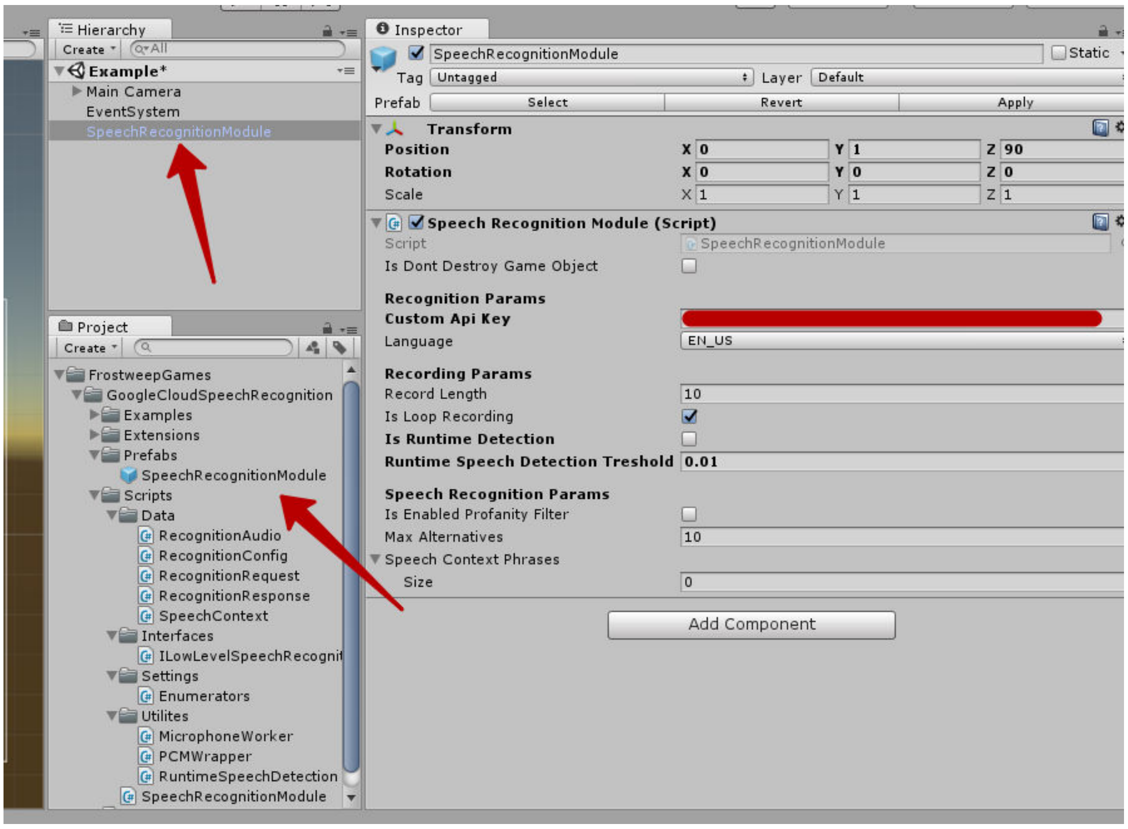
Google Cloud Speech Recognition

Hierarchy

- Interfaces:
  - ILowLevelSpeechRecognition
    - Method: `Recognize`
    - Method: `StartRecord`
    - Method: `StopRecord`
    - Event: `SpeechRecognizedSuccessEvent`
    - Event: `SpeechRecognizedFailedEvent`
- Classes:
  - SpeechRecognitionModule – implements ILowLevelSpeechRecognition
  - RuntimeSpeechDetection
  - RecognitionAudio
  - RecognitionConfig
  - RecognitionRequest
  - RecognitionResponse
  - SpeechContext
  - Enumerators
    - Enum: Language
    - Enum: AudioEncoding
  - MicrophoneWorker
  - PCMWrapper

How to use

First of All you need to add SpeechRecognitionModule prefab from FrostweepGames->GoogleCloudSpeechRecognition->Prefabs folder to your working scene.



