Documentation

Floating Words:

Floating Words is an AR translation app for Android. It allows users to point their smartphone camera at a real-world object, and the app automatically creates a 3D AR anchor at that position with the translated label of the object. This vocabulary is stored in a permanent dictionary that can be accessed for detailed information about each word. In addition, the app provides a "Flashcards" feature for revising and studying newly added vocabs. Finally, a screenshot of the real object is stored for each word, so that the user has a visual reference to the vocab in question.

Requirements:

Unity Engine: Floating Words was developed on the Unity Engine, which, in addition to the AR Foundation for creating augmented reality content, supports a number of other handy APIs for creating user interfaces and builds for Android and iOS. We chose Unity Engine version 2021.3.13f1, which was the latest stable release at the beginning of the development phase.

Android: All users who want to install and use this application must have an Android phone with the version: Android 7.0 'Nougat' API level 24 or higher. Other devices with iOS or Windows as operating systems are not supported by Floating Words at this time.

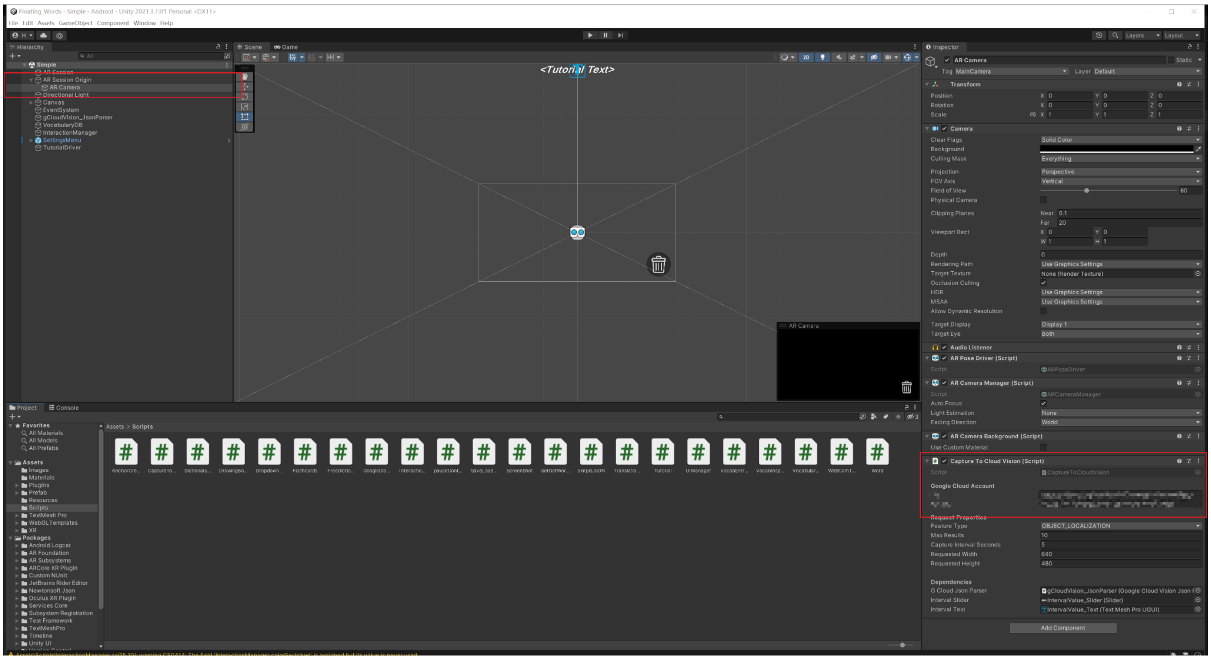
Internet connection: Furthermore, the software must call the Google Cloud Vision API for recognizing image features and the Google Translation API for translating words from the source language into the target language. So a good Internet connection is also a prerequisite for a comprehensive user experience.

Lighting conditions: Since the Vision API is solely reliant on RGB data from the smartphone camera, good lighting conditions are essential for the object recognition algorithm to produce accurate and precise results.

