# Atelier d’Introduction au numérique

[1. Introduction 2](#_Toc163764545)

[2. Environnement matériel et logiciel 2](#_Toc163764546)

[3. Séances informatiques 3](#_Toc163764547)

[3.1. Séance 1 : algorithmes / labyrinthes 7](#_Toc163764548)

[3.2. Séance 2 : Bonjour le monde, open food facts 8](#_Toc163764549)

[3.3. Séance 4 : Application web (open food facts) 10](#_Toc163764550)

[3.4. Séance 5 : Application géolocalisée 10](#_Toc163764551)

[4. Déroulés détaillés 11](#_Toc163764552)

[4.1. Séance 3 : Application de traduction 12](#_Toc163764553)

[4.2. Séance 4 : Application web (open food facts) 15](#_Toc163764554)

[4.3. Séance 5 : Application géolocalisée 15](#_Toc163764555)

[5. Déroulés détaillés 16](#_Toc163764556)

[5.1. séance 2 – "bonjour le monde" 16](#_Toc163764557)

[5.1.1. Introduction avec SMS game : 16](#_Toc163764558)

[5.1.2. Application "bonjour le monde" : 17](#_Toc163764559)

[5.2. séance 3 – "application de traduction" 19](#_Toc163764560)

[5.2.1. V1 : écouter, traduire et dire dans une autre langue 19](#_Toc163764561)

[5.2.2. V2 : automatiser (tout enchainer sur un seul clic) 20](#_Toc163764562)

[5.2.3. V3 : choisir la langue à écouter et à traduire dans une liste 21](#_Toc163764563)

[5.2.4. V4 : traduction bidirectionnelle 22](#_Toc163764564)

[6. Atelier sur le son 24](#_Toc163764565)

[6.1. Principe : 24](#_Toc163764566)

[6.2. Environnement matériel et logiciel : 24](#_Toc163764567)

[6.3. Déroulé : 24](#_Toc163764568)

# Introduction

Le programme décrit ci-dessous propose une initiation au numérique et au développement d’applications mobiles, sur 5 séances d’une heure Le développement d’applications mobiles, avec 5 séances d’une heure avec :

* une introduction aux algorithmes sous forme de jeu (blockly maze),
* la réalisation d’une application ‘Hello world’ qui affiche un texte et le dit à haute voix,  
  puis qui écoute l’utilisateur pour traduire ce qu’il dit dans une autre langue,
* la réalisation d’une application qui lit le code barre d’un produit alimentaire, va chercher ses caractéristiques sur le Web et affiche son nutriscore,
* la réalisation d’une application qui affiche une carte avec la position de l’utilisateur, puis la position d’autres objets
* la réalisation ou modification d’une application de type "chasse aux pokemons"

Un atelier de physique sur le son, sur 2 heures est décrit séparément.

# Environnement matériel et logiciel

Pour l’initiation au numérique et au développement d’applications mobiles, il faut par groupe de 2 élèves (ou par élève) :

* un PC (ou mac ou chromebook) avec un navigateur (Firefox ou Chrome) et un accès à internet
* un smartphone ou tablette (plutôt Android) avec l’application "[AI2 companion](https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&hl=fr)" installée depuis le playstore et un accès Wifi à Internet, de préférence sur même réseau local que les PC.

La configuration est [décrite ici](http://onvaessayer.org/appinventor?app=config). Le bon fonctionnement est à vérifier à l’avance.

Les autres ressources sont en accès libre sur Internet :.

* [blockly maze](http://onvaessayer.org/appinventor/config/hourOfCode.php) pour l’apprentissage des premières notions d’algorithmie.
* [App Inventor](https://appinventor.mit.edu/) pour réaliser des applications mobiles, en exploitant les capteurs des smartphones, Internet et le Web. Comme Scratch, App Inventor vient du MIT, c’est son équivalent pour 12 ans et plus. App Inventor est utilisable de façon anonyme ou avec un compte Gmail comme [indiqué ici](http://onvaessayer.org/villonCycle4?app=config)

