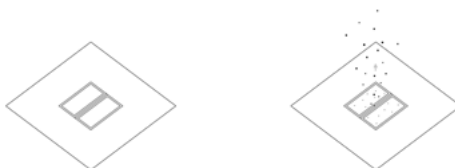
Rysunek 64. AR: ilustracje 01A, 01B, 01C

Ilustracje 02A, 02B i 02C przedstawiają uproszczony obraz okna. W rozszerzeniu AR obraz ten zostaje uzupełniony o karnisz i dwie kotary, w pozycji wyjściowej zakrywające szybę. Po naciśnięciu dolnej ikonki (symbol dwóch strzałek, sugerujących ruch skierowany na zewnątrz), zasłony zostają rozsunięte.



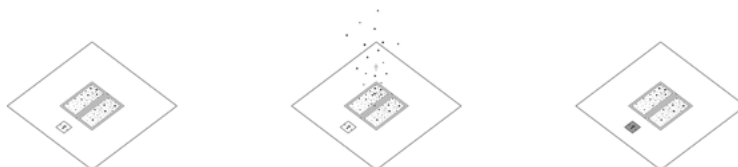
Rysunek 65. AR: ilustracje 02A, 02B, 02C

Ilustracje 03A i 03B przedstawiają analogiczny do wcześniejszego obraz okna, tym razem lekko uchylonego. W rozszerzeniu AR rama okienna zostaje wypełniona poruszającymi się „po drugiej stronie” czarnymi cząsteczkami. Niektóre z nich unoszą się, wędrując przez szczelinę między okiennicami.



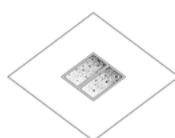
Rysunek 66. AR: ilustracje 03A, 03B

Ilustracje 04A, 04B i 04C także przedstawiają obraz uchylonego okna, ale uzupełniony o osiadłe na szybach czarne drobinki - cząsteczki „za oknem” w rozszerzeniu AR zostają wprowadzone w ruch, a niektóre z nich unoszą się, wędrując przez szczelinę między okiennicami. Po naciśnięciu dolnej ikonki (symbol kluczyka i strzałek sugerujących ruch skierowany do wewnątrz), „wyciek” cząsteczek w górę zostaje wstrzymany.



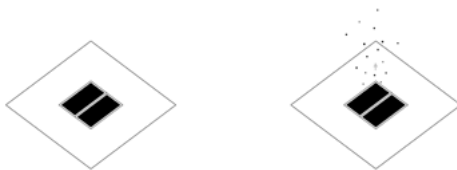
Rysunek 67. AR: ilustracje 04A, 04B, 04C

Ilustracja 05A przedstawia analogiczny do wcześniejszych obraz okna, na którego powierzchni osiadły czarne drobinki - tym razem większe i rozmieszczone gęściej niż poprzednio. W rozszerzeniu AR cząsteczki „za oknem” zostają wprowadzone w dynamiczny ruch.



Rysunek 68. AR: ilustracja 05A

Ilustracje 06A i 06B przedstawiają obraz okna, pokrytego warstwą nieprzejrzystej czerni. Pomimo iż okno jest zamknięte, w rozszerzeniu AR pojedyncze cząsteczki unoszą się w górę.



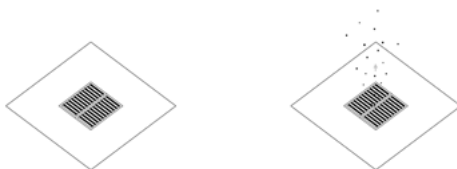
Rysunek 69. AR: ilustracje 06A, 06B

Ilustracje 07A i 07B przedstawiają uproszczony obraz wentylatora, który po naciśnięciu dolnej ikonki (symbol wiatraczka) w rozszerzeniu AR zaczyna się obracać.



Rysunek 70. AR: ilustracje 07A, 07B

Ilustracje 08A i 08B przedstawiają kratkę wentylatora, przez którą w rozszerzeniu AR przenikają unoszące się pojedyncze czarne cząsteczki.



Rysunek 71. AR: ilustracje 08A, 08B

Ilustracje 09A, 09B i 09C przedstawiają syntetyczny widok kuchennego zlewu w rzucie górnym. W rozszerzeniu AR nad jego powierzchnią zostaje nadbudowana transparentna gablota (na planie pomieszczenia: aneks kuchenny w formie wnęki wbudowanej w ścianę). Po naciśnięciu dolnej ikonki (symbol strzałki, sugerującej ruch w lewo) przednia płyta prostopadłościennego „akwarium” zostaje odsunięta.



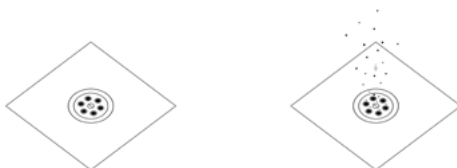
Rysunek 72. AR: ilustracje 09A, 09B, 09C

Ilustracje 10A, 10B i 10C przedstawiają uproszczony obraz kratki odpływowej zlewu. W rozszerzeniu AR czarny okrąg okazuje się przestrzennym krążkiem - korkiem zamykającym odpływ. Po naciśnięciu dolnej ikonki (syntetyczne przedstawienie okrągłej kratki ze strzałką sugerującą ruch w górę) korek unosi się ponad powierzchnię strony, odsłaniając ujście odpływu.



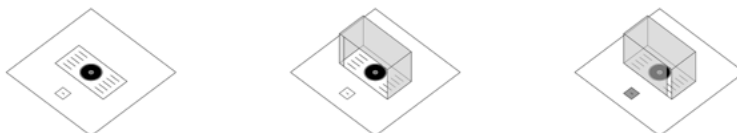
Rysunek 73. AR: ilustracje 10A, 10B, 10C

Ilustracje 11A i 11B przedstawiają ponownie kratkę odpływową. W rozszerzeniu AR dostrzegalne stają się zawieszone w powietrzu czarne cząsteczki, „zasysane” z otoczenia przez odpływ.



Rysunek 74. AR: ilustracje 11A, 11B

Ilustracje 12A, 12B i 12C przedstawiają syntetyczny widok kuchennego zlewu w rzucie górnym - tym razem wnętrze zlewu wypełnione jest czernią. W rozszerzeniu AR nad jego powierzchnią zostaje nadbudowana transparentna gablota. Po naciśnięciu dolnej ikonki (symbol strzałki, sugerującej ruch w prawo) otwarta przednia płyta prostopadłościennego „akwarium” zostaje zasunięta.



Rysunek 75. AR: ilustracje 12A, 12B, 12C

Ilustracje 13A i 13B przedstawiają uproszczony obraz dna zlewu wraz z odpływem. Kontur zaczernionej powierzchni jest organicznie poszarpany. W rozszerzeniu AR ciemna plama pokrywa się cienką warstwą czarnej roślinności.



Rysunek 76. AR: ilustracje 13A, 13B

Ilustracje 14A i 14B nawiązują do poprzedniego obrazu, odznaczają się jednak bardziej syntetycznym charakterem. Poszarpany kontur czarnej plamy rozrasta się promieniście we wszystkich kierunkach. W rozszerzeniu AR z ciemnego pola „wyrasta” duża kępa roślinności.



Rysunek 77. AR: ilustracje 14A, 14B

Ilustracje 15A i 15B przedstawiają zaczerniony prostokąt którego kontur jest organicznie poszarpany - to płyta kuchennego zlewu, zarastająca dziką roślinnością. W rozszerzeniu AR zamknięta transparentna gablota pokrywająca zlew wypełnia się czarną trawą.



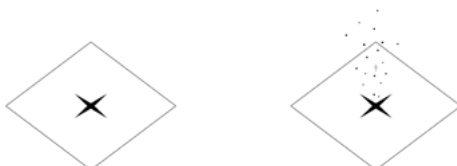
Rysunek 78. AR: ilustracje 15A, 15B

Ilustracje 16A i 16B nawiązują do poprzedniego obrazu, przedstawiając - w geometrycznym uproszczeniu - prostokątny kształt kuchennej płyty. W rozszerzeniu AR czarny czworobok staje się ciemnym prostopadłościem o proporcjach widocznej wcześniej gabloty.



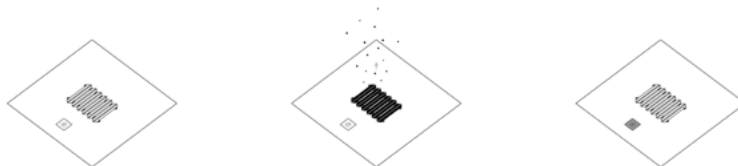
Rysunek 79. AR: ilustracje 16A, 16B

Ilustracje 17A i 17B przedstawiają czarną formę o kształcie zbliżonym do krzyżyka - to syntetycznie zobrazowane pęknięcie, widoczne na ścianie pomieszczenia. W rozszerzeniu AR ze szczeliny unoszą się czarne cząsteczki.



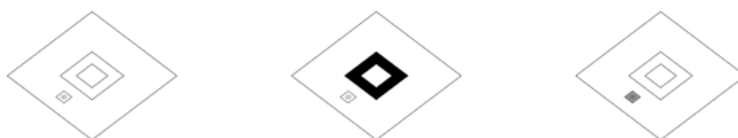
Rysunek 80. AR: ilustracje 17A, 17B

Ilustracje 18A, 18B i 18C przedstawiają uproszczony obraz kaloryfera. W rozszerzeniu AR jest on „zainfekowany” czarną substancją, której cząsteczki unoszą się ku górze. Po naciśnięciu dolnej ikonki (syntetyczny symbol pokrętła ze strzałkami sugerującymi ruch obrotowy) „wyciek” zostaje zatrzymany.