SpeechRecognitionModule ☐ Static
Tag Untagged Layer Default
Prefab

Transform
Position X 0 Y 1 Z 90
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Speech Recognition Module (Script)
Script SpeechRecognitionModule
Is Dont Destroy Game Object ☐

Recognition Params
Custom Api Key [redacted]
Language EN_US

Recording Params
Record Length 10
Is Loop Recording ✓
Is Runtime Detection ☐
Runtime Speech Detection Treshold 0.01

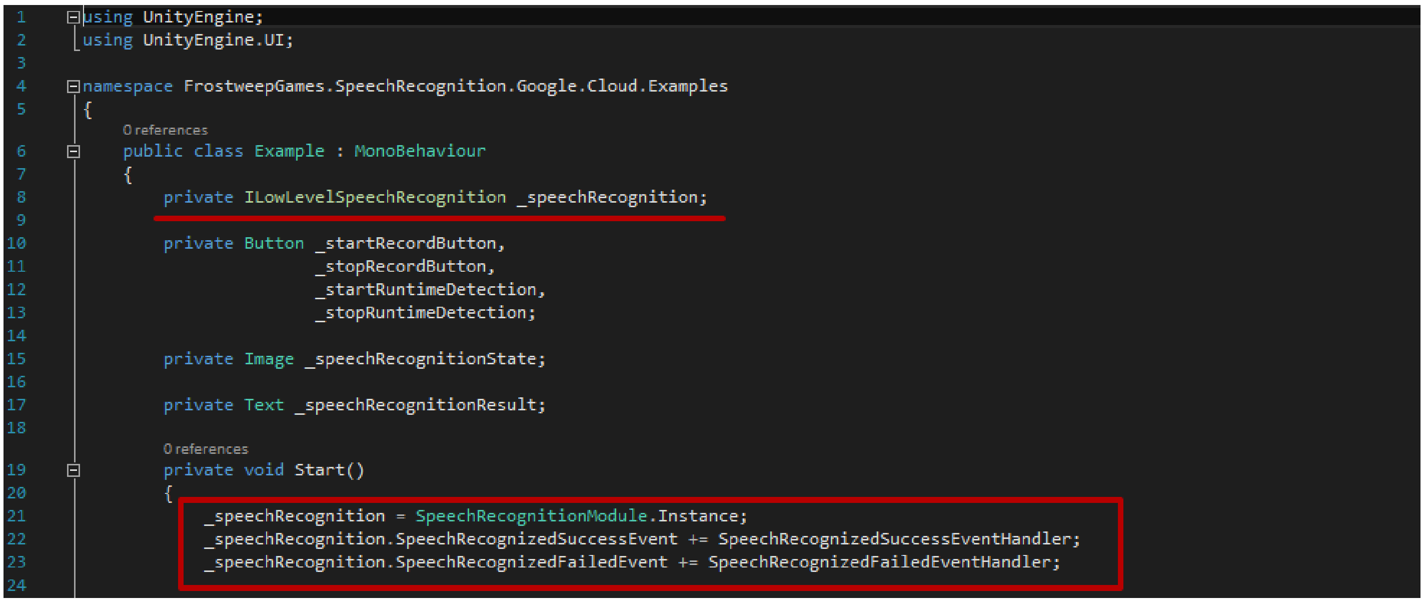
Speech Recognition Params
Is Enabled Profanity Filter ☐
Max Alternatives 10
Speech Context Phrases
Size 0

Then you need to set your own API key of Google Cloud Speech Recognition into **Custom Api Key** field. If you don’t have API Key, you can get it from https://cloud.google.com/speech/ , https://cloud.google.com/speech/docs/common/auth#restrictions

Also you can change default **Language** on what you need, Enable **Runtime Speech Detection**[0], Set runtime speech detection **treshold**, Enable **profanity filter**, Set count of **alternatives** for speech recognition result[1], Set **speech context phrases**[2].