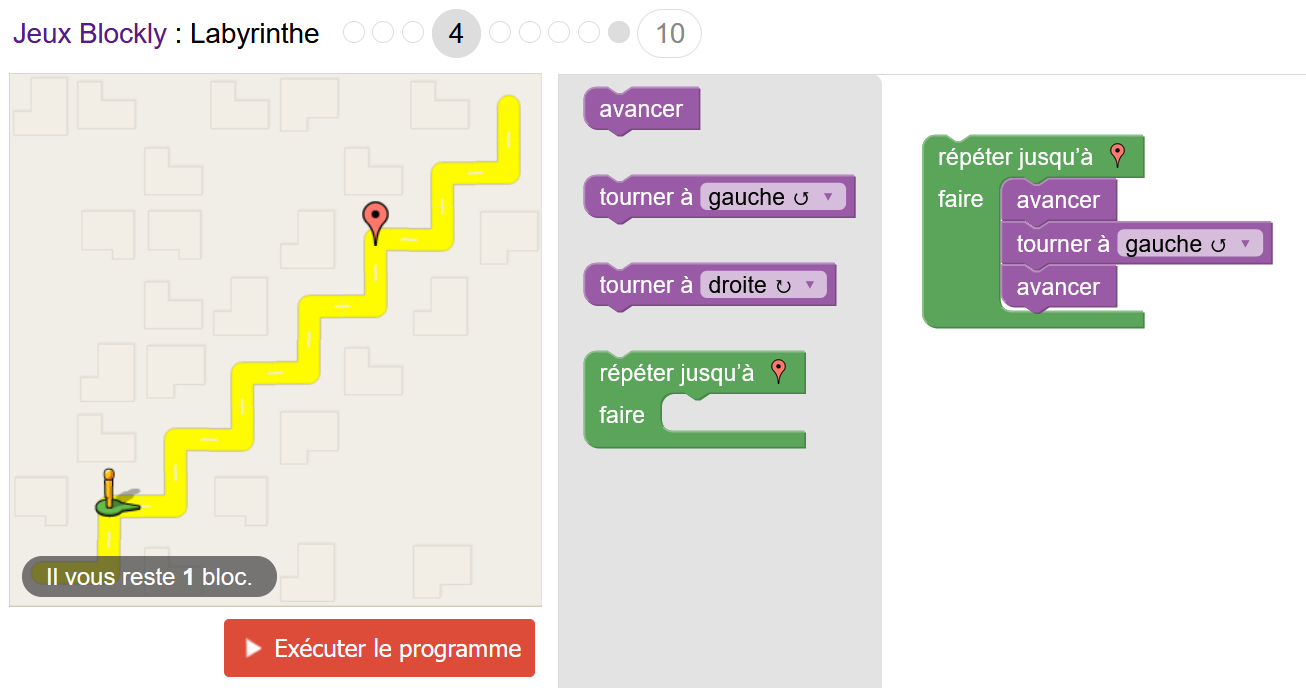
# Séances informatiques

## Revue synthétique

Le tableau ci-dessous donne une indication des activités, des durées, des notions abordées et des supports disponibles. Les vidéos résument le contenu mais on utilise les planches en classe. Les quiz peuvent être utilisés à la fin d’une séance, entre les séances ou au début de la suivante comme rappel. Ces éléments sont normalement disponibles sur à l’adresse <http://onvaessayer.org/villonCycle4>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sujet** | **temps** | **Activité** | **Notions ou points abordés** | **supports** |
| **Qu’est-ce qu’un  algorithme ?** | 5 | introduction | Ce qu’on va faire dans cette 1° séance et les suivantes | [site](http://onvaessayer.org/villonCycle4?app=hourOfCode) |
|  | 25 | Blockly, niveaux 1 à 7 ou 9 | Apprentissage par la pratique (analyse, codage, tests, séquences, itérations, conditions) | blockly |
|  | 10 | Revue : analyser, pseudo-code, traçage, … | Analyser/décomposer,  pseudo-code intermédiaire entr notre langage et celui utlisé dans les programmes |  |
|  | 10 | Définition d’un algorithme | Définition d’un algorithme : comme solution à un problème  Structuré par les séquences, boucles et branchements conditionnels L’algorithme pour faire des pâtes |  |
|  | ? | Quiz | Quiz sur les algorithmes et la lecture (traçage) de code | [Quiz](http://onvaessayer.org/villonCycle4?quiz=algorithmes) |
| **Présentation** | 10 | * rappels et démarche (avec le Quiz ?) * présentation de ce que l’on va faire | Quiz sur les algorithmes et la lecture (traçage) de code  applications mobiles, algorithmes, Internet, capteurs, géolocalisation | [Quiz](http://onvaessayer.org/villonCycle4?quiz=algorithmes) |
| **Hello World et traduction** | 15 | Hello world 1   * au démarrage, affichage d’un texte (design uniquement – pas de programme) | * lancement App Inventor * design, choix composants (ress.) * transfert smartphone, test | [video](http://onvaessayer.org/appinventor?video=jeParleLesLangues) |
|  | 15 | Hello world 2 :   * quand clic sur le bouton :   le smartphone dit le texte à haute voix | Design   * Interface / IHM : bouton * Media : composant texte à parole * nommage des composants   Blocs/programmation   * + quand clic sur bouton : appeler la fonction qui lit le texte   Notions abordées   * identification des types de blocs en fonction de   + leur couleur : évènements, propriétés, procédures (ou fonctions)   + leur forme : obtenir ou écrire (get/set) * procédures et fonctions : certaines ne renvoient pas le résultat tout de suite * évènements   + un algorithme ou un script par évènement. L’application c’est l’ensemble des scripts ou programmes exécutés en réponse à des évènements |  |
|  |  | Hello world 3 (traduction)   * quand on clique sur le bouton :   demander la traduction du texte en anglais   * puis quand la traduction arrive :   afficher la traduction et la dire à haute voix | Design   * media : traducteur /translator * Interface : zone de texte   Blocs/programmation   * initialisation d’une variable pour la langue (langueOut) * quand clic sur bouton : appel de la procédure qui traduit avec 2 paramètres   la langue dans laquelle traduire, et le texte à traduire   * quand la traduction revient   afficher le résultat de la traduction dans la zone du texte traduit  ne pas oublier de modifier la langue du locuteur (texte à parole) et appeler la procédure texte à parole avec le résultat de la traduction  Notions abordées   * variables (couleur orange) et blocs pour les lire et les mettre à jour (get/set) * chaines de caractères (couleur grenat) : get * modification par programme des propriétés d‘un composant * procédures ou foncions asynchrones : le programme n’attend pas le résultat. L’arrivée du résultat donne lieu à un nouvel évènement (quand traduction reçue) * programmation évènementielle : le programme comme série de scripts qui définissent la réaction à des évènements |  |