Rysunek 81. AR: ilustracje 18A, 18B, 18C

Ilustracje 19A, 19B i 19C przedstawiają syntetyczny obraz ściennego wyłącznika światła. W rozszerzeniu AR jego białe wypełnienie w nieregularnych odstępach czasu ustępuje czerni, co wywołuje efekt „migotania”, wywołanego awarią elektryczności. „Migotanie” ustaje po naciśnięciu ikonki (symbol błyskawicy).



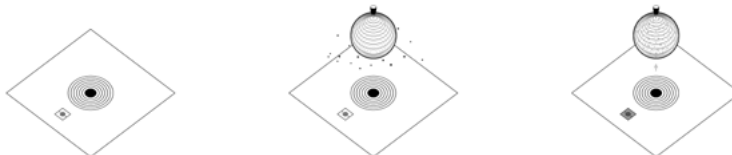
Rysunek 82. AR: ilustracje 19A, 19B, 19C

Ilustracje 20A i 20B przedstawiają obraz złożony z koncentrycznych okręgów - to widok kulistego abażuru, oglądanego z oddolnej perspektywy. W rozszerzeniu AR nad grafiką unosi się przestrzenna symulacja sferycznej lampy.



Rysunek 83. AR: ilustracje 20A, 20B

Ilustracje 21A, 21B i 21C przedstawiają - analogiczny do wcześniejszego - widok abażuru, którego wlot jest jednak wypełniony czernią. W rozszerzeniu AR nad grafiką unosi się przestrzenna symulacja sferycznej lampy - pod nią wirują czarne cząsteczki. Po naciśnięciu dolnej ikonki (uproszczone przedstawienie abażuru ze strzałkami sugerującymi ruch dośrodkowy) wylot klosza lampy zamyka się, „zasysając” wirujące cząsteczki do środka.



Rysunek 84. AR: ilustracje 21A, 21B, 21C

Ilustracje 22A i 22B przedstawiają - analogiczny do wcześniejszego - widok abażuru, którego środkowa część jest całkowicie wypełniona czernią. W rozszerzeniu AR nad grafiką unosi się przestrzenna symulacja sferycznej lampy, której klosz do połowy wypełnia czarna materia.



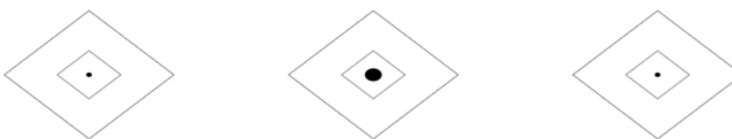
Rysunek 85. AR: ilustracje 22A, 22B

Ilustracje 23A i 23B przedstawiają - analogiczny do wcześniejszego - widok abażuru, całkowicie wypełnionego czernią. W rozszerzeniu AR nad grafiką unosi się przestrzenna symulacja sferycznej lampy, której klosz po wypełnieniu przez ciemne cząsteczki stał się czarną kulą.



Rysunek 86. AR: ilustracje 23A, 23B

Ilustracje 24A, 24B i 24C przedstawiają biały kwadrat obwiedziony czarną linią - to syntetyczne wyobrażenie dywanu zakrywającego podłogę pomieszczenia. Na środku pola znajduje się okrągła czarna plama - w rozszerzeniu AR plama „pulsuje”, rytmicznie powiększając się zmniejszając.



Rysunek 87. AR: ilustracje 24A, 24B, 24C

Ilustracje 25A i 25B przedstawiają - analogiczny do poprzedniego - uproszczony obraz jasnego dywanu z widoczną na środku dużą czarną plamą. W rozszerzeniu AR, po naciśnięciu dolnej ikonki (uproszczony obraz dywanu z zagiętym narożnikiem i strzałką sugerująca ruch w górę), rąbek dywanu zostaje uniesiony w górę, odsłaniając fragment czarnego pola, ukrytego pod spodem.



Rysunek 88. AR: ilustracje 25A, 25B

Ilustracje 26A i 26B przedstawiają czarne pole pod dywanem, wypełnione jasnymi plamami. W rozszerzeniu AR gęszcz jasnych plamek porusza się dynamicznie, przypominając rój żywych organizmów.



Rysunek 89. AR: ilustracje 26A, 26B

Ilustracje 27A i 27B przedstawiają czarne pole pod dywanem - ciemny kwadrat o organicznie poszarpanym konturze. W rozszerzeniu AR, pole szczelnie wypełnia się czarną trawą.



Rysunek 90. AR: ilustracje 27A, 27B

Ilustracje 28A i 28B przedstawiają koncentryczne zgrupowanie czarnych plamek, symbolizujących organiczną podściółkę, zgromadzoną pod dywanem. W rozszerzeniu AR na środku wyrasta niewielkie drzewko.



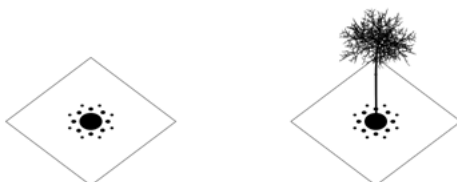
Rysunek 91. AR: ilustracje 28A, 28B

Ilustracje 29A i 29B przedstawiają koncentryczne - bardziej rozległe od poprzedniego - zgrupowanie czarnych plamek. W rozszerzeniu AR na środku wyrasta drzewko, w kolejnej fazie rozwoju.



Rysunek 92. AR: ilustracje 29A, 29B

Ilustracje 30A i 30B przedstawiają koncentryczne - bardziej rozległe od poprzedniego - zgrupowanie czarnych plamek. W rozszerzeniu AR drzewko wyrastające na środku osiąga maksymalny rozmiar.



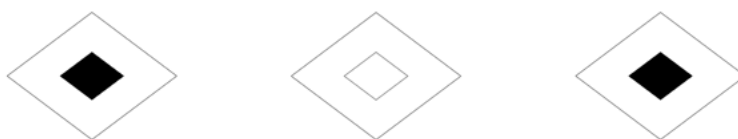
Rysunek 93. AR: ilustracje 30A, 30B

Ilustracje 31A, 31B i 31C przedstawiają biały prostokąt obwiedziony czarną linią - to syntetyczne przedstawienie drzwi wyjściowych, które ujawniły się na planie pomieszczenia. W rozszerzeniu AR, po naciśnięciu dolnej ikonki (dziurka od klucza i strzałki sugerujące ruch odśrodkowy) drzwi rozsuwają się, ujawniając ciemną przestrzeń ukrytą za nimi.



Rysunek 94. AR: ilustracje 31A, 31B, 31C

Ilustracje 32A, 32B i 32C przedstawiają czarny kwadrat. W rozszerzeniu AR ciemne pole w krótkich, nieregularnych odcinkach czasu wypełnia się bielą. Wywołany w ten sposób efekt migotania wywołuje skojarzenie z awarią instalacji elektrycznej.



Rysunek 95. AR: ilustracje 32A, 32B, 32C

3. Implementacja

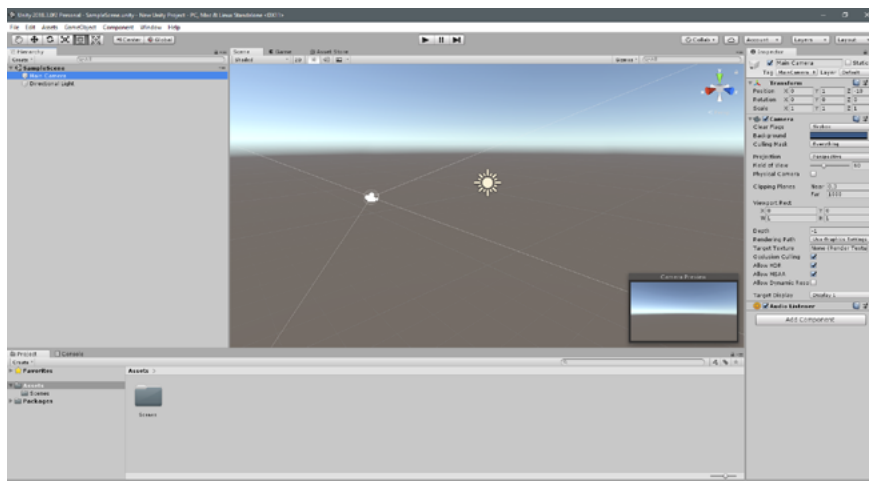
3.1. Aplikacja VR - Unity, VRTK, SteamVR, HTC Vive

Poniższy opis przebiegu implementacji projektu „Stilleben” do wirtualnej rzeczywistości składa się z siedmiu części. Pierwsze cztery pokrótce opisują użyte technologie, kolejne zaś proces tworzenia scen, zaimportowane obiekty oraz stworzone skrypty (najczęściej do obsługi animacji i interakcji).

3.1.1. Unity Engine

Unity to zintegrowane środowisko do tworzenia interaktywnych aplikacji - przede wszystkim gier komputerowych i symulacji 3D. Po raz pierwszy ujrzało światło dzienne w czasie wydarzenia Apple Inc.'s Worldwide Developers Conference w 2005. Początkowo silnik Unity działał jedynie na systemie OS-X i nadawał się przede wszystkim do tworzenia gier 3D. Współcześnie jest to narzędzie wieloplatformowe (aktualnie wspiera imponujące 27 platform) i posiada natywną obsługę środowisk 3D, 2D, VR, AR i innych. Unity jest oprogramowaniem zamkniętym i komercyjnym (ale jego licencja pozwala na bezpłatne użycie w ograniczonym zakresie). Sam silnik jako runtime używa C++ lecz do skryptuje się go za pomocą C# (ze specjalnym Unity Scripting API¹). W poprzednich wersjach do skryptowania można było używać języków JavaScript i pythonopodobnego Boo.

