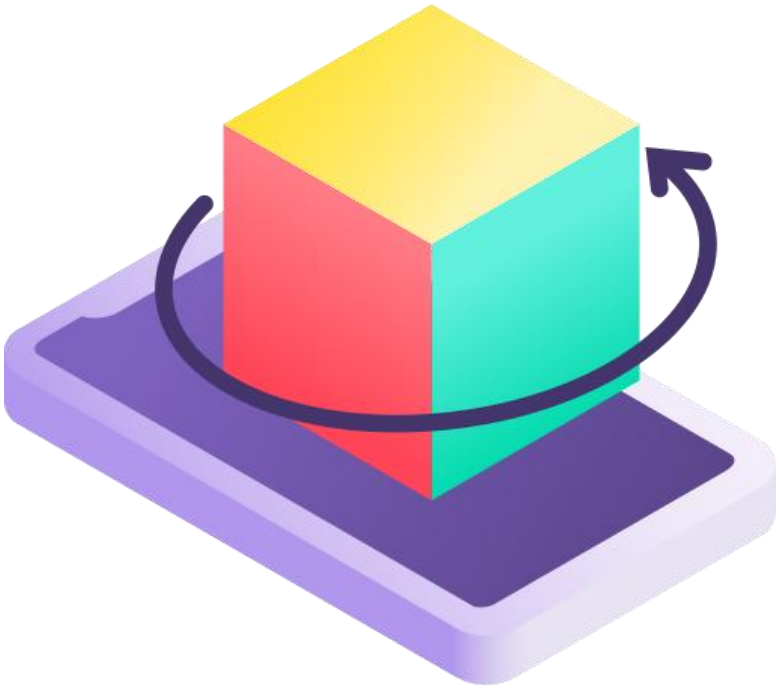
LeARn

Dokumentacja Projektu



Zespół nr 1

Spis treści

I. Członkowie zespołu

Opiekun:

- dr Marta Burzańska

Zespół:

- Adrian Augustyniak
- Aleksander Wierzchowski
- Kacper Pogodziński
- Kamil Orczyk
- Mateusz Przystawski
- Wiktoria Ziętak

II. Opis projektu

Słowa wstępu

Wstępnym założeniem projektu było stworzenie aplikacji wykorzystującej technologie AR[1] do wizualizacji algorytmów i struktur w 3D w sposób interaktywny.

Cel projektu:

- Stworzenie nowoczesnego, odpowiadającego na aktualne potrzeby narzędzia wspierającego zrozumienie algorytmów w sposób jak najbardziej angażujący.
- Wizualizacja przestrzenna to forma przekazu przyspieszająca zrozumienie konceptu danego przypadku w sposób stymulujący.

Grupa docelowa

Grupą docelową są studenci informatyki oraz uczniowie szkół średnich, czyli osoby posiadające elementarną wiedzę na temat algorytmów, chcących zgłębić świat struktur danych i algorytmów.

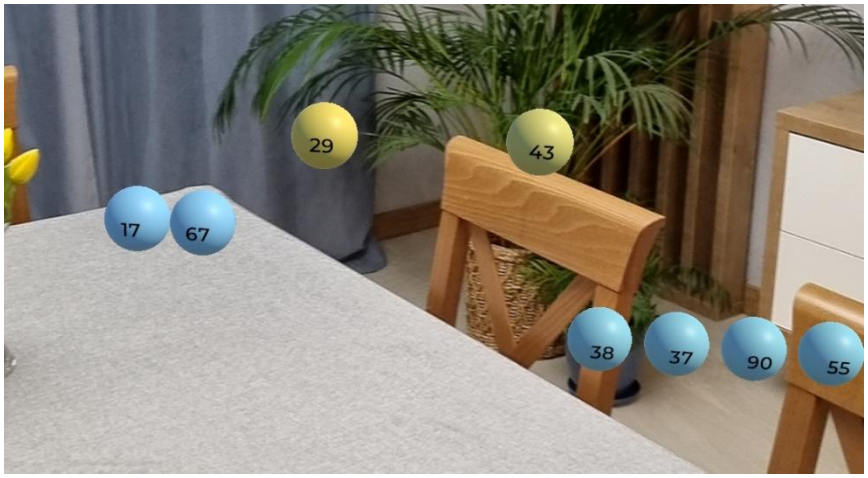
[1] Rozszerzona rzeczywistość (ang. augmented reality)