Record Length field uses for how long we record speech**. Is Loop Recording** field sets the loop of recording every **Record Length** seconds**.**

Then we need to create script with name Example and write base logic:



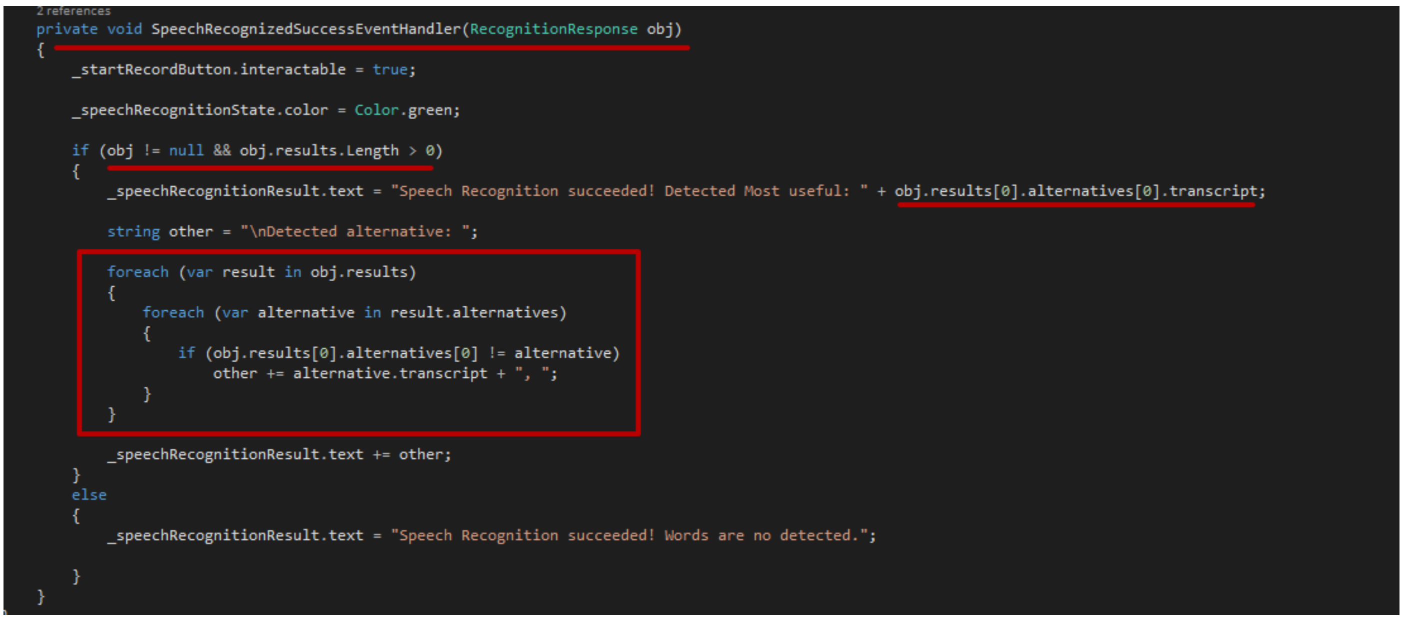
```
using UnityEngine;
using UnityEngine.UI;

namespace FrostweepGames.SpeechRecognition.Google.Cloud.Examples
{
    0 references
    public class Example : MonoBehaviour
    {
        private ILowLevelSpeechRecognition _speechRecognition;

        private Button _startRecordButton,
                       _stopRecordButton,
                       _startRuntimeDetection,
                       _stopRuntimeDetection;

        private Image _speechRecognitionState;

        private Text _speechRecognitionResult;

        0 references
        private void Start()
        {
            _speechRecognition = SpeechRecognitionModule.Instance;
            _speechRecognition.SpeechRecognizedSuccessEvent += SpeechRecognizedSuccessEventHandler;
            _speechRecognition.SpeechRecognizedFailedEvent += SpeechRecognizedFailedEventHandler;

```

Where **SpeechRecognizedSuccessEventHandler** is the event handler of **SpeechRecognizedSuccessEvent** and **SpeechRecognizedFailedEventHandler** is the event handler of **SpeechRecognizedFailedEvent**.

