# Interactive and Immersive Multimedia Systems

Course 2018/2019

Names and surnames:

Kamil Fundakowski
Szymon Świdrak

Title of project

Voice based interaction with websites using Annyang

Date

April 2019

Table of contents

# Introduction

The purpose of this project is to make a site that uses the voice to provide all of the basic actions that you can perform on classic website such as:

- reloading
- handling messages
- handling events
- resizing
- scrolling
- dynamic changing  of contents
- filling forms
- clicking
- redirecting to other pages
- playing and pausing medias
- changing volume

To Improve the experience of using Websites with Voice navigation our project needs to be a conversational and two-way experience, and an ambient experience and adapt the concept of “Request and Reaction”.

Another thing that we need to place in this document is required materials and tools to make that kind of work.

# Goals

Creating a test page that will implement basic features required to comfortable usage.

Research about possible solutions that give us opportunity to convert speech to text and vice versa.

Implementation of Annyang.js library which is javascript based and provides access to SpeechRecognitionAPI that lets developer use speech-to-text conversion and is delivered by Google

Adaptation of SpeechSynthesisAPI that allows us to give the user feedback about action that was performed on the website.

Phase of testing browsers support on usage of Annyang.js

Process of reaserch on the field of possible incompatibility of Text-to-Voice conversion provided by SpeechSynthesis which is built-in at most of the browsers

Detecting problems linked with environment such as noises or big distance between user and microphone, also there may occur difficulties with understanding not native-speakers due to issues with an accent.

# Annyang.js


Annyang is a tiny javascript library that lets your visitors control your site with voice commands. Annyang supports multiple languages, has no dependencies, weighs just 2kb and is free to use.

How does it work? It's simple. The browser will request for user permission to the microphone and after the listening is turned on, everything you say is being transmitted to a cloud where the speech-to-text magic happens. This means that the connection speed is important in order to get fast, accurate results and also this means that it can't be used offline

**Which browsers are supported?**

Annyang works with all browsers that implement the Speech Recognition interface of the Web Speech API (such as Google Chrome, and Samsung Internet).

SpeechRecognition

Update compatibility data on GitHub

| | Chrome | Edge | Firefox | Internet Explorer | Opera | Safari | Android webview | Chrome for Android | Edge Mobile | Firefox for Android | Opera for Android | Safari on iOS | Samsung Internet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SpeechRecognition | 33 -x- ★ | ? | No | No | No | No | ? | Yes -x- ★ | ? | No | No | No | ? |

Full support

No support

Compatibility unknown

Experimental. Expect behavior to change in the future.

★ See implementation notes.

-x- Requires a vendor prefix or different name for use.

As we see at the picture above most of the browsers are not supported by SpechRecognition so we were restricted to use Google Chrome.

# SpeechSynthesis

Speech synthesis is accessed via the SpeechSynthesis interface, a text-to-speech component that allows programs to read out their text content (normally via the device's default speech synthesiser.) Different voice types are represented by SpeechSynthesisVoice objects, and different parts of text that you want to be spoken are represented by

SpeechSynthesisUtterance objects. You can get these spoken by passing them to the SpeechSynthesis.speak() method.

## SpeechSynthesis

Update compatibility data on GitHub

| | Chrome | Edge | Firefox | Internet Explorer | Opera | Safari | Android webview | Chrome for Android | Edge Mobile | Firefox for Android | Opera for Android | Safari on iOS | Samsung Internet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SpeechSynthesis | 33 | Yes | 49 | No | 21 | 7 | 4.4.3 | 33 | Yes | 62 | No | 7.1 | ? |

Full support

No support

Compatibility unknown

Experimental. Expect behavior to change in the future.

User must explicitly enable this feature.

As we see above most of the browsers are supported and it is a built-in feature at major part of them.