III. Algorytmy

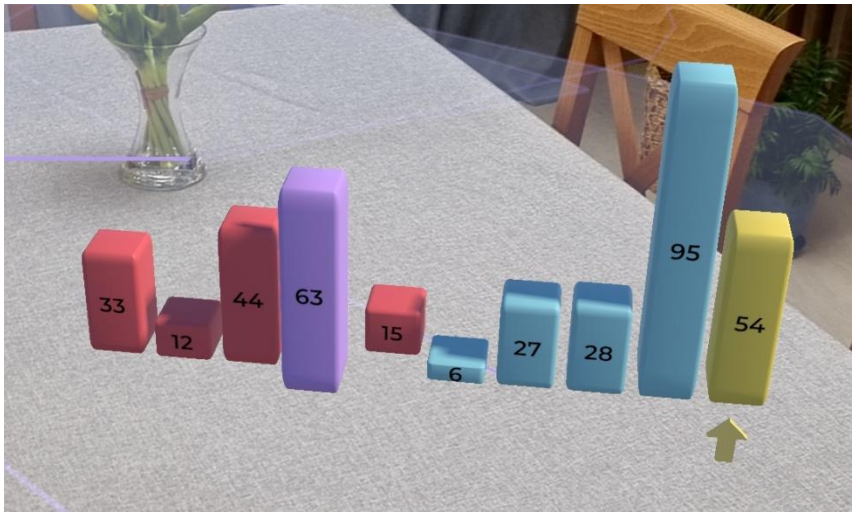
Dzięki AR użytkownik ma możliwość wizualizacji abstrakcyjnych struktur. Dane ukazane są w postaci obiektów 3D które można „zobaczyć z każdej strony” tak, by w swoim tempie zrozumieć cały proces.

Sortowania:

[Sortowanie bąbelkowe]

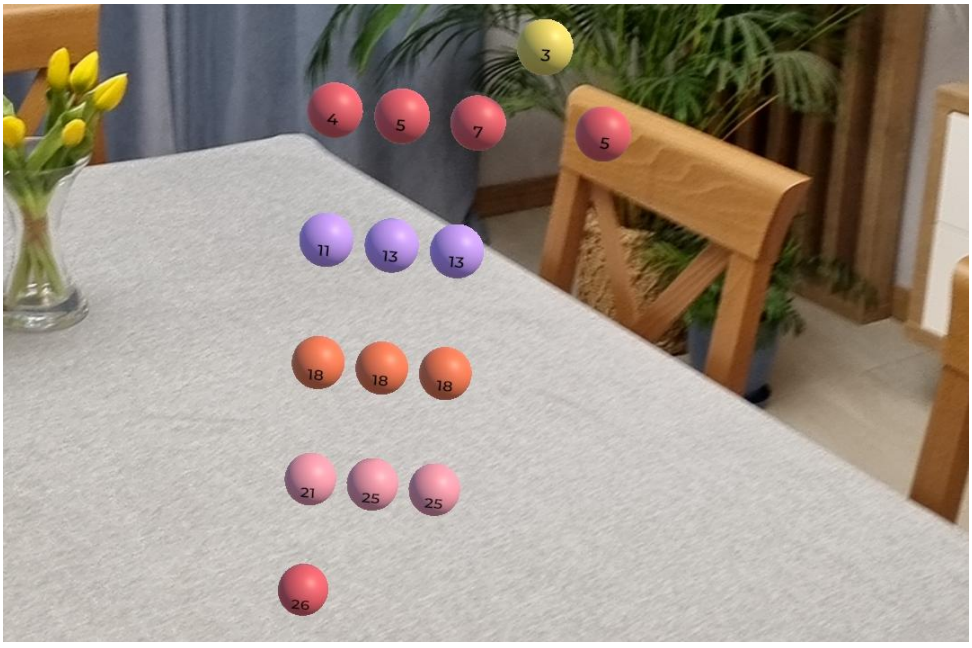


[Sortowanie szybkie]