¹ API (*Application Programming Interface, Interferjs programowania aplikacji*) - jego zadaniem jest dostarczenie pewnych, charakterystycznych dla danego środowiska podprogramów, struktur danych, klas, obiektów i protokołów, które umożliwiają pełne wykorzystanie np. dobrodziejstw silnika lub danej platformy



Rysunek 96. Standardowy widok nowego projektu w unity v2018.3.0f2

3.1.2. Virtual Reality Toolkit

VRTK jest niezależnie tworzonym, dostępnym publicznie na licencji MIT² narzędziem (kolekcją skryptów) umożliwiającym pracę z różnymi platformami VR w silniku Unity. Aktualnie wspierane platformy to SteamVR, Oculus, Ximmerse oraz Daydream. Virtual Reality Toolkit wspomaga obsługę takich kwestii jak: poruszanie się w przestrzeni wirtualnej, interakcje (dotykanie, chwytanie, używanie obiektów), korzystanie z interfejsów (Unity UI), fizyka ciała w VR, dwu i trójwymiarowe guziki, dźwignie, drzwi wskaźniki, szuflady i wiele innych³. VRTK działa z Unity w wersji 5 i wyższych, wymaga manualnej instalacji pakietu i spięcia assetów z dostępnym w natywnym dla silnika Asset Store SteamVR (więcej o SteamVR poniżej).

3.1.3. Steam Virtual Reality

SteamVR to platforma obsługi wirtualnej rzeczywistości stworzona przez firmę Valve. Jest ona dostępna dla różnych urządzeń jednak dedykowana jest dla HTC Vive, który to headset powstał w wyniku współpracy Valve z HTC. Aplikacja ta pozwala - na poziomie użytkownika - na konfigurację i kalibrację zestawu do VR oraz umożliwia używanie go z oprogramowaniem np. z biblioteki Steam lub w aplikacji Vive. Na poziomie silnika (np. Unity) SteamVR pozwala na obsługę poszczególnych elementów urządzenia za pomocą skryptów oraz zapewnia wsparcie dla specyficznego rodzaju wyświetlania (3D-360) oraz akcesoriów (trackery etc.).

3.1.4. HTC Vive

HTC Vive to jeden z wiodących na rynku systemów rzeczywistości wirtualnych. Zasadniczo składa się z hełmu w którego charakterystycznych „goglach” wyświetlany jest obraz dający użytkownikowi złudzenie głębi. Hełm posiada funkcję trackingu tzn. śledzenia ruchów głowy i mapowania ich na położenie kamery w aplikacji. Tracking realizowany jest za pomocą akcelerometru oraz (co najmniej) dwóch specjalistycznych kamer bezprzewodowych (stacje bazowe, lighthouses). Klasyczny HTC Vive posiada dwa kontrolery (również trackowane w przestrzeni). Aktualnie na rynku pojawiły się wersję headsetu ze zwiększoną rozdzielczością, bezprzewodowe (specjalny moduł i karta), z wbudowanym okulary. Projekt „Stilleben” powstał przy użyciu klasycznej, konsumenckiej wersji HTC Vive.

Vive (Viewport) to również nazwa zestawu oprogramowania do obsługi platformy (sterowniki, środowisko startowe, sklep/biblioteka aplikacji).

2 patrz: <https://opensource.org/licenses/MIT>

3 za: <https://vrtoolkit.readme.io>

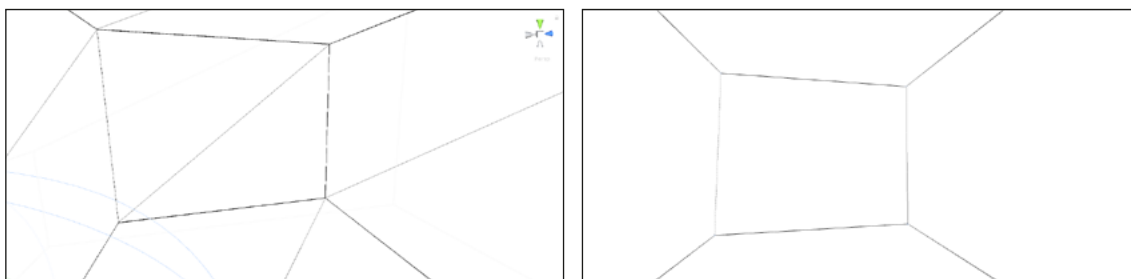
Specyfikacja techniczna headsetu HTC Vive:⁴

- Ekran: Dual AMOLED 3.6" diagonal
- Rozdzielczość: 1080 x 1200 pikseli na oko (2160 x 1200 pikseli razem)
- Częstotliwość odświeżania: 90 Hz
- Pole widzenia: 110 stopni
- Sensory: SteamVR Tracking, G-sensor, gyroscope, proximity
- Połączenia: HDMI, USB 2.0, stereo 3.5 mm headphone jack, Power, Bluetooth
- Wejście: Zintegrowany mikrofon

3.1.5. Sceny

Jak to zostało opisane w senorysie - instalacja VR „Stilleben” składa się z 30 scen w trzech zasadniczych sekwencjach: 1) Pomieszczenie z wyposażeniem (okno, zasłony, drzwi, wentylator, zlew itd.); 2) „Inwazja” organicznych obiektów (w aplikacji reprezentowanych najczęściej jako systemy cząsteczek); 3) „Zarastanie” pomieszczenia - aż do zupełnej czerni - roślinnością.

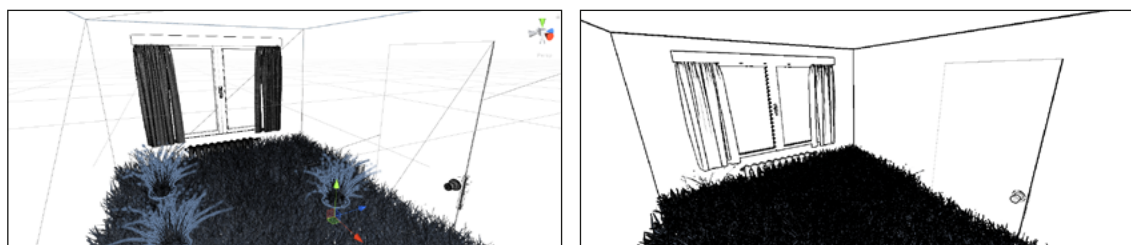
Poniżej zaprezentowano wybrane sceny z każdej sekwencji. W widoku edytora Unity oraz w trybie symulacji VR (niestety zrzut ekranu obrazu symulatora nie oddaje w pełni estetyki obiektów, które są skonstruowane przede wszystkim do oglądania ich z bliskiej odległości i wielu perspektyw).



Rysunek 97. VR sekwencja pierwsza (edytor/symulator)



Rysunek 98. VR sekwencja druga (edytor/symulator)

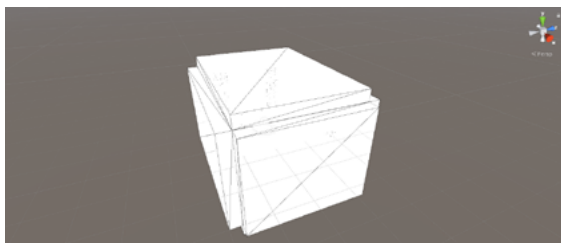


Rysunek 99. VR sekwencja trzecia (edytor/symulator)

4 Źródło informacji - oficjalne materiały producenta

3.1.6. Obiekty

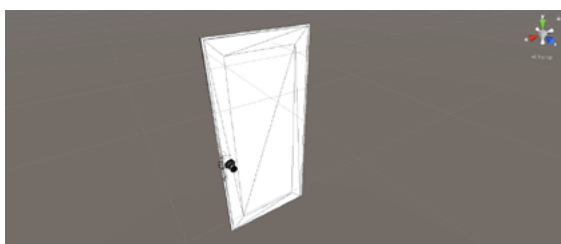
Na potrzeby projektu został przygotowany szereg obiektów 3D do zaprezentowania w wirtualnym środowisku. Część z nich została opracowana samodzielnie, niektóre pochodzą z publicznie dostępnych, darmowych repozytoriów z modelami, część jest przerobionymi tego rodzaju obiektami. Poniżej przegląd najważniejszych modeli:



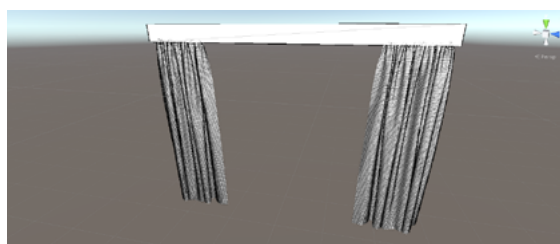
Rysunek 100. Model pomieszczenia
(ściany z zewnątrz)