How to get started:

Setup Google Cloud Vision API:

- To use the Vision API, you need to create an API key in your Google Cloud account. To do this, you need to enter your credit card information at Google Cloud. This will automatically grant you a free $300 credit that you can use for the machine learning API. (Setup)
- Enter the API key in Unity GameObject “AR Camera” (AR Session Origin/AR Camera/CaptureToCloudVision.cs)



AR Camera
Tag MainCamera   Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Camera
Clear Flags: Solid Color
Culling Mask: Everything
Projection: Perspective
FOV Axis: Vertical
Field of View: 60
Clipping Planes: Near 0.1, Far 20
Viewport Rect: X 0 Y 0 W 1 H 1
Depth: 0
Rendering Path: Use Graphics Settings
Target Texture: None (Render Texture)
Occlusion Culling: ✓
HDR: Use Graphics Settings
MSAA: Use Graphics Settings
Target Display: Display 1
Target Eye: Both

Audio Listener

AR Pose Driver (Script)
Script: ARPoseDriver

AR Camera Manager (Script)
Script: ARCameraManager
Auto Focus: ✓
Light Estimation: None
Facing Direction: World

AR Camera Background (Script)
Use Custom Material: □

Capture To Cloud Vision (Script)
Script: CaptureToCloudVision
Google Cloud Account
[redacted]

Request Properties
Feature Type: OBJECT_LOCALIZATION
Max Results: 10
Capture Interval Seconds: 5
Requested Width: 640
Requested Height: 480

Dependencies
G Cloud Json Parser: gCloudVision_JsonParser (Google Cloud Vision Json
Interval Slider: IntervalValue_Slider (Slider)
Interval Text: IntervalValue_Text (Text Mesh Pro UGUI)

Set Vision API key in Unity

How to set up Google Cloud Translation API:

The Google Cloud Translation API can be used without an account or API key as long as a limited number of requests are made to the API. To unlock even more requests, you can create an account. For this app, the version without an account has proven to be sufficient, so you don't need to do anything for it.

Unity:

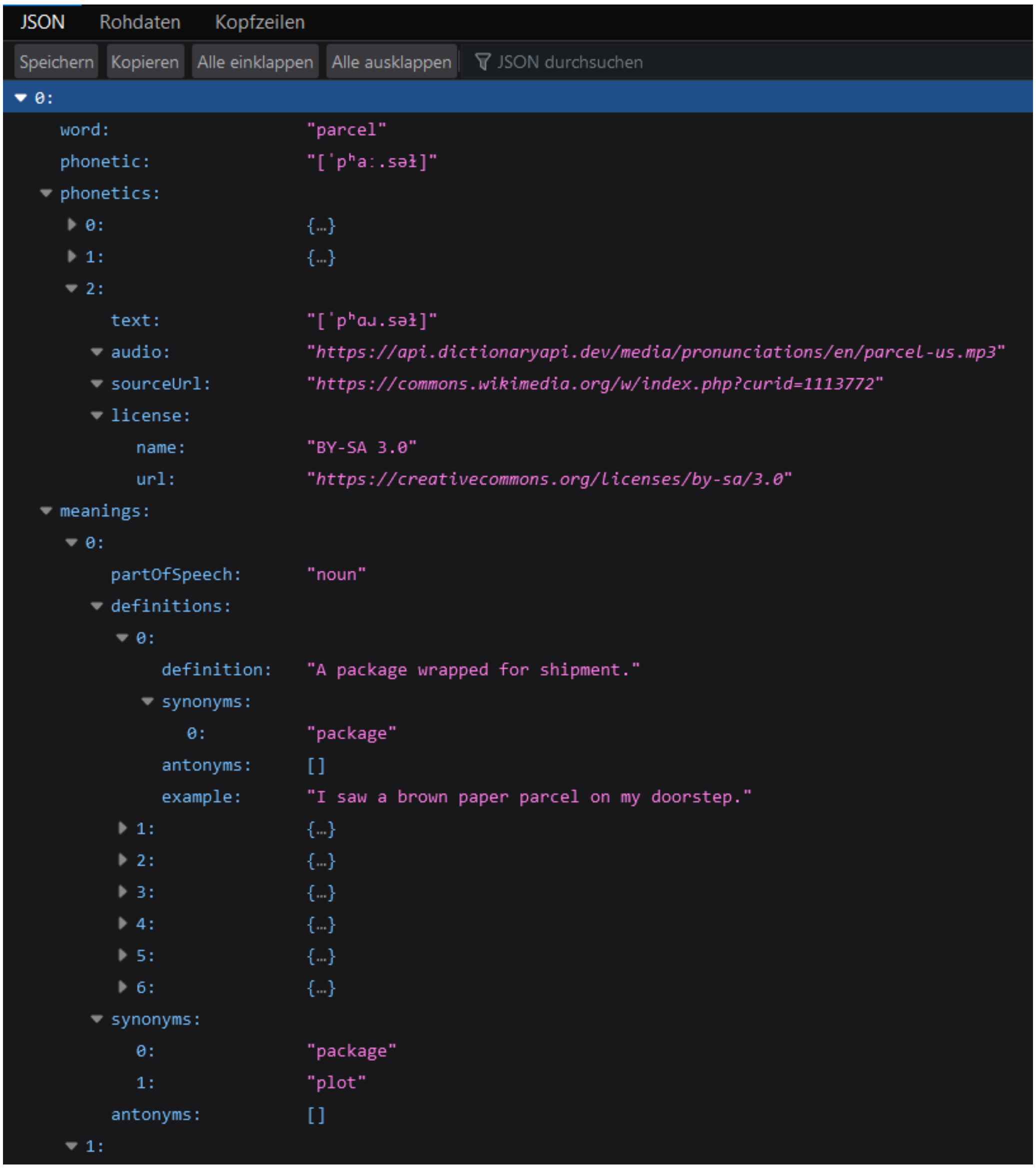
If the Google Cloud Vision API key is referenced in the CaptureToCloudVision component, you can now run the application in the editor.

Note: The camera will not open on PC since AR Foundation requires AR Core for the camera feed. To see the camera feed and test the AR functionalities you need to build the APK and run it on your Android device.

Components:

Free Dictionary API:

In addition to the Google Translation API we use the free and open source dictionary API “Free Dictionary API”. Googles Translation API is only used for translating the english word into multiple languages but does not provide more information about a word. For this, we use the dictionary API to fetch additional information about a vocab from the internet by creating a web request. This includes details like the definition, part of speech (noun, vocab, adjective), synonyms, example sentences, etc. For some words, it will also provide a URL to an audio clip of the English pronunciation of the word. The API does not require an account or key to access this information.



```
▼ 0:
    word:                "parcel"
    phonetic:            "[ˈpʰaː.səɫ]"
  ▼ phonetics:
    ▶ 0:                 {…}
    ▶ 1:                 {…}
    ▼ 2:
        text:            "[ˈpʰɑɹ.səɫ]"
      ▼ audio:           "https://api.dictionaryapi.dev/media/pronunciations/en/parcel-us.mp3"
      ▼ sourceUrl:       "https://commons.wikimedia.org/w/index.php?curid=1113772"
      ▼ license:
          name:          "BY-SA 3.0"
          url:           "https://creativecommons.org/licenses/by-sa/3.0"
  ▼ meanings:
    ▼ 0:
        partOfSpeech:    "noun"
      ▼ definitions:
        ▼ 0:
            definition:  "A package wrapped for shipment."
          ▼ synonyms:
              0:         "package"
            antonyms:    []
            example:     "I saw a brown paper parcel on my doorstep."
        ▶ 1:             {…}
        ▶ 2:             {…}
        ▶ 3:             {…}
        ▶ 4:             {…}
        ▶ 5:             {…}
        ▶ 6:             {…}
      ▼ synonyms:
          0:             "package"
          1:             "plot"
        antonyms:        []
    ▼ 1:
```

Sample JSON response for the word “parcel”

UI components:

In the following the key UI components and their functionalities are briefly outlined.