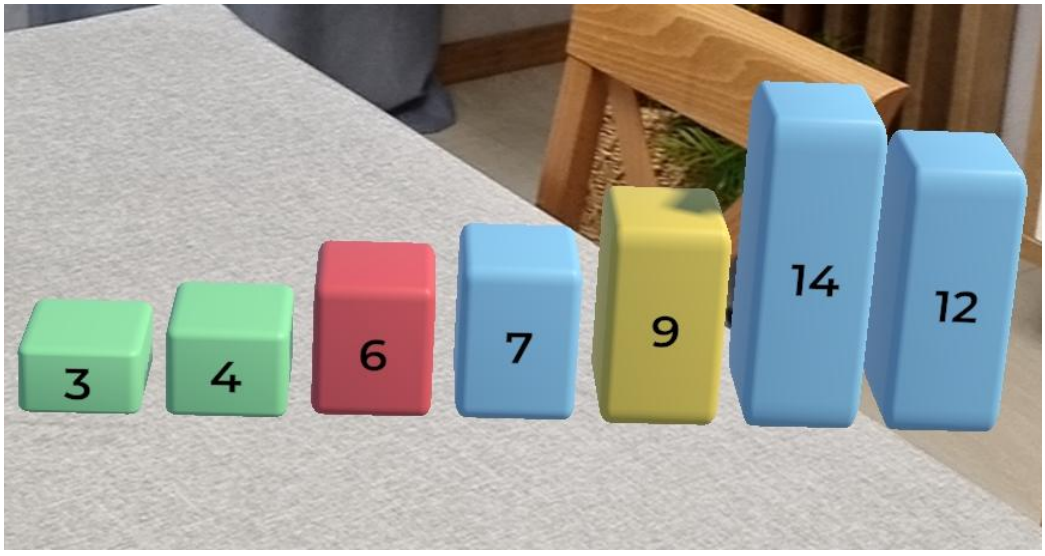
[Sortowanie przez scalanie]

[Sortowanie kubełkowe]



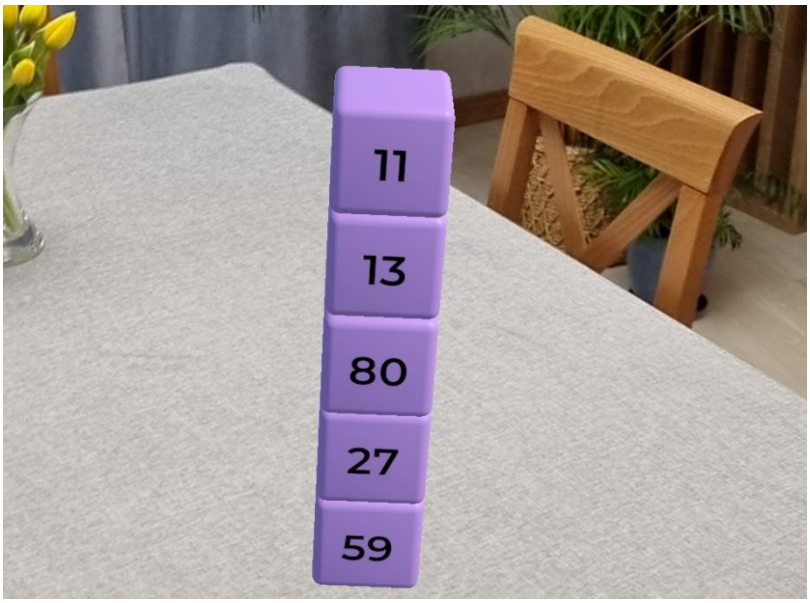
[Sortowanie przez wstawianie]

[Sortowanie przez wybór]

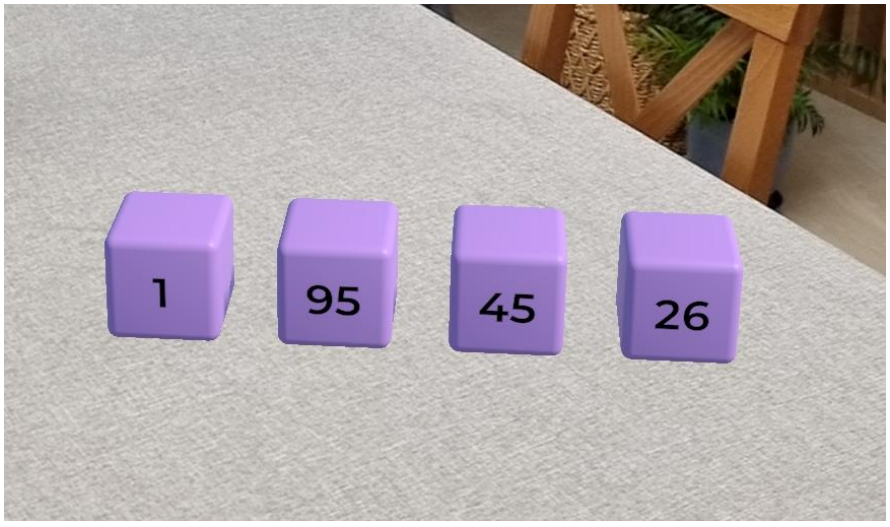


Struktur:

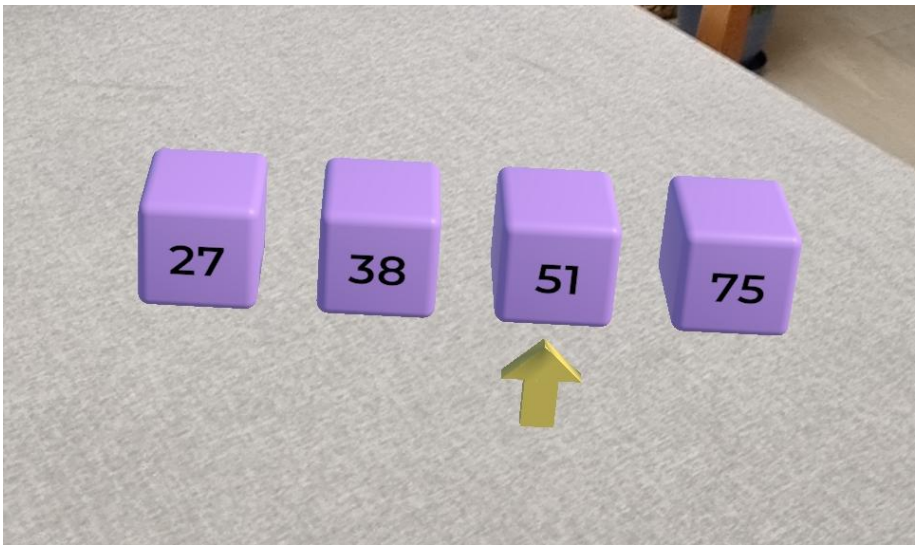
[Stos]



[Kolejka]

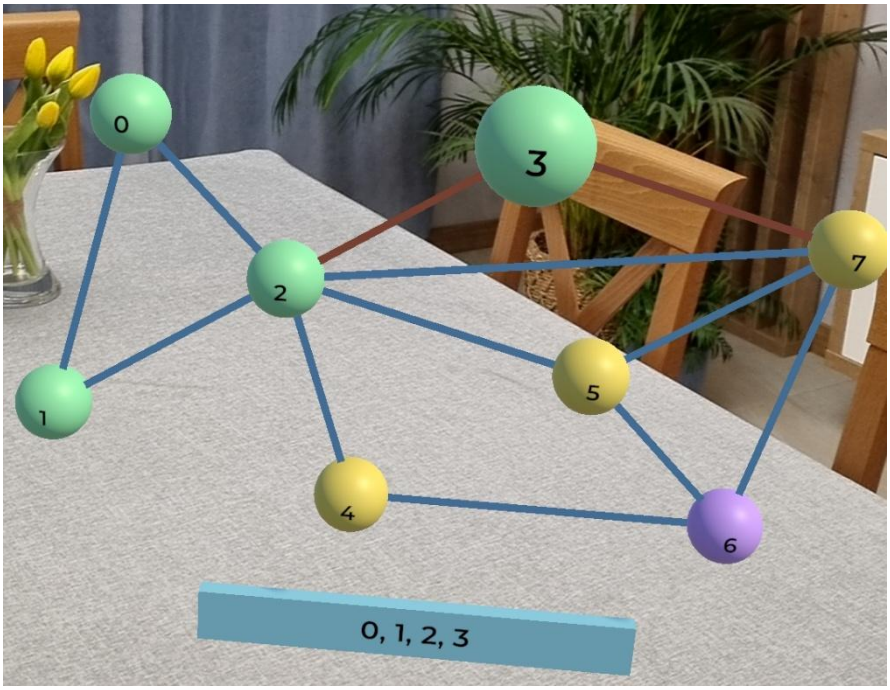


[Lista]