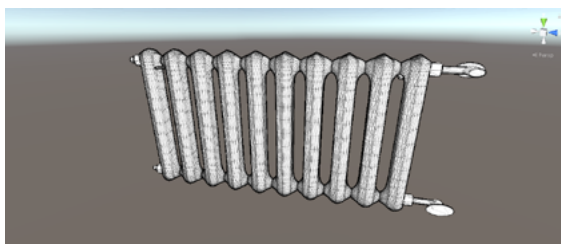
Rysunek 101. Model okna



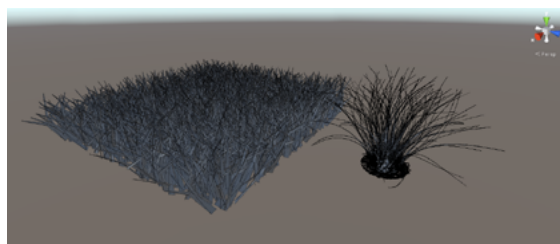
Rysunek 102. Model drzwi



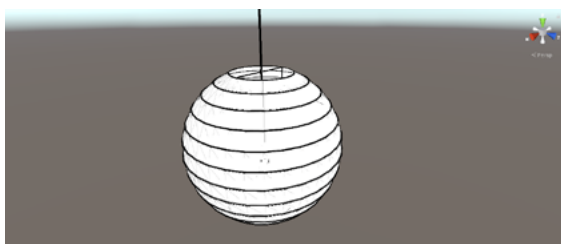
Rysunek 103. Animowany model zasłon



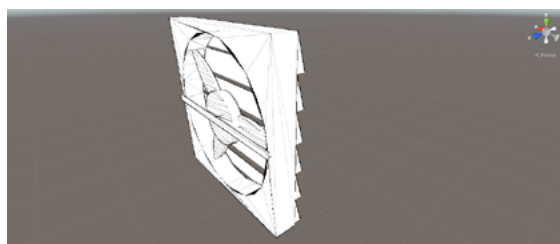
Rysunek 104. Model kaloryfera



Rysunek 105. Model roślinności wyrastającej z podłogi



Rysunek 106. Model lampy



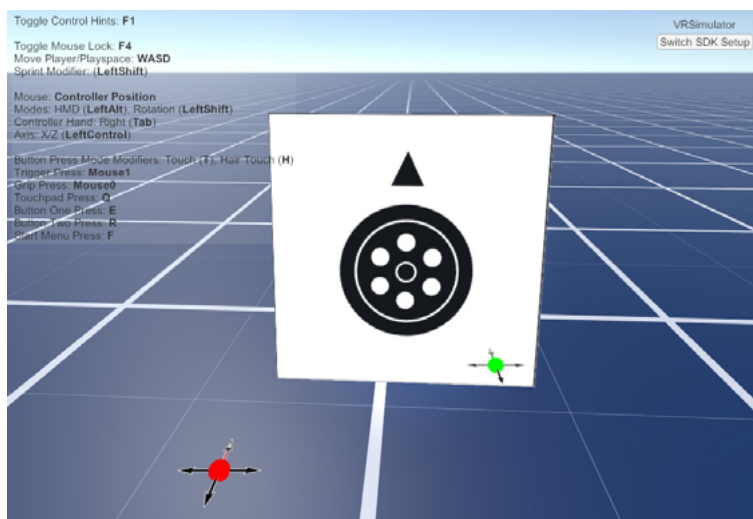
Rysunek 107. Model wentylatora
emitującego cząsteczki



Rysunek 108. Model zlewu z otworu kuchennego

3.1.7. Skrypty/interakcje

Przestrzeń interakcji w „Stilleben” oznaczona jest wymiarami rzeczywistego pomieszczenia w laboratorium UBU. Samych interakcji nie jest wiele - polegają one zasadniczo na uruchamianiu kolejnych elementów sekwencji animacji za pomocą wirtualnych guzików. Poruszanie się oraz interakcje realizowane są za pomocą skryptów dostarczanych przez SteamVR oraz VRTK. Poniższa ilustracja przedstawia konfigurowanie i testowanie przycisku testowego w środowisku VR przy użyciu symulatora dostarczanego z VRTK. Czerwony i zielony punkt to położenie rąk użytkownika:



Rysunek 109. Projektowanie interakcji w środowisku VRTK

Do największych problemów w czasie pracy nad „Stilleben” w wirtualnej rzeczywistości należało stworzenie efektu płaskiego świata, sprawiającego wrażenie wyjętego spod deski kreślarskiej. Zasadniczo Unity 3D wyświetla obiekty z wykorzystaniem wszystkich składowych klasycznego modelu: światła odbitego (*specular*), światła rozproszonego (*diffuse*) oraz światła otoczenia (*ambient*). Światło padając na obiekt jest w odpowiedni sposób odbijane w zależności od charakterystyki wirtualnej powierzchni tworząc realistyczne, trójwymiarowe, oparte na światłocieniu wrażenie optyczne. W „Stilleben” coś podobnego nie może mieć miejsca. Pierwszym momentem było więc wyłączenie elementów *specular* oraz *diffuse*. Światło w instalacji nie ma swojego źródła, jest czymś w rodzaju medium wypełniającego równomiernie całą symulację. Oznacza to m.in. brak cieni. Do opisu powierzchni obiektów użyto specjalnie stworzonych materiałów oraz shaderów⁵, które dają efekt idealnie matowy. Powoduje to, że scena nie posiada żadnej plastyczności. W sytuacji kiedy mamy do czynienia z bezcieniowym, białym boksem i pozbawionymi właściwościami takimi jak odbijanie światła obiektami - wszystko zlewa się w jednolitą biel. Dlatego konieczne okazało się stworzenie specjalnej kamery, która nie tylko „patrzy” na świat ale też przetwarza go w sposób aktywny. W projekcie wykorzystano znaną twórcom animacji oraz gier technikę zwaną *cell-shading*. Polega ona za zastosowaniu specyficznych filtrów, które „spłaszczają” obraz (w mniejszym lub większym stopniu) a uwydatniają krawędzie. Uzyskuje się w ten sposób nieco „komiksowy” efekt znany np. z takich gier jak „Borderlands” (Gearbox Software, 2009) lub filmów podobnych do „Scanner Darkly” (reż. Richard Linklater, 2006).

Rozwiązanie zostało zrealizowane z wykorzystaniem specjalnej kamery oraz zestawu shaderów. Kamera tak naprawdę składa się z dwóch tego typu obiektów realizujących szereg zadań. Całość oparta jest na architekturze zaproponowanej przez użytkownika Unity o pseudonimie „Cakeslice”. Ogólnie i nieco upraszczając - główna kamera wyposażona jest w skrypt

⁵ Shader to krótki program komputerowy, który - mówiąc najprościej - opisuje właściwości pikseli i wierzchołków. Zasady wyliczania i renderowania elementów graficznych.

wykrywania krawędzi (*edge detection*) obiektów na scenie. Jej pierwsza podkamera rejestruje normalne (wektory prostopadłe do płaszczyzny), druga „głębnię”. Skrypt dokonuje odpowiedniego zestawienia danych w wyniku którego jest w stanie wskazać i zabarwić wybranym kolorem krawędzie obiektów:

```

1 if (cam != null) {
2     Vector3 a =
3         -cam.transform.InverseTransformPoint(cam.ScreenToWorldPoint(Vector3.
4         forward));
5         a.z = cam.farClipPlane * farClipPlaneMultiplicand;
6     edgeDetectMat.SetMatrix("_Cam2World", Matrix4x4.TRS(Vector3.zero,
7     cam.transform.rotation, Vector3.one));
8     Shader.SetGlobalVector("_EdgeDetectDepthArgs", a);
9
10    normalsCapturingCamera.fieldOfView =
11    depthCapturingCamera.fieldOfView =
12    cam.fieldOfView;
13    normalsCapturingCamera.farClipPlane =
14    depthCapturingCamera.farClipPlane =
15    cam.farClipPlane;
16
17    if (prevNormalsSensitivity != normalsSensitivity ||
18        prevDepthSensitivity != depthSensitivity ||
19        prevDepth2Sensitivity != depthSensitivity2) {
20        edgeDetectMat.SetVector("_SensitivityAndWidthArgs",
21            new Vector4(normalsSensitivity, depthSensitivity,
22            depthSensitivity2, 0));
23        prevNormalsSensitivity = normalsSensitivity;
24        prevDepthSensitivity = depthSensitivity;
25        prevDepth2Sensitivity = depthSensitivity2; }
26
27    if (prevOutlineColor != outlineColor) {
28        edgeCombineMat.color = outlineColor;
29        prevOutlineColor = outlineColor; }
30 }

```

Pozostałe skrypty stworzone na potrzeby „Stilleben” VR to standardowy kod służący do opisu interakcji, zdarzeń, przejść między scenami, animacji. W sumie na potrzeby „Stilleben” VR wytworzone zostało około 50 skryptów.

3.2. Aplikacja AR - Unity, Vuforia AR, Android SDK

Poniższy opis przebiegu implementacji projektu „Stilleben” do poszerzonej rzeczywistości składa się z siedmiu części. Pierwsze trzy pokrótce opisują użyte technologie, kolejne zaś proces tworzenia scen, zaimportowane obiekty oraz stworzone skrypty (najczęściej do obsługi animacji i interakcji).

3.2.1. Unity Engine

Patrz: 3.1.1

3.2.2. Vuforia Augmented Reality SDK

Vuforia to SKD (*Software Development Kit* - zestaw narzędzi programistycznych i deweloperskich), który umożliwi zaawansowaną pracę z aplikacjami w poszerzonej rzeczywistości. Wykorzystuje technologię rozpoznawania obrazu (*computer vision*) do wykrywania obiektów prostych (płaskie obrazy) oraz złożonych (obiekty 3D takie jak pudełka, walce etc.) w czasie rzeczywistym. Pozwala to na określenie położenia obiektu w przestrzeni względem np. kamery telefonu komórkowego lub tabletu i „zbudowanie” wokół niego układu odniesienia. Dzięki temu manipulując „markerem” (celem) manipulujemy właściwie sceną 3D. Powoduje to stworzenie niezwyklej perspektywy i umożliwia interesujące, hybrydowe interakcje.