Main Scene:

In this scene, users see their AR camera feed. Here all the objects which were detected by the Vision API get a label that is attached to a 3D anchor. The AR Foundation anchors are attached to “feature points” (visualized as blue dots) and “planes” (not visualized). Note: For the anchoring to work the system first needs to identify feature points and planes to attach the anchors. These are detected when the user moves his camera slowly around the objects to be detected.

Dictionary UI:

The *Dictionary UI* window consists of a scrollable view that lists all the vocabs contained in the dictionary database as clickable buttons. They are automatically updated when the window is opened and can be clicked on to open the *Vocab Inspector UI* described in the next part.

Vocab Inspector UI:

The *Vocab Inspector UI* functions as an inspector window for a specific word. It is opened by clicking on a vocab entry in the *Dictionary UI* and reveals additional information about the vocab stored, such as the translation (which depends on the currently selected language), definition, or part of speech. In addition, it includes an interactable speaker icon that when clicked on plays an audio clip of the English vocab. This button will only appear if the active word contains an audioClipURL (The Free Dictionary API does not provide an audio clip for every English word). For now, the audio clip is always the English pronunciation but should be changed to the respective pronunciation of the translated word in later works. Lastly, the window features a screenshot button which opens the *Screenshot UI* window.

Screenshot UI:

This window will load and display an image (screenshot) of the real word object linked to the active vocab. This screenshot is captured once whenever a new word is added to the dictionary database. Screenshots are stored persistently in the application's data path.

Update Interval Slider:

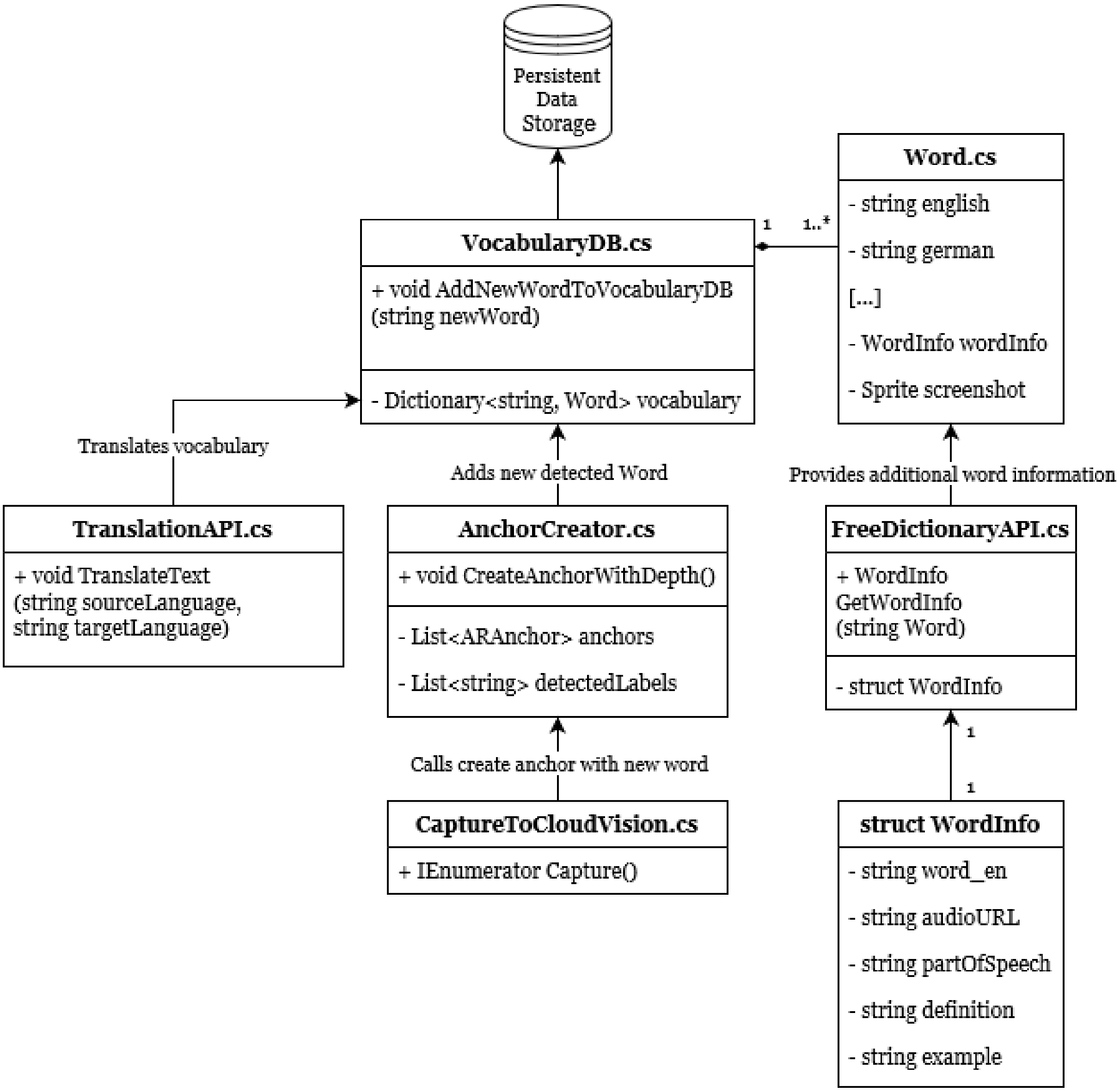
The main settings menu includes a slider to adjust the update rate of the Google Cloud Vision API requests. When the value is decreased (lower interval) the program will send more frequent web requests with the current camera image to detect object labels. In essence, this means lowering the value leads to better real-time object detection but can slow down the application due to the increased amount of web requests, detected objects, and created AR anchors. The slider's value ranges from 1 to 5 seconds.

