

# WIFIBOT

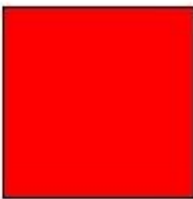
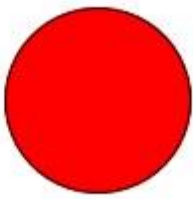
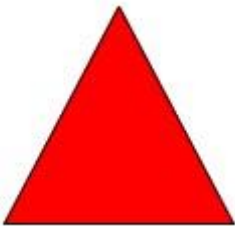
## Activité complémentaire : Synthèse vocale

### 1. Introduction

Le but aujourd'hui est de mettre en œuvre une reconnaissance vocale ceci afin de contrôler les déplacements du Wifibot.

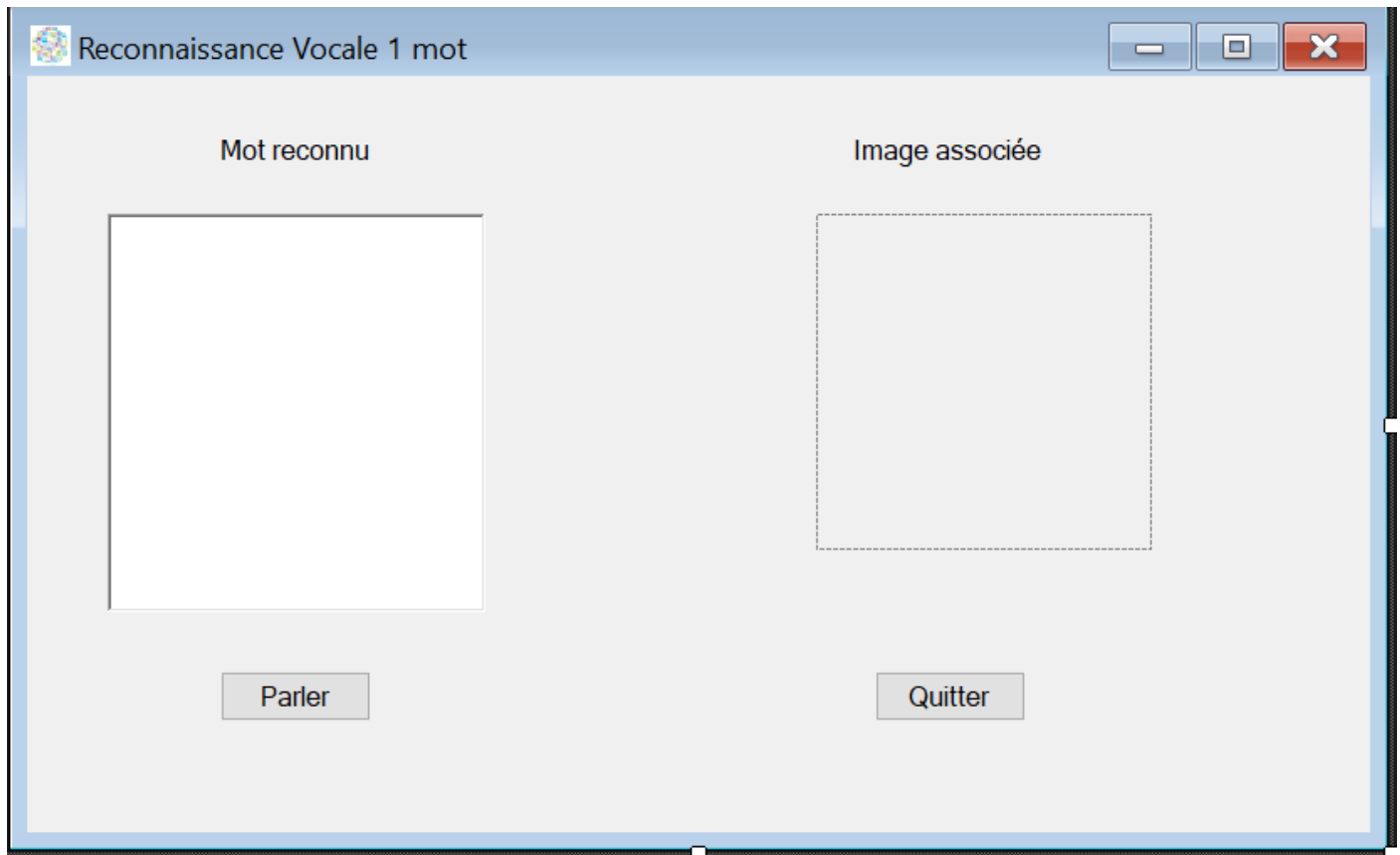
Pour y parvenir nous utiliserons le moteur de reconnaissance vocale de Windows ce moteur est utilisé dans l'assistant vocal de Windows « Cortana »

Afin de vous préparer à l'utilisation de ce composant nous allons traiter un exemple, cet exemple permet de reconnaître un mot et d'y associer l'image correspondante.