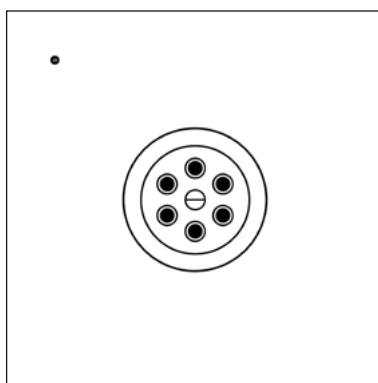
Vuforia wspiera rozwiązania 2D oraz 3D i aktualnie jest jednym z najważniejszych dostawców rozwiązań AR. Posiada API dla języków/środowisk C++, Javy, Objective-C++, .NET i wielu innych. Dostępny jest również zestaw narzędzi w postaci pakietu dla Unity. Od 2017 roku Vuforia jest natywnie zintegrowana z tym silnikiem.

Poza wsparciem dla silników Vuforia udostępnia też usługi w chmurze (hostowanie dynamicznych markerów, bazy obiektów) oraz umożliwia pracę we własnym środowisku (Vuforia Studio), a także coraz poważniej rozwija obsługę dedykowanych urządzeń do poszerzonej rzeczywistości takich jak Microsoft HoloLens czy MagicLeap.

W przypadku projektu „Stilleben” platforma on-line była wykorzystana do stworzenia paczki (pakietu) dla Unity zawierającego markery (ilustracje) wraz z metadanymi umożliwiającymi kamerze ich rozpoznawanie i orientację w przestrzeni 3D. Konieczne było także wygenerowanie klucza licencji aby umożliwić silnikowi korzystanie z narzędzia.

Dygresja: kwestia markerów w poszerzonej rzeczywistości. Do działania prawdziwych aplikacji AR (nie będących po prostu nakładkami graficznymi na obraz z kamery) niezbędne jest „obsłużenie” w jakiś sposób geometrii „świata rzeczywistego”. Najczęściej realizuje się to przez albo: a) wykrycie geometrii za pomocą kamery - jeśli na podstawie samego obrazu np. pomieszczenia software jest w stanie wykryć linie perspektywy to często może „zlokalizować” ściany i podłogę pomieszczenia. Kwestia staje się problematyczna gdy pomieszczenie jest duże i puste lub znajdujemy się na zewnątrz. Współcześnie sprawę ułatwiają nieco nowoczesne telefony komórkowe, które posiadają wiele kamer (niejednokrotnie po 4-6 lub nawet 8) w tym niektóre na podczerwień. Kamery 3D wyświetlają czasem niewidoczne gołym, ludzkim okiem linie i rejestrują ich odkształcenia spowodowane padaniem na rzeczywiste obiekty. W ten sposób mogą budować geometrię. Metod jest coraz więcej i będą zapewne działały coraz lepiej. Na razie jednak nie jest to jeszcze standard; albo: b) wykrycie geometrii za pomocą markera - silnik posiada cyfrowy wzorzec danego obrazka i oprogramowanie potrafi go wykryć w realnej przestrzeni na podstawie cech szczególnych (ficzerów, *features*). To nieco prostsza metoda - wymagająca za to przygotowania odpowiedniej jakości markerów i wgrania ich do systemu. Ten sposób został wykorzystany w projekcie „Stilleben” ponieważ umożliwia on precyzyjne umiejscowienie wirtualnego obiektu „na” ilustracji w publikacji.

Interesującym może być przebieg testów markerów i ich dostosowywanie do technologicznych ograniczeń mobilnego sprzętu. W pierwszej wersji projektu poszczególne markery miały być oparte wyłącznie na minimalistycznych obrazach stworzonych przez Jakuba Woynarowskiego:



Rysunek 110. Ilustracja J. Woynarowskiego. Marker bez ramek

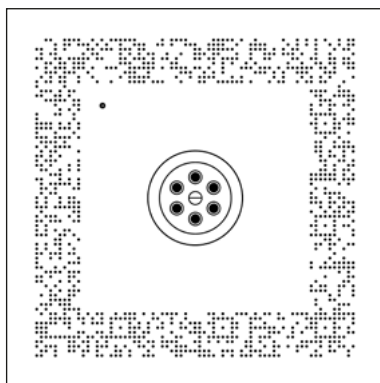
Okazało się jednak, że taki obraz posiada zbyt małą ilość „ficzerów” zaś jego wysoka symetryczność uniemożliwia systemowi rozpoznanie gdzie jest przód w przypadku np. obracania markerem.

Na potrzeby testów do obrazka dołożona została ramka (losowa fotografia ściągnięta z Internetu). Okazało się, że znakomicie podnosi ona czytelność obrazu dla kamery. Psując niestety pożądaną estetykę:



Rysunek 111. Ilustracja J. Woynarowskiego. Marker z ramką testową

Ostatecznie zapadła decyzja o stworzeniu ramek składających się ze spójnych estetycznie z formułą artbooka czarnych punktów. Dla każdego markera został wygenerowany za pomocą skryptu unikatowy zestaw/układ punktów, dzięki któremu kamera nie ma problemów z rozpoznawaniem i orientowaniem markera:



Rysunek 112. Ilustracja J. Woynarowskiego. Marker z wzorcem z kropek

3.2.3. Android SDK

Android SDK to bardzo rozbudowany zestaw narzędzi programistycznych przeznaczony do tworzenia aplikacji na platformę Android. Składa się z SDK Tools oraz Platform Tools (ze względu na ogromną ilość wersji systemu). SDK składa się z dokumentacji, skryptów, przykładowych programów, tutoriali, bibliotek, emulatorów, debuggerów etc.

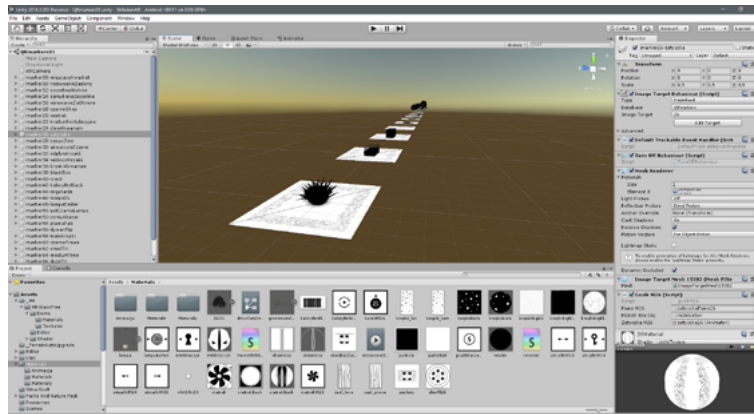
Android SDK jest koniecznym elementem w przypadku publikowania projektów stworzonych za pomocą silnika Unity jako aplikacji mobilnych dla tego systemu (najczęściej w postaci plików .apk).

3.2.4. Obiekty, skrypty i interakcje w podziale na sceny

Poniższa część raportu składa się z opisu poszczególnych (choć nie wszystkich) scen AR. Opis sceny zawiera token. Opis/ilustrację obiektu/interakcji oraz ew. listing kodu odpowiedzialnego za działanie sceny.

Należy zaznaczyć, że aplikacja była realizowana w postaci jednej sceny (vide ilustracja poniżej) zaś ewentualna animowana bądź interaktywna zawartość poszczególnych „platform” pod markerami jest aktywowana specjalnym skryptem reagującym na wykrycie

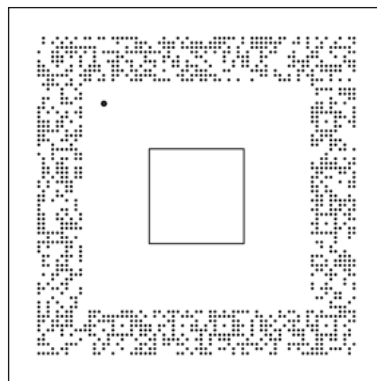
konkretnego markera. Dzięki temu uniknięto przeładowywania scen między markerami choć rozwiązanie to jest nieco bardziej wymagające co do zasobów urządzenia:



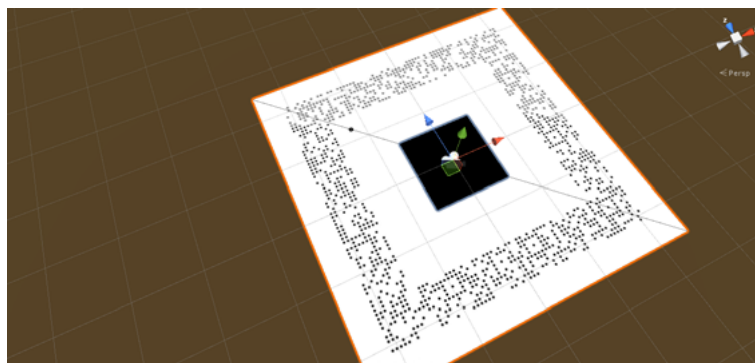
Rysunek 113. Widok projektu aplikacji AR w Unity

Scena01 „...przedstawia biały kwadrat, obwiedziony czarną ramką. W rozszerzeniu AR jasne pole w krótkich, nieregularnych odcinkach czasu wypełnia się czernią. Wywołany w ten sposób efekt migotania wywołuje skojarzenie z awarią instalacji elektrycznej.”

Marker i widok w Unity



Rysunek 114. Wygląd markera dla sceny 01



Rysunek 115. Widok sceny 01 w silniku

Skrypt. W Unity skrypty dodaje się najczęściej jako komponenty do obiektów znajdujących się na scenie. Dzięki temu do niektórych parametrów skryptu - takich jak wybrane rodzaje zmiennych - mamy dostęp z poziomu edytora i nie musimy edytować kodu źródłowego za każdym razem kiedy chcemy coś zmienić lub przetestować.

