**Призначення та коротка характеристика програми**

*Мета*

Метою проекту “VR Ideas” поєднати між собою *Speech Recognition, Image Search* та WebVR задля миттєвої візуалізації мови картинками та побудувати новий User Experience у мережі Веб.

*Mind-mapping*

Програма “VR Ideas” розроблена для mind-mapping у віртуальній реальності. Mind-mapping – це один з найкращих способів візуалізувати та структурувати думки. Цей метод – це практично діаграма, котра поєднує інформацію, щодо конкретної теми чи об’єкту. Mind-mapping використовується практично в будь-чому, що вимагає навчання, планування, розвиток чи організацію думок. Відповідно програма може бути використана для визначення цілей, запам’ятовування інформації, розв’язку проблем та планування проектів.

*Проблема*

Проблема, з котрою стикаються люди, що використовують даний метод – це те, що малювання картинок на папері, їх пошук в інтернеті чи запис тексту займає чимало часу та перериває процес. Саме тому ми розробили програму, котра за мінімальний час та з максимальною ефективністю візуалізує думки, використовуючи Speech Recognition та Image Search. Користувачу більше не потрібно відволікатись чи витрачати час на непотрібні процеси, він повністю занурюється в VR середовище, обирає місце в вільній кімнаті, каже слово і воно візуалізується. Отже, проект “VR Ideas” суттєво економить час та суттєво підвищує концентрацію, що і забезпечує ефективність.

*Аудиторія*

Як відомо, більшість людей – візуали (65%), тобто найкраще сприймають інформацію, котру бачать. Наш проект орієнтований на проектних менеджерів, студентів, дизайнерів, та користувачів mind-mapping.

*Технології*

Програма “VR Ideas” використовує наступні технології:

* *WebVR,* а конкретніше бібліотеку “A-Frame” – для створення веб-додатку з VR середовищем
* *HTML & CSS* – для бази веб-додатку
* *JS* – для створення динамічним об’єтів в “A-Frame” та UX інтеракцій
* *Python Flask* – для серверної частини
* *Google Speech Recognition* – для перетворення голосу в текст, за яким пізніше буде здійснено пошук картинок
* *Bing Image Search API* – для пошуку картинок, котрі будуть візуалізовані

**Вхідні та вихідні дані програми**

* Вхідними даними програми є голос користувача, записаний мікрофоном.
* Він обробляється Google Speech Recognition і повертає текст запиту.
* Далі, цей текст обробляється Bing Image Search API, котрий повертає картинку.
* Ця картинка встановлюється як кожна грань обраного куба за допомогою бібліотеки

“A-Frame”, котру вже бачить користувач і котра є вихідними даними програми.

**Структура програми з коротким описом модулів, функцій, класів та методів.**

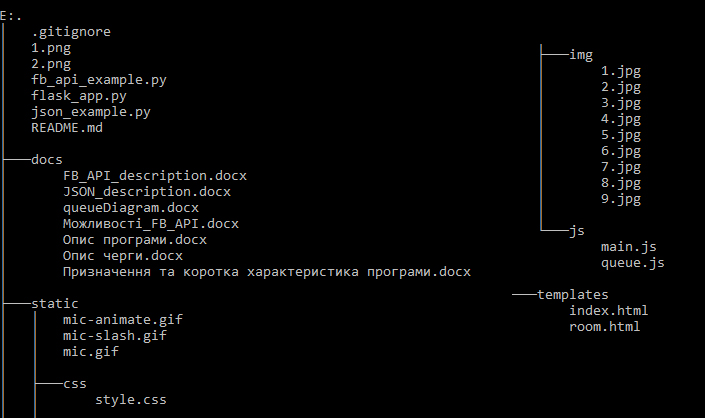
*Модулі*

* main.js
  + displayText(recognized) – дана функція приймає як аргумент recognized, стрічку, що була отримана шляхом розпізнавання голосу, модифікує її, якщо потрібно доповнити її додатковою інформацією і виводить її на екран в елемент a-text.
  + voiceToImg(keyword) – дана функція приймає аргумент keyword – стрічку, що була отримана шляхом розпізнавання голосу, знаходить картинку за даним словом та розміщує її на екрані.
  + linebreak(s) – дана функція приймає як аргумент стрічку та повертає модифіковану стрічку, де максимальний відступ рівний одному пустому рядку.
  + capitalize(s) – дана функція приймає як аргумент стрічку та повертає модифіковану стрічку, що починається буквою верхнього реєстру, та використовується для Speech Recognition.
* queue.js

Queue() – клас, котрий реалізовує структуру даних “Черга”.