# Code and Functionalities

```
var commands = {
  'hello': function() {
    console.log("elo")
    hello();
  },
  'email *cos': function(cos) {
    console.log("email")
    document.getElementById('email').value = cos;
  },
  'password *cos': function(cos) {
    console.log("psw")
    document.getElementById('psw').value = cos;
  },
  'repeat password *cos': function(cos) {
    console.log("psw-repeat")
    document.getElementById('psw-repeat').value = cos;
  },
  'show password': function() {
    console.log("pass show")
    document.getElementById('psw-repeat').type = "text";
    document.getElementById('psw').type = "text";
  },
  'hide password': function() {
    console.log("pass hide")
    document.getElementById('psw-repeat').type = "password";
    document.getElementById('psw').type = "password";
  },
  'tell me a joke': function() {
    console.log("joke")
    getJoke();
  },
```

Example above shows us how we define the commands that will be used by Annyang to awake the JaveScript's function that will affect our website and change the content on it or perform basic action such as scrolling down or up or redirecting to other pages. We also check here if the browser that is used to display the webpage support the annyang and if it don't we give a popup warning to the user.

To start Annyang listening we need to add an object  with key-value pairs that represent our command-code excution behavior.  Code example below:

```
annyang.addCommands(commands);

annyang.start({ autoRestart: true, continuous: false });
```

Our Annyang command can execute more complicated JS function for example we used Axios to make API call to obtain joke for our “tell me a joke” functionality.

```
function getJoke() {
  axios.get('https://api.chucknorris.io/jokes/random?category=animal')
    .then(function (response) {
      console.log(response.data.value);
      var msg = new SpeechSynthesisUtterance(response.data.value);
      window.speechSynthesis.speak(msg);
      document.getElementById('volume').style.visibility = 'visible';
      setTimeout(()=>{
        document.getElementById('volume').style.visibility ='hidden';
      },6000)
    })
    .catch(function (error) {
      console.log(error);
    })
```

We also used at the example above SpeachSynthesisUtterence function to build an speech synthesis object which was then passed to the speak function. Which allowed us to read the joke. Most of our voice feedback was given in that way.

```
function hello (){
  document.getElementById("dupa").style.visibility="hidden";
  document.getElementById("dupa2").style.visibility="hidden";
  document.getElementById("title").innerHTML = "HOLA !!";
  var msg = new SpeechSynthesisUtterance("Hola !!!");
  window.speechSynthesis.speak(msg);
  setTimeout(()=>{
    document.getElementById("dupa").style.visibility="visible";
    document.getElementById("title").innerHTML = "";
    document.getElementById("dupa2").style.visibility="visible";
  },2000);
}
```

There is an example of dynamic change of the content at our website with an voice feedback to the user. The Annyang command to run this function was “Hello”.

# How You can use this website

At the webpage we automatically start the Annyang to listen for the commands. We wanted to make a website that can be fully manged only by voice without need of use mouse or keyboard.

To achive that we provided set of command to navigate through the webpage such as:

1: “Scroll Down” - Scrolling page down for 250 units.
2: “Scroll up” - Scrolling page up for 250 units.
3: “Start Scrolling down” – Continues scrolling page down.

4: “Stop scrolling” – Command to stop continues scrolling .
5: “Start scrolling up” – Continues scrolling up.
6: “Go to function” – Redirect to features page.
7: “Go to homepage” – Redirect to the Home page.
8: “Go to list” – Redirect to list of other libraries that are used for same purposes.
9: “Go to presentation” – Redirect to our presentation page.

Also we made a custom features for features page such as:

1: “Hello” – Changes content and provide voice feedback.
2: “Tell me a joke” – Sending request to API to obtain the joke then read it for the user.
3: “Find *something” – Give the user option to “Google” something after you say “Find” Annyang take the next words as a parameter and the loads you google page with the results of searching.
4: “Email *email” – Command use to the fill the email input
5: “ Password *” - Command used to fill the password input
6: “ Repeat password *” – Command used to fill repeat password input.
7: “Show password” – Provides an option to check if the password input was filled properly.
8: “Hide password” – Hides the password input.

# Problems

During development of this project we encounter some smaller or bigger problems, some of them were connected with differences of human speech, other with changing surroundings, but there was also some unexpected technical ones.