Flashcards UI:

The flashcards feature can be accessed by clicking on the respective button in the main settings menu. This view will display a random vocabulary from the dictionary and ask the user for the correct translation of the word by providing four optional answers gathered from other existing words.

Architecture:

The following diagram displays the key components of our architecture:

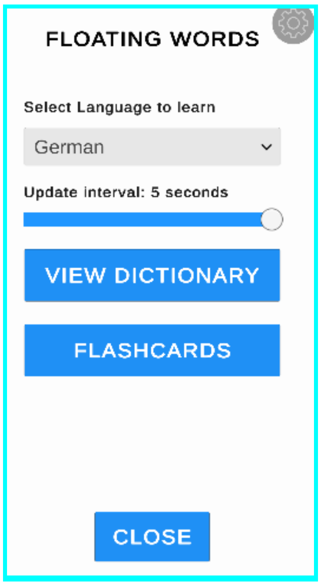


Important Links:

- Floating Words GitHub page
- Google Cloud Vision API
- Google Translation API
- Free Dictionary API
- AR Foundation

What we did:

1. Google Cloud Vision API and Google Cloud Translation API
    a. License
    b. Generating keys
2. Setup Google APIs from (1) into Unity- not directly supported- GIT project reference
    a. No bounding boxes or anchors by default in GIT project
    b. Camera orientation was not correct for Android phones - GIT project
    c. Unity and android versions
        i. 2021.3.13f1 Unity
        ii. Android Phone with Android Version: Android 7.0 'Nougat' API Level 24 or higher (required for Unity AR Foundation)
    d. Challenges with Google Cloud API
        i. Some objects/shapes are very prominent, e.g. Person, packaged goods, etc.
        ii. Needs good internet
        iii. Need good lighting
3. Needed Dictionary API (other than google) for details of words (definitions, part of speech, synonyms, etc.) - Google only labels the objects in English- https://dictionaryapi.dev/
4. UI- all UI interactions and settings are accessible using the setting menu