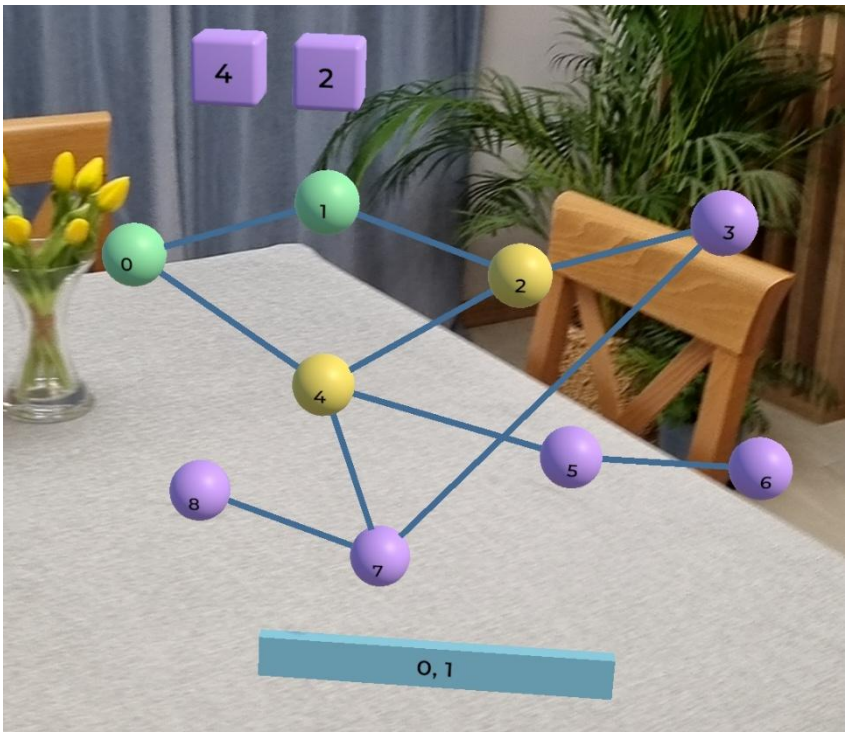


Grafów:

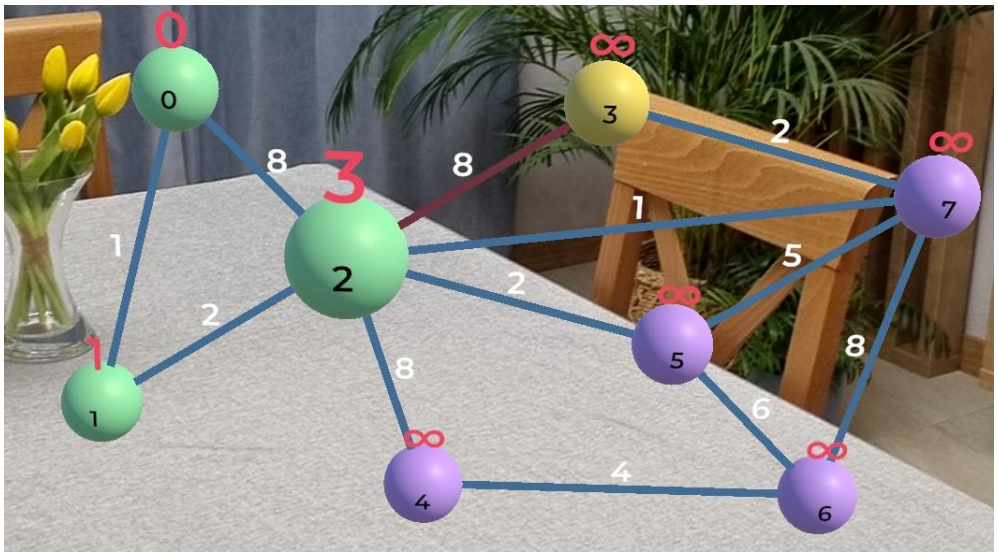
[DFS]