First things first – when you speak foreign language is of normal that your accent may differ from native speakers of this language, even they often say words in few ways. Relating to our app, we had troubles when we were talking to microphones because app sometimes wouldn’t recognize commands. Going next, for example in personal computer or laptop there is only one max two microphones that aren’t best quality so every noise or echo makes Annyang much more difficult to recognize words correctly.

When talking about new technologies it is impossible not to mention compatibility and device support. As we shown above, not many browsers let user use this technology, but I hope it will change in few next months maybe years. From the other hand there is biggest group of devices that are mobile devices – unfortunately we couldn’t make Annyang and SpeechSynthesis work at the same time there.

And last problem about this is that it needs continues and high speed Internet connection, so due to this, apps can not become only Voice-controlled.

# Live demo

To try this yourself you can visit our dedicated website. Here are few steps that have to be done before

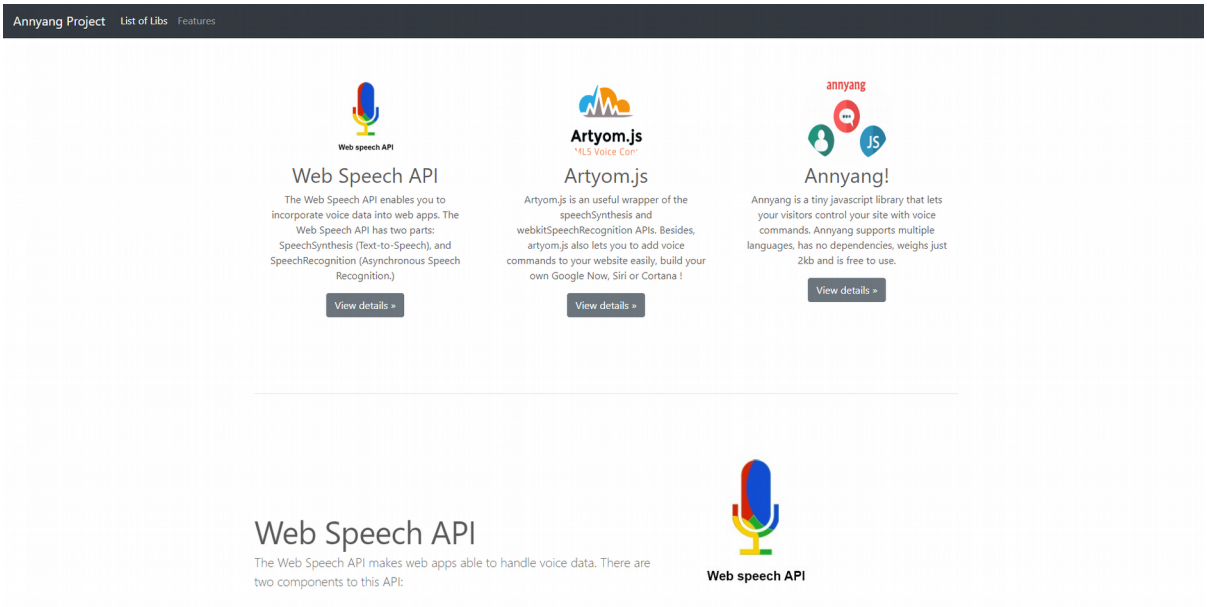
## 1: You need to use Chrome.

## 2: Go to chrome://flags/

## 3: Enable Autoplay policy

## 4: Enter https://no-hablo-espaniol.herokuapp.com/

## 5: Allow website to use Your microphone



# Bibliography

Annyang documentation on github - https://github.com/TalAter/annyang/blob/master/docs/README.md

Annayng main page - https://www.talater.com/annyang/

JavaScript documentation provided by w3schools - https://www.w3schools.com/

Bootstrap - https://getbootstrap.com/

Speech Synthesis doc - https://developers.google.com/web/updates/2014/01/Web-apps-that-talk-Introduction-to-the-Speech-Synthesis-API