|  |  | Hello world 4 (reconnaissance de la parole)   * quand on clique sur le bouton Ecouter   demander au smartphone d’écouter ce que dit l’utilisateur   * quand l’application reçoit le texte reconnu   afficher le texte reconnu dans la zone de texte  et reprendre le programme précédent pour traduire et dire. | Design   * renommer le bouton : BoutonEcouter * ajouter media : reconnaissance vocale   Blocs/programmation   * quand clic sur BoutonEcouter :   + lancer la reconnaissance vocale * quand texte reconnu :   + si c’est le texte complet (partial= faux)  afficher le résultat dans la zone de texte   + puis (reprise du programme précédent) :  demander la traduction   Attention  Dans la reconnaissance vocale (nouvelle version), il ne faut pas traiter les résultats partiels, donc vérifier que le paramètre partial est faux. |  |
|  |  | Hello world 5 (choisir la langue en sortie)   * choix de la langue de traduction avec un spinner (curseur animé) | Design   * Interface :   + ajout spinner (ou curseur animé) : "SpinnerLangueOutput" et modification de la propriété "Elements " liste des langues : fr,en,es,it,de,ar,ru   Blocs/programmation   * pour l’évènement « après sélection du spinner » :   + mettre à jour la variable langue avec la sélection du spinner * Au démarrage de l’application, évènement "quand screen initialize"   + mettre à jour la sélection du spinner avec la variable langue, pour que la valeur affichée par le spinner soit – dès le départ - la même que la variable langue |  |
|  |  | Hello world 6 (choisir la langue en entrée)   * choisir la langue en entrée avec un spinner (curseur animé) * ajouter une image | Design   * Interface :   + ajouter un spinner (ou curseur animé) : SpinnerLangueInput   + ajouter une image   Blocs/programmation   * initialiser la variable langueInput : "fr" * au démarrage : quand screen1 Initialize   + copier la liste des langues (propriété Elements) de spinnerOutput  dans la liste des langues (propriété Elements de spinnerInput   + mettre à jour la propriété sélection de spinnerInput avec langueInput (pour qu’elles correspondent dès le départ) * pour l’évènement « après sélection du spinnerIput » :   + mettre à jour la variable langueInput avec la sélection du spinner * Dans le script de l’évènement "quand clic sur bouton Ecouter" :   + mettre à jour la propriété langue du composant de reconnaissance vocale avec la variable langueInput |  |
|  |  | Hello world 7 (traduction bidirectionnelle)   * quand on clique sur le bouton   demander au smartphone d’écouter ce que dit l’utilisateur et le traduire en texte   * quand l’application reçoit le texte reconnu   affichage du texte reconnu dans la zone de texte  et reprise du programme précédent (traduction et vocalisation) | Design   * interface :   + renommer "BoutonEcouter" en "BoutonDirectEcouter", mettre couleur en gris   + créer un bouton "BoutonReverseEcouter" et mettre couleur en rouge   + créer un arrangement horizontal et mettre les 2 boutons à l’intérieur   + ajouter un label entre les deux, avec un texte vide en largeur 25%   Blocs/programmation   * Dans l’évènement quand clic sur "boutonDirectEcouter", ajouter   + mettre la couleur de fond de screen1 à gris   + mettre" langueInput" à SpinnerInput.Selection   + mettre" langueOutput" à SpinnerOutput.Selection * dupliquer le script quand clic sur "boutonDirectEcouter" et le recommmer en quand clic sur "boutonReverseEcouter", et modifier son contenu (couleur et inversion input/output)   + mettre la couleur de fond de screen1 à rouge   + mettre" langueOutput" à SpinnerInput.Selection   + mettre" langueInput" à SpinnerOutput.Selection |  |
| **Open food facts**   (scan de produits au supermarché) |  | Conception et réalisation d’une application  Qu’est-ce qu’Internet ? | Web et Internet (URL)  Capteurs | [video](https://www.youtube.com/watch?v=svhZlFkFm7s) |
| **Géolocalisation et cartographie** |  | Exercices sur la géolocalisation avec un smartphone  Présentation des types de géolocalisation (GPS, Wifi, …)  Réalisation d’une application qui affiche sa position sur une carte | Géolocalisation (GPS, Wifi, …)  Serveurs de cartes (open street map) |  |
| Application géolocalisée  Pokemons |  | ~~Lecture et affichage d’une liste d’objets géo-localisés (format geoJSON)~~  ~~Édition d’objets géolocalisés sur un serveur (Git)~~  ~~Calcul de distance entre l’utilisateur et un objet~~  ~~Sélection d’objet conditionné à la distance~~ | ~~Décomposition fonctionnelle~~  ~~Représentation et abstraction de données~~  ~~Utilisation d’un serveur pour partager des données~~  ~~Édition de données~~  ~~Calculs de distance~~  ~~Utilisation d’un serveur (Git)~~ |  |