Zwróćmy uwagę, że każdy skrypt w Unity zawiera na początku dostęp do przestrzeni nazw zawierającej interfejsy i klasy, które definiują różne kolekcje obiektów etc. W poniższym wypadku są to *UnityEngine* (dzięki któremu możemy skryptować specyficzne dla silnika kwestie), *System.Collections* oraz *Vuforia* (dzięki której mamy dostęp do specyficznych klas dla augmented reality). Dodatkowo nasza główna klasa standardowo dziedziczy po *MonoBehaviour* a w przypadku interaktywnych elementów AR (wirtualne przyciski) również po *IVirtualButtonEventHandler*.

Poniżej prosty skrypt, który w pierwszej scenie powoduje losowe w czasie (*WaitForSeconds(Random.Range(MinTime,MaxTime));*) migotanie (*ToggleVisibility*) obiektu będącego czarną płaszczyzną (*public GameObject ObjectToHide*);

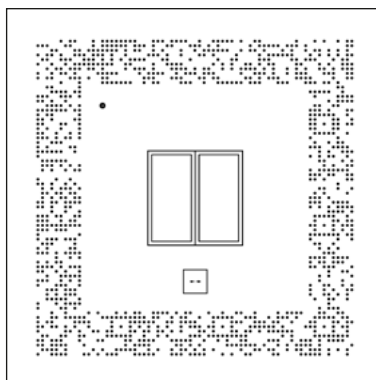
```

1 using UnityEngine;
2 using System.Collections;
3 using Vuforia;
4
5 public class flicker : MonoBehaviour
6 {
7     public GameObject GameObjectToHide;
8     public float MinTime = 0.01f;
9     public float MaxTime = 0.5f;
10
11     void Start()
12     {
13         StartCoroutine(ToggleVisibilityCo(GameObjectToHide));
14     }
15
16     IEnumerator ToggleVisibilityCo(GameObject someObj)
17     {
18         if (someObj == null) yield break;
19
20         while (true)
21         {
22             someObj.SetActive(!someObj.active);
23
24             yield return new WaitForSeconds(Random.Range(MinTime, MaxTime));
25         }
26     }
27 }

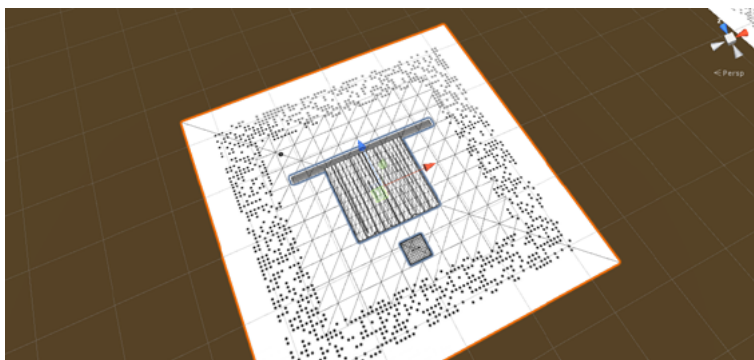
```

Scena02 „...przedstawia uproszczony obraz okna. W rozszerzeniu AR obraz ten zostaje uzupełniony o karnisz i dwie kotary, w pozycji wyjściowej zakrywające szybę. Po naciśnięciu dolnej ikonki (symbol dwóch strzałek, sugerujących ruch skierowany na zewnątrz), zasłony zostają rozsunięte.”

Marker i widok w Unity



Rysunek 116. Wygląd markera dla sceny 02



Rysunek 117. Widok sceny 02 w silniku

Skrypt w drugiej scenie skrypt odpowiada za animowanie wirtualnych zasłon znajdujących się na „realnym” obrazku z oknem. Połączone są więc dwa elementy: 1) animacja (*public Animator zaslLewaRuch* i *zaslPrawaRuch*) oraz 2) wirtualny przycisk, który aktywuje się gdy ikonka zostanie w jakiś sposób zasłonięta przed oknem kamery (np. przez naciśnięcie palcem). Skrypt reaguje na aktywację przycisku: *public void OnButtonPressed(VirtualButtonBehaviour vb)* lub jego odsłonięcie: *public void OnButtonReleased(VirtualButtonBehaviour vb)* aktywując animację odsłonięcia zasłon: *zaslPrawaRuch.Play("zaslonaPrawaAnim");* lub jej odwrotność: *zaslLewaRuch.Play("zaslonaLewaAnimRev");*

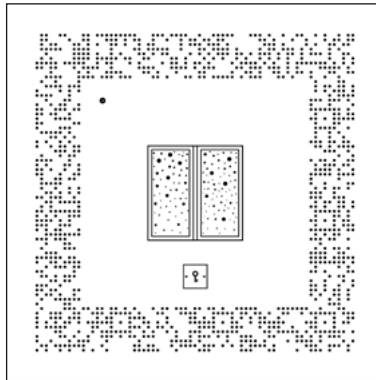
```

1 (...
2
3 public class zaslonyAnim : MonoBehaviour, IVirtualButtonEventHandler
4 {
5     public GameObject zaslonyBtnObj;
6     public Animator zaslLewaRuch;
7     public Animator zaslPrawaRuch;
8     public GameObject butonikM10;
9
10    // Start is called before the first frame update
11    void Start()
12    {
13        zaslonyBtnObj = GameObject.Find("zaslonyButton");
14        zaslonyBtnObj.GetComponent<VirtualButtonBehaviour>()
15            .RegisterEventHandler(this);
16        zaslLewaRuch.GetComponent<Animator>();
17        zaslPrawaRuch.GetComponent<Animator>();
18        butonikM10.SetActive(true);
19    }
20
21    public void OnButtonPressed(VirtualButtonBehaviour vb)
22    {
23        zaslPrawaRuch.Play("zaslonaPrawaAnim");
24        zaslLewaRuch.Play("zaslonaLewaAnim");
25        butonikM10.SetActive(false);
26    }
27
28    public void OnButtonReleased(VirtualButtonBehaviour vb)
29    {
30        zaslLewaRuch.Play("zaslonaLewaAnimRev");
31        zaslPrawaRuch.Play("zaslonaPrawaAnimRev");
32        butonikM10.SetActive(true);
33    }
34 }

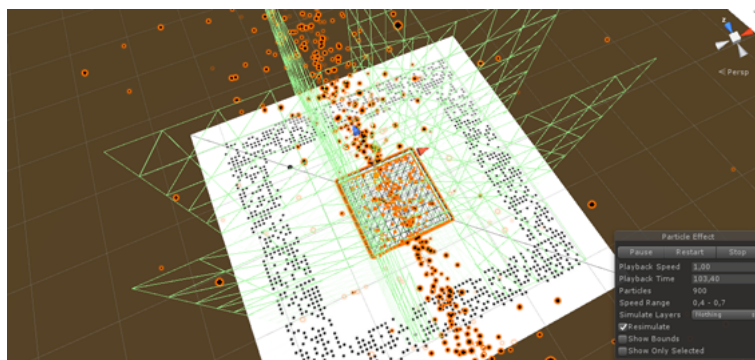
```

Scena 04 ...przedstawia także obraz uchylonego okna, uzupełnionego o - osiadłe na szybach - czarne drobinki. W rozszerzeniu AR cząsteczki „za oknem” zostają wprowadzone w ruch, a niektóre z nich unoszą się do góry przez szczelinę między okiennicami. Po naciśnięciu dolnej ikonki (symbol kluczyka i strzałek sugerujących ruch skierowany do wewnątrz), „wyciek” cząsteczek w górę zostaje wstrzymany.

Marker i widok w Unity



Rysunek 118. Wygląd markera dla sceny 04



Rysunek 119. Widok sceny 04 w silniku

Skrypt. W tej scenie najciekawsze są efekty cząsteczkowe. Na scenie znajdują się dwa ich źródła. Jedno odpowiada za ruch cząsteczek „za szybą”, drugie za ich emisję przez „szczelinę” (*private ParticleSystem m14Kulki;*). W skrypcie znajduje się działający analogicznie do tego w scenie 02 przycisk - tym razem jednak oddziałuje on na emiter.

```

1 // Ustawiony emiter:
2 void Update()
3 {
4     var emission = m14Kulki.emission;
5     emission.rateOverTime = eRate;
6 }

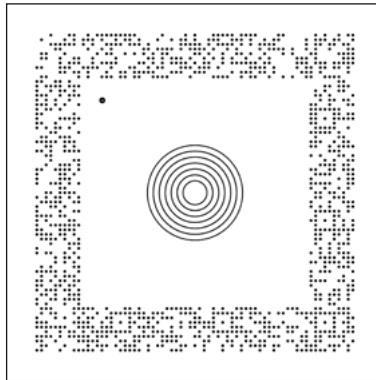
1 // Zwiększanie i zmniejszanie emisji przez interakcje z guzikami:
2 public void OnButtonPressed(VirtualButtonBehaviour vb)
3 { eRate = 5.0f; }
4
5 public void OnButtonReleased(VirtualButtonBehaviour vb)
6 { eRate = 2000.0f; }

```

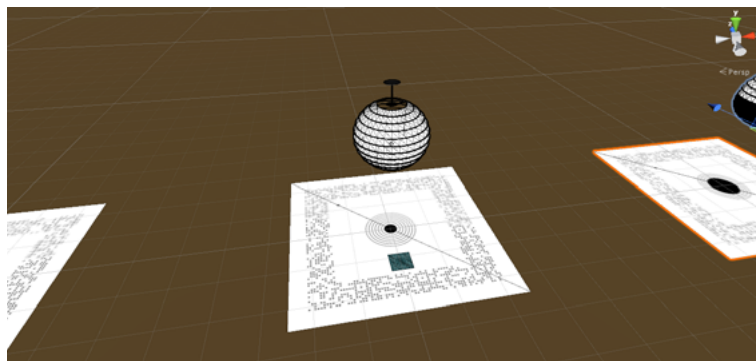
Scena 20 „...przedstawia obraz złożony z koncentrycznych okręgów - to widok kulistego abażuru, widocznego z oddolnej perspektywy. W rozszerzeniu AR nad grafiką unosi się przestrzenna symulacja sferycznej lampy.”