FLOATING WORDS

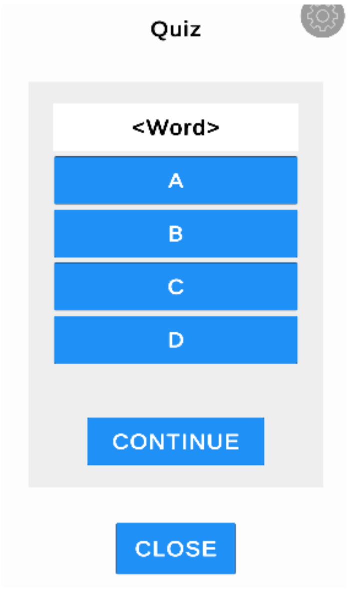
Select Language to learn

German

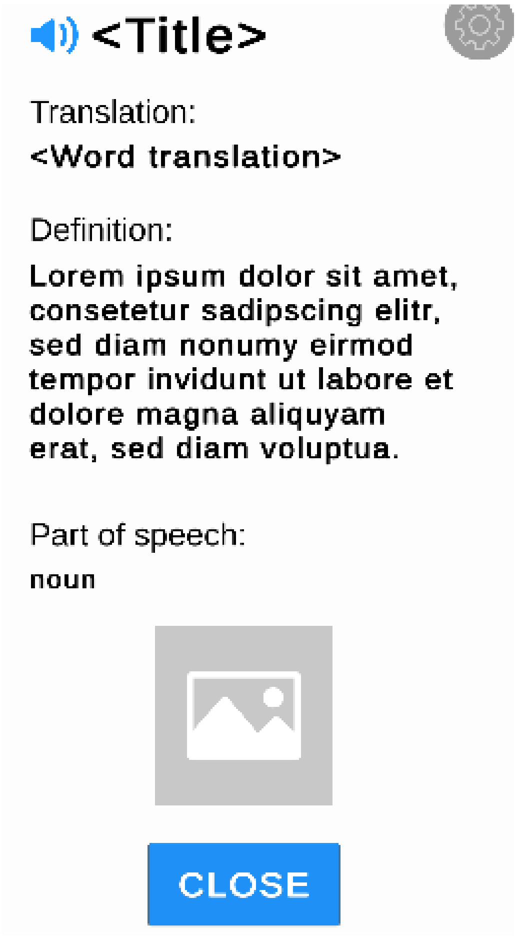
Update interval: 5 seconds

    a. <u>Language selection(drop down)</u>- User can select a preferred language to learn
        i. English (base language)
        ii. German
        iii. Chinese
        iv. Spanish
        v. Japanese

b. Flash cards- revise the learned words by asking multiple choice questions



c. Dictionary- see all the detailed of a saved word (all the detected words are saved automatically with each refresh)



Translation:
<Word translation>

Definition:
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Part of speech:
noun

i. Translation
ii. Definition
iii. Part of speech
iv. Image for better cognition
v. A close button to go back to previous menu

    d. Update interval slider- we can set the time frequency, how often we detect the objects. This is inversely proportional to refresh rate.

5. AR foundation- to use anchors instead of bounding boxes.
    a. Maybe explain why anchors used
    b. Why automating saving of words rather than manual interactive saving?
6. Logcat
7. DictionaryUI
8. VocabInspectorUI
9. ScreenshotUI
10. Flash cards
11. Tutorial Gyroscope to check the initial rotation
12. Slider can change the refresh rate of object detection
13. Doxygen for generating code documents

Fabi
Anup
Wang
Chen

UML-like diagram with most important components of the project

- CaptureToCloudVision.cs
- VocabularyDB.cs
- Word.cs
- FreeDictionaryAPI.cs
- TranslationAPI.cs

- Note: Phone needs to be moved around the environment first until feature points are detected (visualized as blue dots on the screen)