* + Атрибути:
    - queue – список, в котрий будуть додані стрічки, розпізнані програмою
    - i – індекс, число типу integer
  + Методи:
    - enqueue(item) – додавання елемента в чергу, повертає ‘undefined’; item - стрічка
    - dequeue() – видалення елемента з черги, повертає ‘undefined’
    - length() – знаходження довжини черги, повертає число типу integer
    - isEmpty() – перевірка чи в черзі наявні елементи, повертає True, якщо черга пуста і False у іншому випадку.
    - peek() – знаходження початку черги, повертає перший елемент, якщо черга не пуста і NULL в протилежному випадку

Серверна частина програми “VR Ideas” написана на Python та розміщена на pythonanywhere, а частина, що відповідає за інтеракцію з користувачем написана на JavaScript та використовує відповідну стандартну структуру. Проект складається з папок

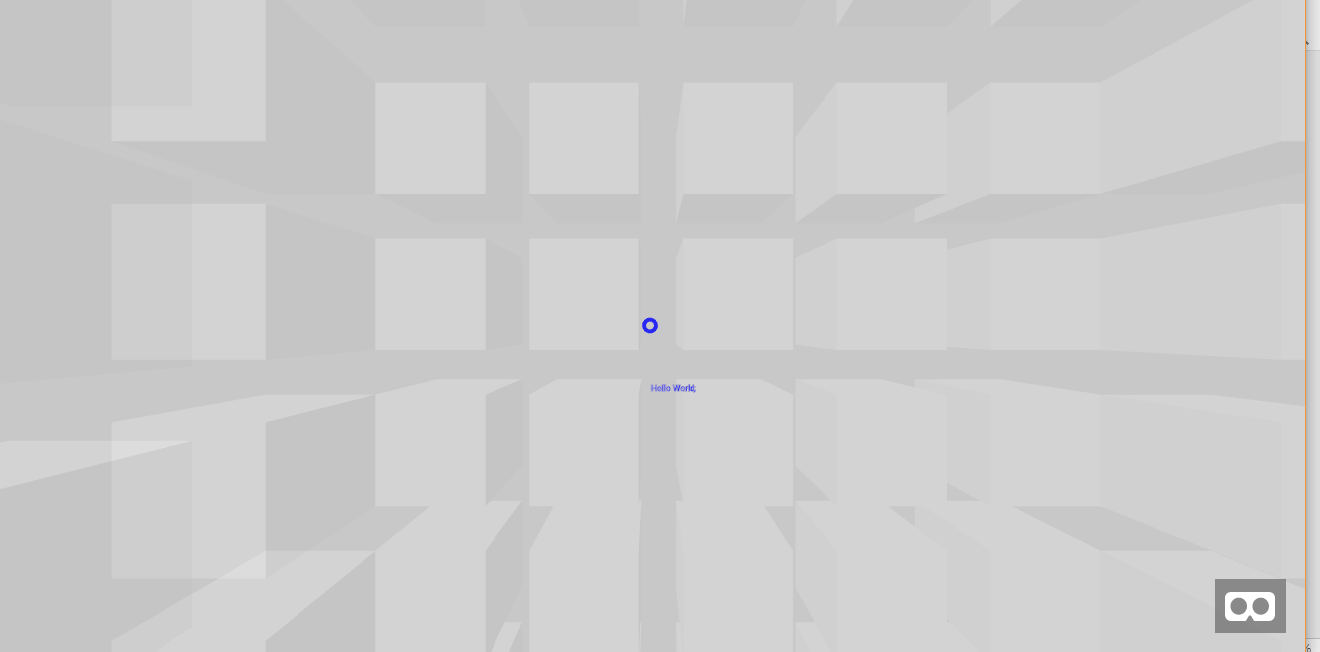


**Коротка інструкція по користуванню програмою**

1. Користувач заходить на веб додаток <http://vrhome.pythonanywhere.com/>
2. Перше, чим він користується – курсор та камера. Якщо користувач користується комп’ютером, тоді він керує курсором вручну, якщо мобільним телефоном – тоді курсор переміщається одночасно з рухом голови.
3. Далі, він наводить курсор на один з кубів, котрий починає підсвічуватись та виступає вперед.
4. Наступним кроком він каже предмет, або те, що хоче візуалізувати.