SpeechRecoginezedSuccessEvent will fire when speech recognition will returned response. This event has a **SpeechRecognitionResponse** param type.

SpeechRecoginezedFailedEvent will fire when speech recognition failed. This event has a **string** param type.

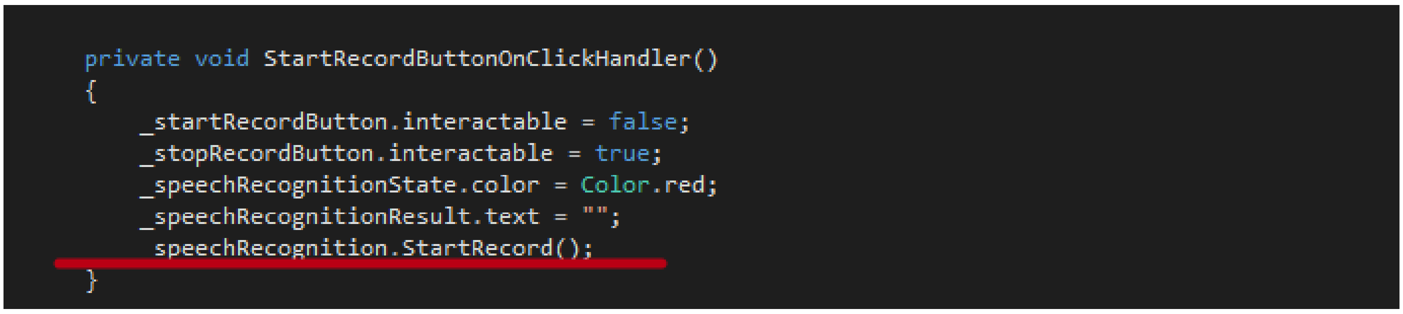
You can handle response of Speech Recognition in **SpeechRecognizedSuccessEventHandler**



```
2 references
private void SpeechRecognizedSuccessEventHandler(RecognitionResponse obj)
{
    _startRecordButton.interactable = true;

    _speechRecognitionState.color = Color.green;

    if (obj != null && obj.results.Length > 0)
    {
        _speechRecognitionResult.text = "Speech Recognition succeeded! Detected Most useful: " + obj.results[0].alternatives[0].transcript;

        string other = "\nDetected alternative: ";

        foreach (var result in obj.results)
        {
            foreach (var alternative in result.alternatives)
            {
                if (obj.results[0].alternatives[0] != alternative)
                    other += alternative.transcript + ", ";
            }
        }

        _speechRecognitionResult.text += other;
    }
    else
    {
        _speechRecognitionResult.text = "Speech Recognition succeeded! Words are no detected.";

    }
}
```

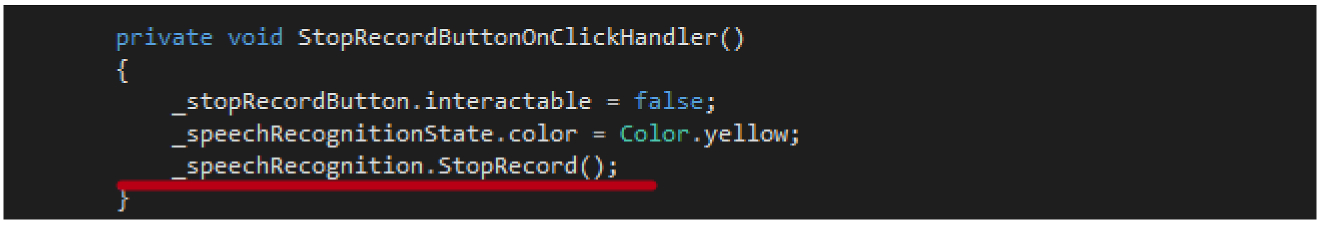
To get result of recognition you can use “RecognitionResponse->results->alternatives.transcript” path. Where RecognitionResponse is instance of RecognitionResponse object.