Mot reconnu	Image associée
Carré	
Cercle	
Triangle	

## 2. IHM utilisée

Pour traiter cet exemple nous réaliserons un projet Windows Form :



- L'IHM comporte une richTextBox que nous nommerons **richTextBoxReco**
- Un bouton « Parler » que nous nommerons **buttonParler**, celui-ci permet de démarrer la reconnaissance vocale
- Une pictureBox destinée à recevoir l'image associée au texte reconnu, nous la nommerons **pictureBoxAssociee**
- Un bouton « Quitter » qui permet de clore l'application, nous le nommerons **buttonQuitter**

## 3. Code Behind

Nous allons définir un certain nombre d'attributs dans la classe Form1, ils vont nous permettre d'utiliser la synthèse vocale :

```
private SpeechRecognitionEngine moteurReconnaissance;  
private Choices formeChoisie;  
private GrammarBuilder contraintesReconnaissance;  
private Grammar motsAReconnaitre;  
  
private Bitmap imageAssociee;
```

- **SpeechRecognitionEngine** : C'est le moteur de reconnaissance vocale
- **Choices** : C'est la liste des mots à reconnaître
- **GrammarBuilder** et **Grammar**: Ce sont les règles de reconnaissance des mots

Pour pouvoir utiliser la reconnaissance vocale il convient d'ajouter la référence **System.Speech** à votre projet

De même qu'il sera nécessaire d'ajouter les "using" suivants : **System.IO** et **System.Speech.Recognition**

### Le constructeur de Form1 :

```
public Form1()
{
    InitializeComponent();
    // instanciation d'une reconnaissance vocale
    moteurReconnaissance = new SpeechRecognitionEngine();
    //On precise que l'acquisition se fera sur le canal d'entrée audio par défaut (micro)
    moteurReconnaissance.SetInputToDefaultAudioDevice();
    // On construit le dictionnaire des mots à reconnaître, ceux qui ne figurent pas dans cette liste ne seront pas reconnus
    formeChoisie = new Choices(new string[] { "carré", "cercle", "triangle" });
    // On implante le dictionnaire dans le moteur de reconnaissance en utilisant un GrammarBuilder
    contraintesReconnaissance = new GrammarBuilder(formeChoisie);
    motsAREconnaître = new Grammar(contraintesReconnaissance);
    moteurReconnaissance.LoadGrammarAsync(motsAREconnaître);

    // Abonnement aux événements liés à la reconnaissance vocale
    // Evénement déclencher lorsqu'un mot est reconnu
    moteurReconnaissance.SpeechRecognized += MoteurReconnaissance_SpeechRecognized;
    // Evénement déclencher lorsqu'un mot n'est pas reconnu
    moteurReconnaissance.SpeechRecognitionRejected += MoteurReconnaissance_SpeechRecognitionRejected;
}
```

Le constructeur instancie le moteur de reconnaissance vocale, initialise la liste des mots à reconnaître et implante le dictionnaire des mots à reconnaître dans le moteur de reconnaissance vocale.

Le moteur de reconnaissance vocale génère un certain nombre d'événements lors de son utilisation, ici ce sont les événements **SpeechRecognized** (conversation reconnue) et **SpeechRecognitionRejected** (conversation rejetée) qui nous intéressent. En effet lorsqu'un mot sera reconnu nous réaliserons une action, ici nous afficherons l'image correspondant au mot reconnu (carré, cercle ou triangle). L'autre cas sera utilisé pour informer l'utilisateur que le mot n'a pas été reconnu, nous utiliserons ici une message box pour afficher un message.

Pour générer les méthodes associées aux événements il suffit de saisir :

`moteurReconnaissance.SpeechRecognized +=` puis d'appuyer deux fois sur Tab, cela généré automatiquement la méthode qui sera associée à l'événement.

Vous procéderez de la même manière pour l'événement `moteurReconnaissance.SpeechRecognitionRejected`.