Niewątpliwie najbardziej interesującymi estetycznie formami rzeczywistości poszerzonej są obiekty 3D „zawieszane” w zapośredniczonym obiektywie kamery i ekranem urządzenia „realu”. Najlepiej gdy da się wchodzić z nimi w interakcję - obracać, oglądać, przedstawiać. Przykładem takiego elementu jest scena 20, która przedstawia zawieszoną pod niewidzialnym stropem lampę. Wrażenie przestrzenności potęgowane jest tym właśnie swobodnym zawieszeniem modelu w przestrzeni przy jednoczesnym pozostawianiu w ścisłej relacji do podłoża - markera.

Marker i widok w Unity



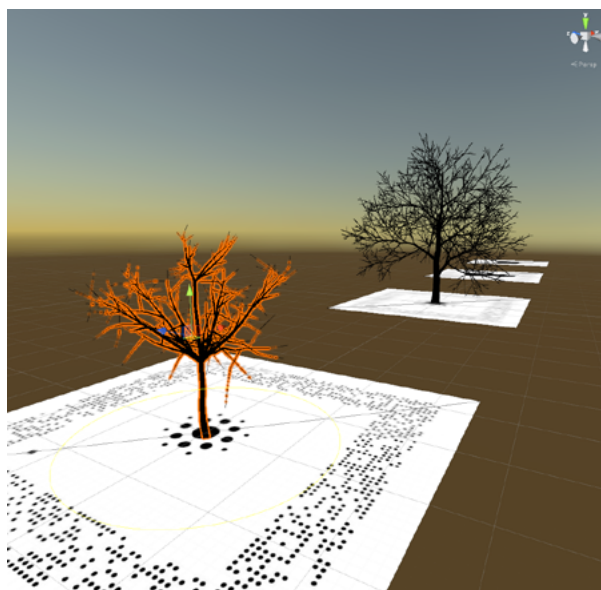
Rysunek 120. Wygląd markera dla sceny 20



Rysunek 121. Widok sceny 20 w silniku

Scena 29 i 30 „... W rozszerzeniu AR z jego środka wyrasta drzewko, podlegające kolejnej fazie rozwoju (...) W rozszerzeniu AR drzewko wyrastające z jego środka osiąga maksymalny rozmiar.”

Kolejnym interesującym z punktu widzenia implementacji elementem jest widoczne w kilku scenach drzewo. Początkowo jest to drobna roślina (scena 28), potem staje się już widocznie „drzewiaste” (scena 29), w ostatniej scenie (30) stając się w pełni rozłożyste.



Rysunek 122. Widok drzew w scenach 29 i 30

Do stworzenia tego elementu użyty został generator proceduralnych drzew. W tym narzędziu można decydować o ilości segmentów, odgałęzieniach, stopniach poskręcania, losowych złamaniach, skręcaniu konarów w kierunku źródła światła etc. Przy użyciu podobnych generatorów jesteśmy w stanie - w razie potrzeby - stworzyć cały las składający się z podobnych wirtualnych roślin lecz żadne spośród nich nie będą identyczne. Jak w rzeczywistości.



Rysunek 123. Komponent do tworzenia proceduralnych drzew

4. Podsumowanie

Na potrzeby projektu AR stworzono 32 sceny, napisano 25 skryptów (około 1000 linii kodu), użytych zostało kilkaset obiektów 3D. Projekt w wersji nieoptymalizowanej zajmuje prawie 1,5GB (sic!), natomiast wygenerowana (nieoptymalizowana) aplikacja 78MB.

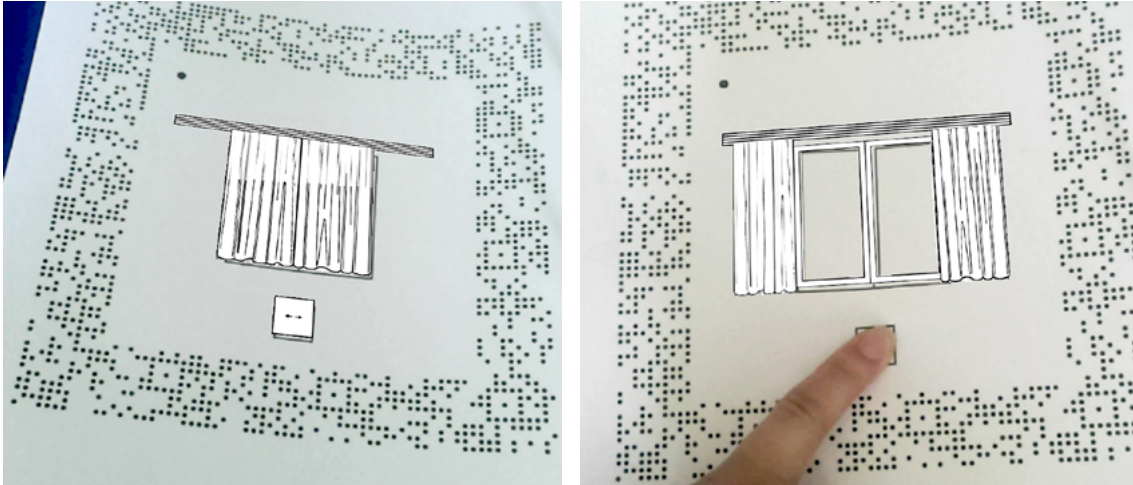
4.1. Testy i release AR

Na etapie produkcji aplikacja była testowana przy użyciu kamer internetowych podłączonych do komputera, które służyły za substytut kamer w urządzeniu mobilnym. Jedna kamera była wykorzystywana do obserwacji „z ręki”, druga - umieszczona na statywie pozwalała obserwować działanie aplikacji bez odrywania się od środowiska developerskiego.

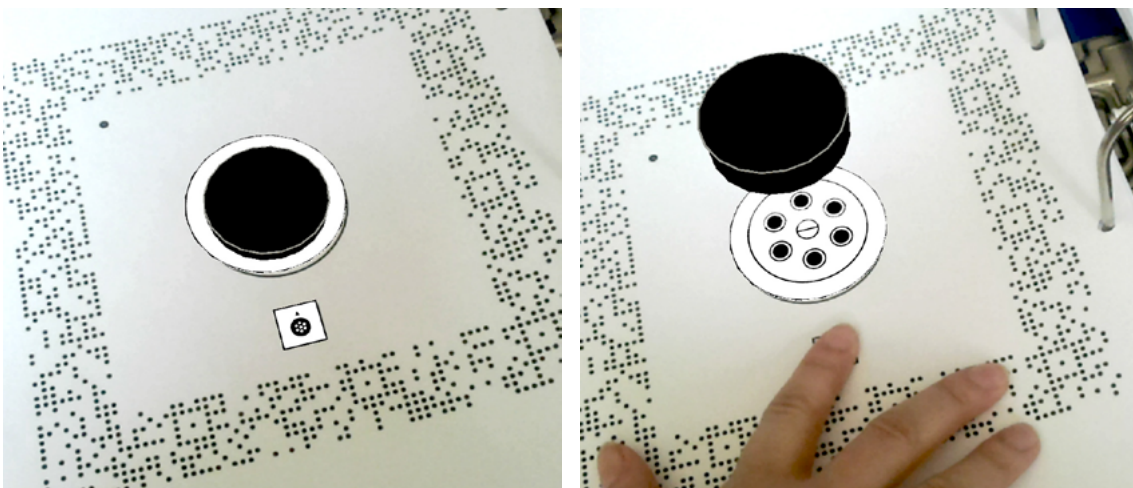


Rysunek 124. Zestaw deweloperski do testowania aplikacji

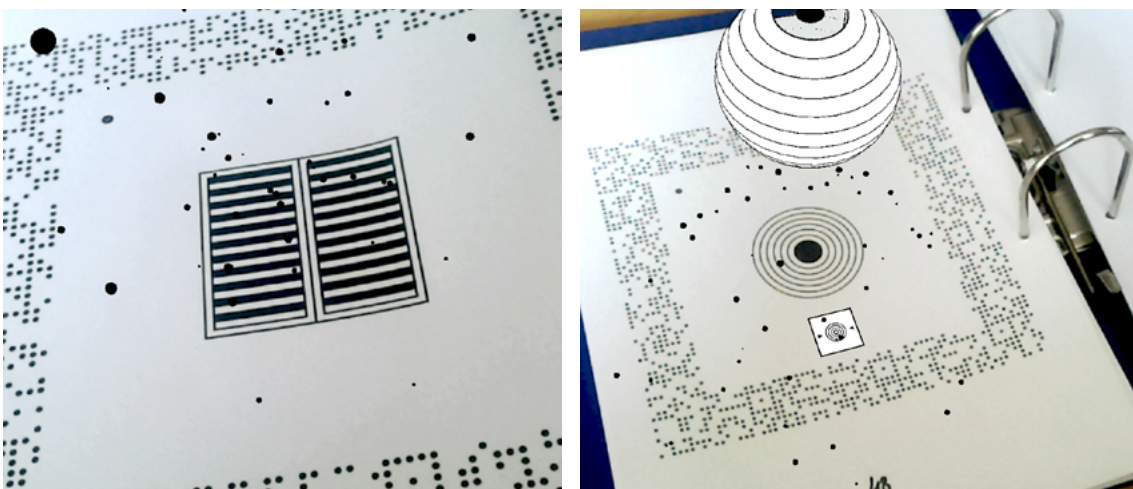
Stworzona aplikacja działa na systemie operacyjnym Android w wersji 7.1.2 lub wyższej. Urządzenie musi być wyposażone w kamerkę i dysponować odpowiednią mocą obliczeniową. Aplikacja testowana była na dedykowanym do poszerzonej rzeczywistości telefonie Lenovo Phab 2 Pro oraz standardowym, „przeciętnym” urządzeniu - Xiaomi Redmi 4A.