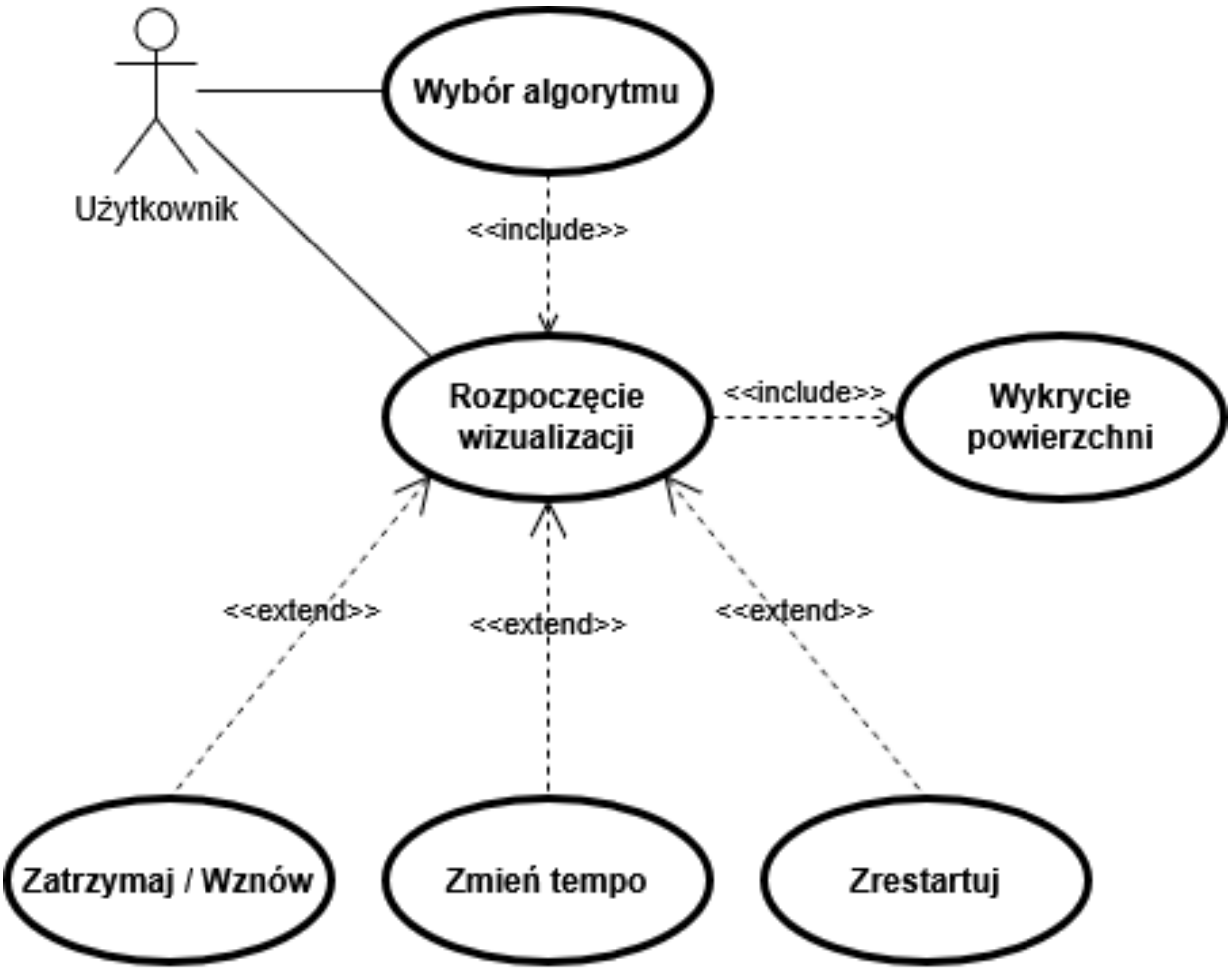
[BFS]



[Algorytm Dijkstry]



IV. Diagram przypadków użycia



V. Architektura aplikacji

Aplikacja została podzielona przy pomocy architektury warstwowej.

Warstwa UI

Odpowiada za interakcje z użytkownikiem. Zawiera komponenty takie jak:

- Menu wyboru
- Przyciski sterujące
- Wiadomości tekstowe

Warstwa AR

Odpowiada za funkcje AR. To między innymi wykrywanie powierzchni, umieszczanie obiektów czy praca kamery. W aplikacji te warstwę reprezentują komponenty:

- AR Session
- XR Origin

Warstwa Logiki Aplikacji

Odpowiada za wywołanie odpowiedniego algorytmu oraz kontrolę i obsługę jego przebiegu. Najważniejsze klasy, od których wychodzi kontrola logiki, to:

- Dictionaries
- OpenAR
- Sortings
- Structures
- Graphs

Uzasadnienie podejścia warstwowego

Jak widać, aplikacja jest podzielona pomiędzy poszczególne funkcjonalności, co znacząco ułatwia kontrolę i testowanie poszczególnych segmentów, jak i rozszerzanie o kolejne elementy w zakresie danej kategorii.

VI. Dodawanie nowych algorytmów

Tworzenie skryptu

Pierwszym krokiem jest stworzenie skryptu w C# odpowiadającego za całą logikę („Assets/Scripts/”).

Ważne, aby skrypt dziedziczył z jednej z klas abstrakcyjnych. Obecnie przygotowane to:

- Sortings
- Structures
- Graphs

Zależnie od implementowanego algorytmu, dobieramy odpowiedną klasę abstrakcyjną. Obecnie, do prawidłowego działania minimalnie potrzebne, do obsłużenia zmienne to:

- Anim Duration (float): odpowiada za czas trwania animacji Lerp.
- Timeout (float): Jest to czas między kolejnymi krokami algorytmu. Bez implementacji timeout, algorytm wykonywałby się od razu bez wizualizacji.