For the start recording you can call this method:



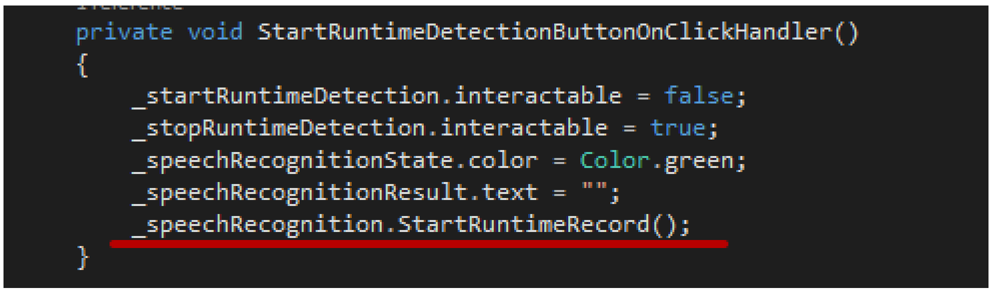
```
private void StartRecordButtonOnClickHandler()
{
    _startRecordButton.interactable = false;
    _stopRecordButton.interactable = true;
    _speechRecognitionState.color = Color.red;
    _speechRecognitionResult.text = "";
     speechRecognition.StartRecord();
}
```

For the stop recording you can call this method:



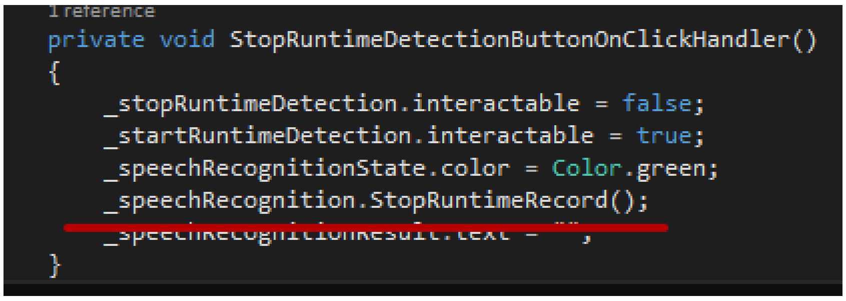
```
private void StopRecordButtonOnClickHandler()
{
    _stopRecordButton.interactable = false;
    _speechRecognitionState.color = Color.yellow;
    _speechRecognition.StopRecord();
}
```

For the start runtime speech detection and recording you can call this method:



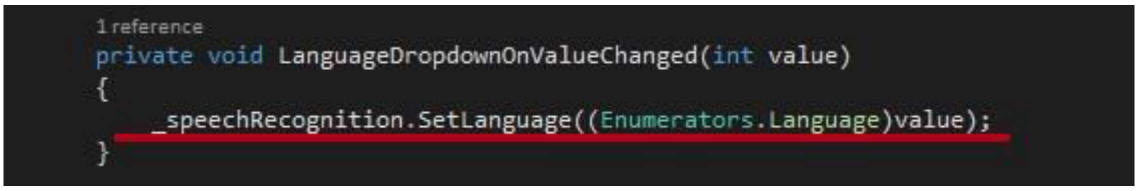
```
private void StartRuntimeDetectionButtonOnClickHandler()
{
    _startRuntimeDetection.interactable = false;
    _stopRuntimeDetection.interactable = true;
    _speechRecognitionState.color = Color.green;
    _speechRecognitionResult.text = "";
    _speechRecognition.StartRuntimeRecord();
}
```

For the stop runtime speech detection and recording you can call this method:



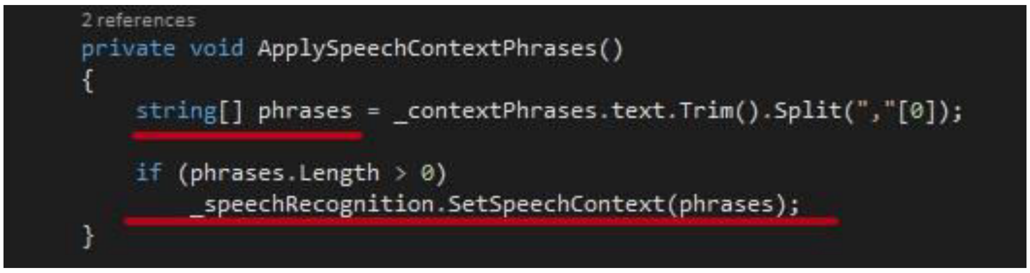
```
1 reference
private void StopRuntimeDetectionButtonOnClickHandler()
{
    _stopRuntimeDetection.interactable = false;
    _startRuntimeDetection.interactable = true;
    _speechRecognitionState.color = Color.green;
    _speechRecognition.StopRuntimeRecord();
    _speechRecognitionResult.text = "";
}
```

If you want to set language you can call this method (where value is integer converted to Language enum):



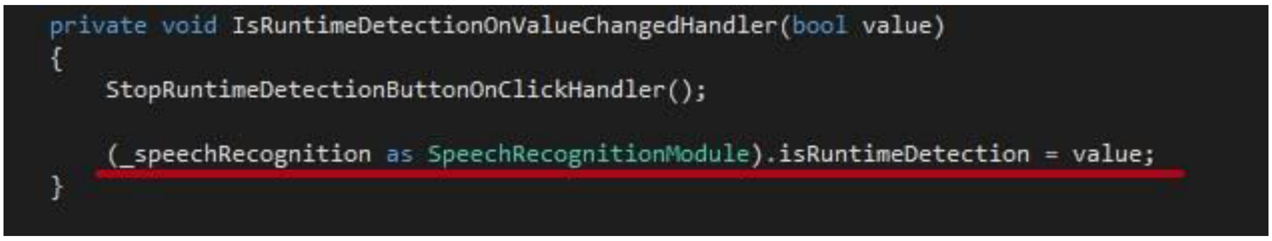
```
1 reference
private void LanguageDropdownOnValueChanged(int value)
{
    _speechRecognition.SetLanguage((Enumerators.Language)value);
}
```

If you want to set speech context you can call this method (where arg0 is string array):



```
2 references
private void ApplySpeechContextPhrases()
{
    string[] phrases = _contextPhrases.text.Trim().Split(","[0]);

    if (phrases.Length > 0)
        _speechRecognition.SetSpeechContext(phrases);
}
```

If you want to enable or disable runtime speech detection you can change this field (can be true or false):



```
private void IsRuntimeDetectionOnValueChangedHandler(bool value)
{
    StopRuntimeDetectionButtonOnClickHandler();

    (_speechRecognition as SpeechRecognitionModule).isRuntimeDetection = value;
}
```

Example scene included to project:

FrostweepGames-> GoogleCloudSpeechRecognition->Examples

[0] – Enable runtime speech detection and disable solitary speech recording

[1] – Count of alternative words range: 1 - 30

[2] – An array of phrases in context

Note

- Example script included in unitypackage!
- Working with il2cpp and mono
- Supported all platforms*
- Plugin Support Unity3D 4 or above

* - Plugin doesn't support WebPlayer.
  - On WebGL Unity engine doesn't support "Microphone" the class

Version Updates

• 2.1

- implemented new features
- updated and improved example
- removed 3rd party libraries

• 2.0

- UPDATED Speech Recognition API to the latest Google Cloud Speech API
- implemented new features
- implemented speech detection threshold
- changed namespaces
- fixed bugs

• 1.1

- Changed Code Namespace with FrostweepGames.SpeechRecognition on FrostweepGames.SpeechRecognition.Google
- Implemented Runtime Speech Detection Utility