## Les méthodes associées aux événements :

Si le mot n'est pas reconnu on informe l'utilisateur

```
1 référence
private void MoteurReconnaissance_SpeechRecognitionRejected(object sender, SpeechRecognitionRejectedEventArgs e)
{
    //Si le mot n'est pas reconnu on informe l'utilisateur
    MessageBox.Show("Mot non reconnu !");
}
```

Si le mot est reconnu on doit afficher l'image correspondante

```
1 référence
private void MoteurReconnaissance_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
    // Si le mot est reconnu on affiche l'img correspondant à la forme reconnue
    // et on ajoute le texte dans le richtext box prévu à cet effet
    richTextBoxReco.AppendText(e.Result.Text);
    richTextBoxReco.AppendText("\n");
    FileStream fluxImage;

    switch(e.Result.Text)
    {
        case "carré":    fluxImage = new FileStream(@"..\images\carre.jpg", FileMode.Open);
                        pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                        fluxImage.Close();
                        break;

        case "cercle":   fluxImage = new FileStream(@"..\images\cercle.jpg", FileMode.Open);
                        pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                        fluxImage.Close();
                        break;

        case "triangle": fluxImage = new FileStream(@"..\images\triangle.jpg", FileMode.Open);
                        pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                        fluxImage.Close();
                        break;

        default: break;
    }
}
```

**Attention** à bien définir le chemin d'accès aux fichiers image

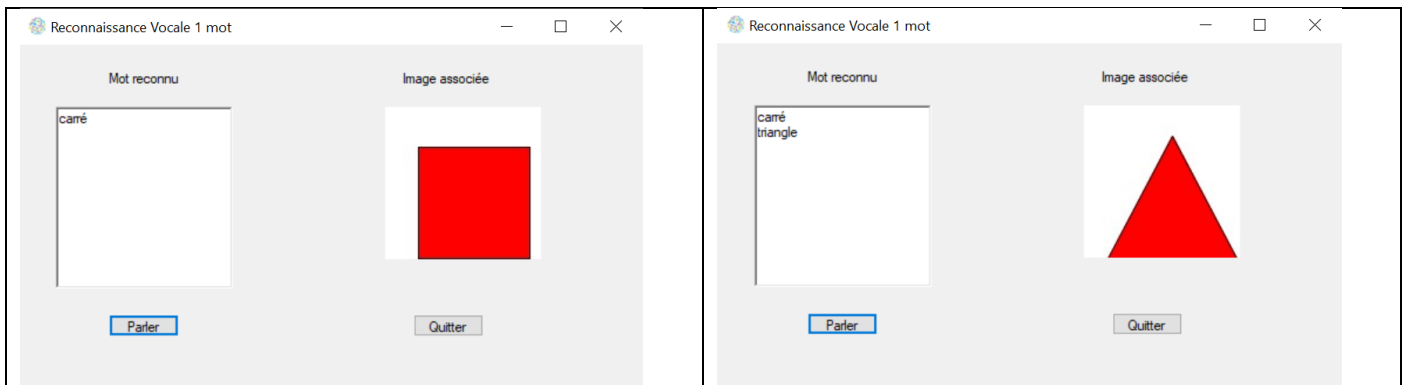
Chaque fois qu'un mot est reconnu, l'événement **speechRecognized** est déclenché et l'argument **e** contient le texte correspondant au mot reconnu. Il suffit ensuite de comparer avec les mots de la liste pour savoir quelle image on doit afficher.

Pour démarrer la reconnaissance vocale il suffit de lancer la méthode **RecognizeAsync**, ici on choisira le mode **Single**, c'est-à-dire que le moteur procède à la reconnaissance d'un mot et pas plus.

```
1 référence
private void buttonParler_Click(object sender, EventArgs e)
{
    //on ne reconnaît qu'un mot à la fois
    moteurReconnaissance.RecognizeAsync(RecognizeMode.Single);
}
```

Le bouton "Quitter" permet de fermer l'application

## 4. Un exemple de résultat.

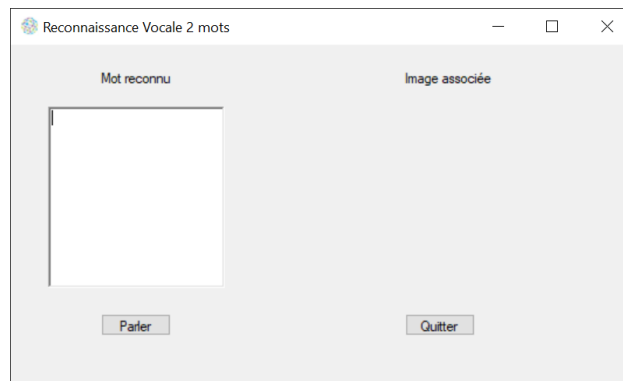


## 5. Reconnaissance de plusieurs mots

Il peut parfois être nécessaire de reconnaître et d'associer plusieurs mots. Dans cet exemple nous allons toujours reconnaître nos formes géométriques mais nous allons également leur associer une couleur. Voyons comment cela modifie la solution.

**Je ne saurai que trop vous conseiller de créer un nouveau projet en reprenant les bases du précédent.**

L'IHM est inchangée hormis le titre.