- Prefab (GameObject): Jest to prefab modelu 3D wykorzystywanego przy prezentacji działania algorytmu. Wcześniejsze przygotowanie oraz przekazanie do skryptu (patrz „Tworzenie Prefabu Animacji”)

Należy zaznaczyć, że są to tylko podstawowe zmienne wynikające z aktualnych potrzeb (wymagań do prawidłowej integracji z aplikacją). Tworząc nową klasę należy bezwzględnie pamiętać o tych parametrach przez wzgląd na fundamentalne założenia projektu i jego skalowalność.

Niedopuszczalna jest sytuacja stworzenia nowego algorytmu, który nie dziedziczy z żadnej klasy abstrakcyjnej. W przypadku, gdy żadna z już istniejących klas abstrakcyjnych nie wpisuje się z założenia nowo tworzonego algorytmu, należy napisać nową z uwzględnieniem powyższych zasad integralności.

Tworzenie prefabu animacji

Drugim krokiem jest stworzenie prefabu „Animacji”, czyli obiektu „GameObject”, do którego będzie przypisany skrypt, przekazane domyślne parametry oraz zapisany jako prefab, by móc dowoływać się do niego w kodzie („Assets/Resources/Animations/”).

Uwaga: Nazwa prefabu animacji nie jest przypadkowa. W kodzie odwołujemy się właśnie po nazwie, więc powinna być ona tworzona wedle wzoru:

- Pierwsza część: konkretna nazwa algorytmu
- Druga część: kategoria.

Wszystko powinno być pisane łącznie w stylu PascalCase, przykłady:

- Algorytm szybkiego sortowania: „QuickSort”
- Struktura kolejki: „QueueStruct”

Jednym z parametrów przekazywanych w „Inspector” w Unity jest „prefab” obiektu 3D.

Aktualnie nie ma konkretnych restrykcji dotyczących konkretnego rodzaju, choć należy zachować pewne zasady:

Dla sortowań:

- Model powinien posiadać label (etykietę) z numerem po obu stronach (przód | tył)
- Jeśli to możliwe (ma sens), należy zaokrąglić krawędzie

Dla struktur:

- Identycznie jak w sortowaniu

Dla grafów:

- Prefab animacji posiada na sztywno wklejone „obiekty potomne” z ustawionymi odpowiednimi układami. Należy zduplikować inny prefab animacji grafu zawierający te układy by zachować spójność.
- Niezalecana jest zmiana/edycja tych układów. W przypadku, gdy chcemy dodać kolejny układ, należy pamiętać o obsłużeniu dodatkowego przypadku dla każdego innego grafu.

Dodanie algorytmu do menu

Kolejnym krokiem jest dodanie nowo powstałego algorytmu do menu wyboru.

Przechodzimy do sceny „MainMenu” i w hierarchii obiektów otwieramy:

- („Canvas/ScrollArea/Content”)

Tam duplikujemy ostatni przycisk, następnie zmieniamy kolejno (w inspektorze):

- Nazwę dokładnie tak samo jak prefab Animacji
- Tag na odpowiednią kategorię algorytmu (klasa abstrakcji), odpowiada za filtrowanie w menu
  - W przypadku braku odpowiedniego tagu, należy go stworzyć oraz edytować skrypt odpowiedzialny za filtrowanie.
- Tekst wyświetlany na przycisku
- Kolor przycisku zgodny z kolorystyką aplikacji

Sam przycisk należy umiejscowić w hierarchii jako ostatni w danej kategorii, np.:

- Jeśli mamy już przyciski („ListStruct”, „QueueStruct”, „MergeSort”) i chcemy dodać nową strukturę, to wstawiamy ją pomiędzy „QueueStruct” i „MergeSort”.

Dopasowanie UI

Jeśli wykonano wszystkie powyższe zalecenia, algorytm powinien być już dostępny z poziomu aplikacji.

W razie potrzeby, należy dostosować przyciski w UI w scenie AR oraz stworzyć skrypt odpowiadający za ich zachowane.

Opis i lista kroków

Ostatnim krokiem jest dodanie opisu, listy kroków oraz złożoności.