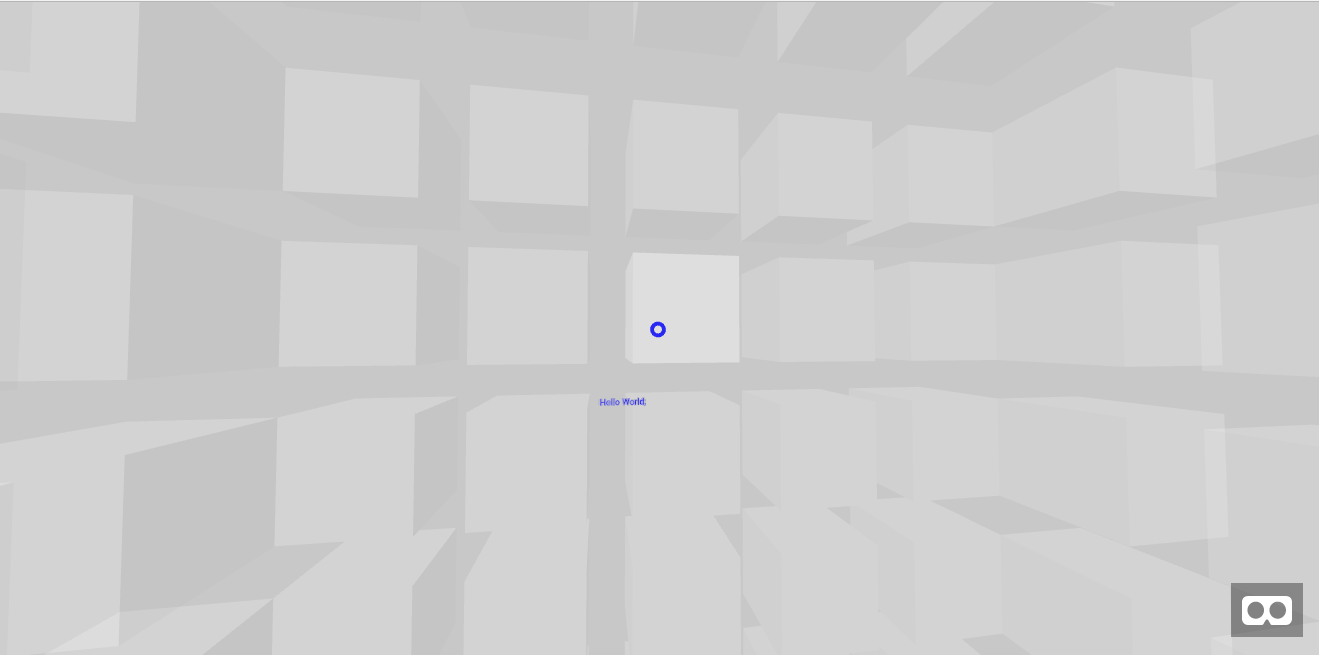
**Опис тестових прикладів для перевірки працездатності програми**

Нехай користувач хоче побачити яблуко. Для цього він заходить на <http://vrhome.pythonanywhere.com/> і бачить початкову сторінку (зображення 1).



***Зображення 1***

Переводячи курсор, він обирає місце, куди хоче помістити яблуко (нехай, обраний куб на зображенні 2) і зупиняється на ньому. Обраний куб підсвічується і виступає вперед.



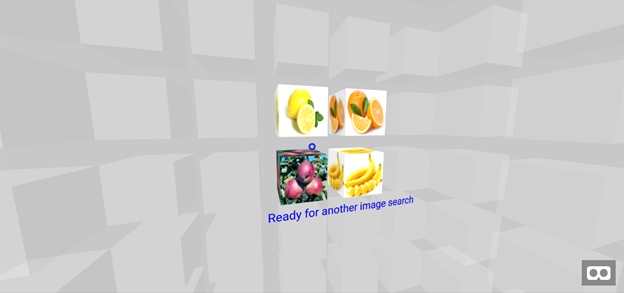
***Зображення 2***

Наступним кроком він каже “Apple”. Запит успішно обробляється і на кубі з’являється яблуко (зображення 3).



***Зображення 3***

Якщо ж користувач скаже таку комбінацію, як “Apple”, “Banana”, “Lemon”, “Orange”, то отримає наступний результат.

****