**Modifications à apporter:**

Un attribut couleur de type Choices est nécessaire en plus de ceux déjà définis dans la solution précédente.

`private Choices couleurChoisie;`

Dans le constructeur:

```
1 référence
public Form1()
{
    InitializeComponent();
    // instantiation d'une reconnaissance vocale
    moteurReconnaissance = new SpeechRecognitionEngine();
    // On précise que l'acquisition se fera sur le canal d'entrée audio par défaut (micro)
    moteurReconnaissance.SetInputToDefaultAudioDevice();
    // On construit le dictionnaire des mots à reconnaître, ceux qui ne figurent pas dans cette liste ne seront pas reconnus
    formeChoisie = new Choices(new string[] { "carré", "cercle", "triangle" });
    couleurChoisie = new Choices(new string[] { "rouge", "vert", "bleu", "jaune" });
    // On implante le dictionnaire dans le moteur de reconnaissance en utilisant un GrammarBuilder
    contraintesReconnaissance = new GrammarBuilder(formeChoisie);
    contraintesReconnaissance.Append(new SemanticResultKey("couleur", couleurChoisie));
    motsAreconnaitre = new Grammar(contraintesReconnaissance);
    moteurReconnaissance.LoadGrammarAsync(motsAreconnaitre);

    // Abonnement aux événements liés à la reconnaissance vocale
    // Evénement déclencher lorsqu'un mot est reconnu
    moteurReconnaissance.SpeechRecognized += MoteurReconnaissance_SpeechRecognized;
    // Evénement déclencher lorsqu'un mot n'est pas reconnu
    moteurReconnaissance.SpeechRecognitionRejected += MoteurReconnaissance_SpeechRecognitionRejected;
}
```

On ajoute une seconde liste de mots "couleurChoisie" et on ajoute cette seconde liste comme un groupe de mots supplémentaires dans le dictionnaire des mots à reconnaître :

```
contraintesReconnaissance.Append(new SemanticResultKey("couleur", couleurChoisie));
```

Notre dictionnaire comporte une seconde section de mot nommée "couleur". Pour que la reconnaissance fonctionne il faudra donc prononcer deux mots, une forme suivie d'une couleur.

Ex : carré rouge, cercle vert etc.

Pour le reste pas de changement / à la solution précédente.

## Méthode associée à la reconnaissance :

```
private void MoteurReconnaissance_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
    // Si le mot est reconnu on affiche l'image correspondant à la forme reconnue
    // et on ajoute le texte dans le richtext box prévu à cet effet
    richTextBoxReco.AppendText(e.Result.Text);
    richTextBoxReco.AppendText("\n");
    FileStream fluxImage;

    switch(e.Result.Text)
    {
        case "carré rouge": fluxImage = new FileStream(@"..\images\carre.jpg", FileMode.Open);
                           pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                           fluxImage.Close();
                           break;
        case "cercle rouge": fluxImage = new FileStream(@"..\images\cercle.jpg", FileMode.Open);
                           pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                           fluxImage.Close();
                           break;
        case "cercle bleu": fluxImage = new FileStream(@"..\images\cercle_bleu.jpg", FileMode.Open);
                           pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                           fluxImage.Close();
                           break;
        case "cercle vert": fluxImage = new FileStream(@"..\images\cercle_vert.jpg", FileMode.Open);
                           pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                           fluxImage.Close();
                           break;
        case "cercle jaune": fluxImage = new FileStream(@"..\images\cercle_jaune.jpg", FileMode.Open);
                           pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                           fluxImage.Close();
                           break;
        case "triangle rouge": fluxImage = new FileStream(@"..\images\triangle.jpg", FileMode.Open);
                              pictureBoxAssociee.Image = Image.FromStream(fluxImage);
                              fluxImage.Close();
                              break;
        default: break;
    }
}
```

La technique reste identique sauf qu'on reconnaît une association de mots au lieu d'un seul, dans cet exemple je n'ai traité qu'une partie des combinaisons possibles.

## Reconnaître en continu.

Il est tout à fait possible d'envisager une reconnaissance vocale en continu, pour cela il faut démarrer le moteur de reconnaissance vocale en mode multiple. Dans la méthode associée au click sur le bouton "parler" modifiez l'option de démarrage du moteur de reconnaissance.

```
moteurReconnaissance.RecognizeAsync(RecognizeMode.Multiple);
```

Il faut également être en capacité d'arrêter la reconnaissance, pour cela ajoutez un bouton "arrêter" auquel vous associez la commande

```
moteurReconnaissance.RecognizeAsyncCancel();
```

## Conclusion:

Nous avons vu comment utiliser la reconnaissance vocale pour réaliser une action, vous pouvez maintenant transposer cet exemple à la commande du rover "WIFIBOT".

## Fin de l'activité 5.