Uwaga: W skrypcie „Assets/Scripts/Dictionaries.cs” jest słownik zawierający nazwę algorytmu oraz przypisany mu indeks. Należy dodać kolejny wpis na końcu, uzupełniając o odpowiednie dane. Od tego momentu aplikacja będzie próbować zaciągać dane z poniższych plików. Ich zawartość nie jest przypadkowa, każdy wpis zawarty jest pomiędzy „###”. Ważne, aby nie pomylić kolejności, ponieważ opisy dobierane są na podstawie indeksów. Znaczy to tyle, że jeśli np. „BubbleSort” ma indeks „0” to opis będzie szukany w pliku „descriptions” jako pierwszy wpis między znakami „###”.

Ścieżki do plików:

Opis:

- "Assets/Resources/Dictionaries/descriptions.txt"

Lista kroków:

- "Assets/Resources/Dictionaries/steps.txt"

Złożoność:

- "Assets/Resources/Dictionaries/complexity.txt"

VII. Testy

Aplikacja została przetestowana pod kątem wydajności oraz potencjalnych błędów.

Testy UI:

- Działanie przycisków UI

  Sprawdzenie działania przycisków UI jak restart czy pauza/start.

- Responsywność i intuicyjność interfejsu

  Aplikacja została poddana sprawdzeniu, czy osoby trzecie naturalnie (intuicyjnie) rozumieją działanie aplikacji i zawartych w niej funkcjonalności.

- Czytelność

  Teksty oraz ikony zostały ponownie przeanalizowane w celu dopasowania do stylistyki przy jednoczesnym zachowaniu minimalizmu i czytelności.

Testy AR:

- Wykrywanie powierzchni

  Poprawność wykrywania płaszczyzny oraz stabilności obiektów położonych na niej.

- Obsługa kliknięcia

  Testy tworzenia obiektów na wykrytej płaszczyźnie przy jednoczesnej próbie tworzenia obiektów poza nią.

- Wpływ otoczenia

  Sprawdzono wpływ oświetlenia oraz różnego rodzaju powierzchni na działanie AR.

Testy wydajności

Aplikacja została sprawdzona na telefonach z systemem android różnych producentów. Uwzględniając wymagania minimalne w postaci androida w wersji 7.1 na wszystkich sprawdzonych urządzeniach aplikacja działała poprawnie.

- Lista modeli telefonów wykorzystanych do testów:
  - Samsung Galaxy S21 FE 5G (Android 14)
  - LG G6 H870 (Android 8)
  - Samsung Galaxy S21 5G (Android 14)
  - Motorola Edge 40 neo (Android 13)
  - Xiaomi Poco X3 Pro (Android 13)
  - Oneplus Nord 4 (Android 15)
  - Samsung S24 Ultra (Android 15)
  - Samsung Galaxy Tab S9+ (Android 15)
  - Samsung Galaxy A16 (Android 14)
  - Samsung Galaxy A5 2017
  - Samsung Galaxy A53

Wnioski

Testy wykazały, że główne funkcje działają zgodzie z założeniami.

Zauważono, że niektóre ikony były kojarzone z zupełnie inną funkcjonalnością niż ta, którą faktycznie pełniły. Podjęto działania mające na celu zmianę „wadliwych” ikon. Ponowne testy nie wykazały kolejnych błędów w tym zakresie.

VIII. Wykorzystane technologie

Platforma docelowa

Projekt LeARn to aplikacja mobilna na system android w min. wersji 7.1 (Android Nougat)

Silnik Aplikacji

Aplikacja została stworzona w Unity, wraz z wykorzystaniem języka C#

Pakiet do tworzenia AR

AR Foundation to platforma do tworzenia aplikacji rozszerzonej rzeczywistości.

Program do grafiki 3D

Oprogramowanie służące do tworzenia grafiki 3D

Unity:

- Wersja:

  2022.3.53f1

- Dokumentacja:

  https://docs.unity.com/

- Licencja:

  https://unity.com/products/unity-personal/

AR Foundation:

- Wersja:

  5.1.5

- Dokumentacja:

https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/index.html

- Licencja:

Jest to pakiet dostępny w Unity i podlega tym samym zasadą. Pakiet korzysta ARCore(5.1.5) firmy Google na licencji „APACHE LICENSE”.

https://developers.google.com/ar/develop/terms
https://www.apache.org/licenses/LICENSE-2.0

Blender:

- Licencja:

https://www.blender.org/about/license/
https://www.gnu.org/licenses/gpl-3.0.html

Wymagania:

- Android:

  Urządzenie z systemem android 7.1 i nowsze.

- Urządzenie obsługujące ARCore:
  Do działania wymagany jest zainstalowany pakiet ARCore, pełna lista urządzeń obsługujących:
  https://developers.google.com/ar/devices?hl=pl