Rysunek 125. Animacja uruchamiana „przyciskiem” w AR

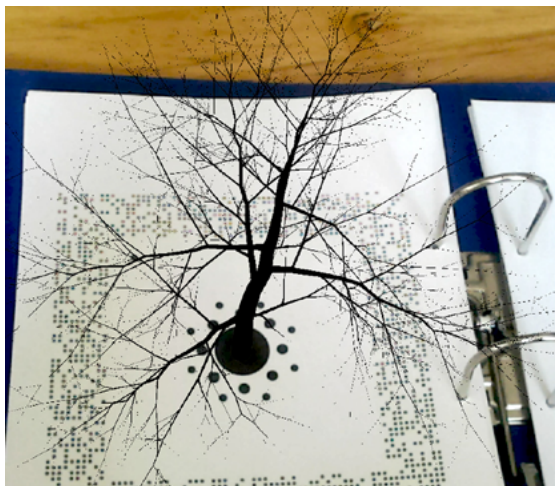


Rysunek 126. Animacja uruchamiana „przyciskiem” w AR



Rysunek 127. Cząsteczki unoszące się „ponad” kartką

Rysunek 128. Lampa „pochłania” cząsteczki

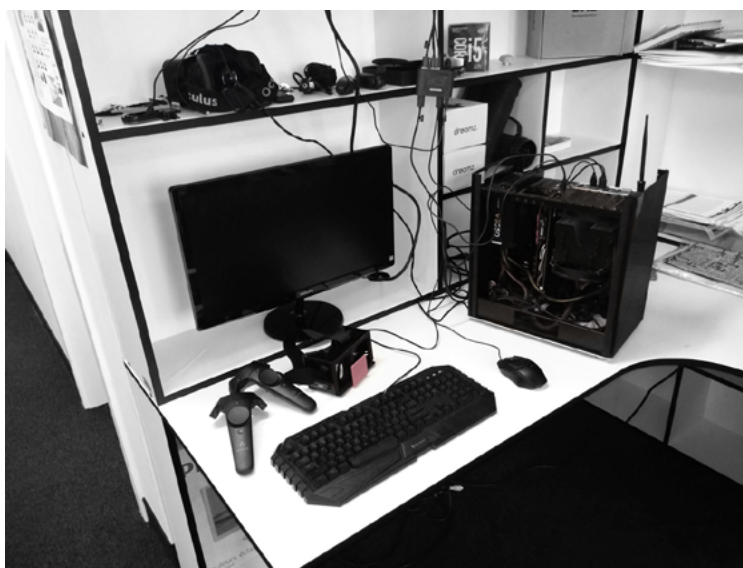


Rysunek 129. Proceduralnie wygenerowane drzewo

Aplikacja dostępna jest w repozytoriach UBU lab - docelowo na stronie internetowej publikacji. Instalacja aplikacji wymaga wyrażenia zgody w systemie na instalowanie obcego (nieautoryzowanego) oprogramowania. Kwestia przekazania aplikacji do Google Play Store jest w tym momencie rozważana.

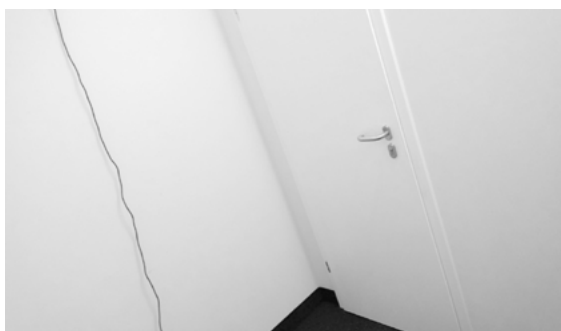
Testy i release VR. „Stilleben” VR powstała i jest uruchamiana na dedykowanym komputerze zdolnym do uruchamiania wymagających projekcji VR. Labowy PC posiada następujące parametry:

- procesor Intel i5-7600 3.50GHz 6MB
- karta graficzna MSI GeForce GTX 1060 6GB GDDR5
- pamięć HyperX 8GB 133MHz Fury Black CL14 x2
- płyta główna MSI Z270M MORTAR
- dysk WD 1TB 7200 obr. 32MB
- dysk WD 250GB SSD
- zasilacz SilentiumPC 550W Supremo M2 Gold
- chłodzenie SilentiumPC Fortis 3
- obudowa HeadTrip VR
- zestaw peryferiów Octane
- monitor 24" Philips

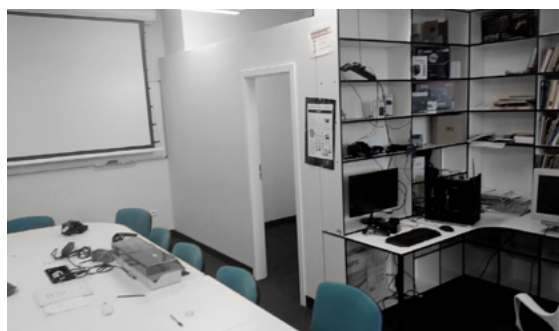


Rysunek 130. PC VR setup w UBULab

Instalacja VR jest przystosowana dokładnie do specyficznego miejsca - zbudowanego dedykowanego pomieszczenia w laboratorium w budynku Wydziału Zarządzania i Komunikacji Społecznej UJ w Krakowie. Pomieszczenie zostało zmapowane w taki sposób, że wirtualne ściany pokrywają się dokładnie z realnymi. Dzięki temu usunięty został jeden z głównych blokerów immersji w wirtualnej rzeczywistości - niematerialność widocznych obiektów. Standardowo w tego typu doświadczeniach użytkownik widzi ścianę ale nie może jej dotknąć, może natomiast przez nią przejść. Jest to jedno z bardzo istotnych ograniczeń realizmu w tej technologii, które nie ma na razie racjonalnego (tzn. nie będącego absurdalnie drogim albo nieergonomicznym) rozwiązania. W „Stilleben” wirtualne i realne ściany są takie same co oznacza, że użytkownik wyciągając rękę w stronę wirtualnego obiektu napotyka na prawdziwe ograniczenie - ten fakt drastycznie zwiększa klaustrofobiczne poczucie zamknięcia - jeden z kluczowych elementów instalacji.



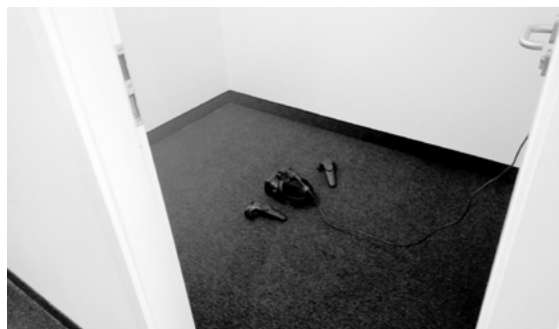
Rysunek 131. Wnętrze pomieszczenia VR (widoczny kabel od HMD)



Rysunek 133. Pomieszczenie VR w kontekście laboratorium



Rysunek 132. Stacja bazowa HTC Vive na ścianie boksu



Rysunek 134. Wnętrze boks z zestawem czeka na widza

„Stilleben” VR był testowany w laboratorium z udziałem autorów, osób związanych z labem oraz uczestników na dedykowanym zestawie sprzętu i oprogramowania. Ponieważ ze względu na charakter utworu nie jest planowane udostępnianie go w postaci aplikacji do pobrania - optymalizacja i testy multiplatformowe nie były wymagane.



Rysunek 135. J. K. Argasiński testuje aplikację w UBULab

Literatura

- [1] G. Doré, *The Rare and Extraordinary History of Holy Russia*. Londyn: Pallas Athene Arts, 2017.
- [2] R. Fludd, *Utriusque Cosmi... metaphysica, physica atque technica Historia*. Oppenheim & Frankfurt: &c, 1617-24.
- [3] P. Gravett, *Comics art*. Londyn: Yale University Press, 2013.
- [4] S. Lem, *Ciemność i pleśń*. Kraków: Wydawnictwo Literackie, 1988.
- [5] E. Lissitzky, *A Suprematist Tale of Two Squares in Six Constructions*. London: Tate, 2014.
- [6] R. McGuire, *Here*. Nowy York: Pantheon, 2014.
- [7] OuBaPo, *OuPus 1*. Paryż: L'Association, 1996.
- [8] W. Phoenix, *Rumble Strip: If You Want to Get Away with Murder, Buy a Car*. Brighton: Myriad Editions, 2008.
- [9] N. Rüegg, *SPUK*. Zurich: Edition Fink, 2004.
- [10] M. Vaughn-James, *The Cage*. Toronto: Coach House Press, 2013.
- [11] C. Ware, *Building Stories*. Nowy York: Pantheon, 2012.