## Séance 1 : algorithmes / labyrinthes

[](https://blockly.games/maze?lang=fr&level=1&&skin=0)  
[cliquer sur l’image pour lancer](https://blockly.games/maze?lang=fr&level=1&&skin=0)

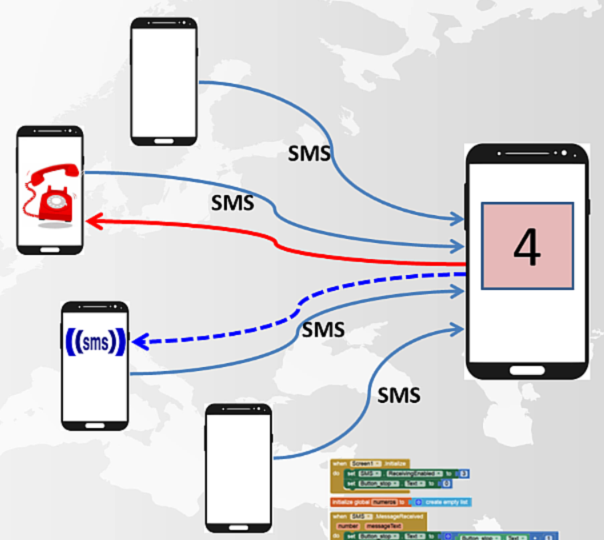
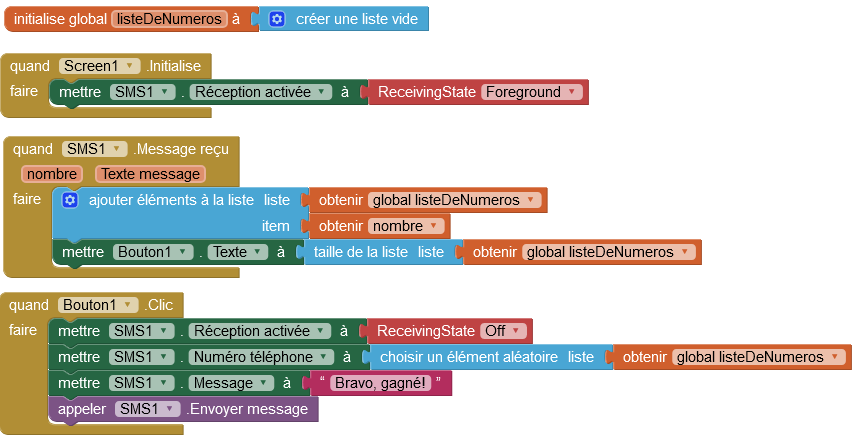
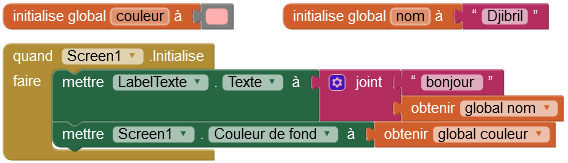
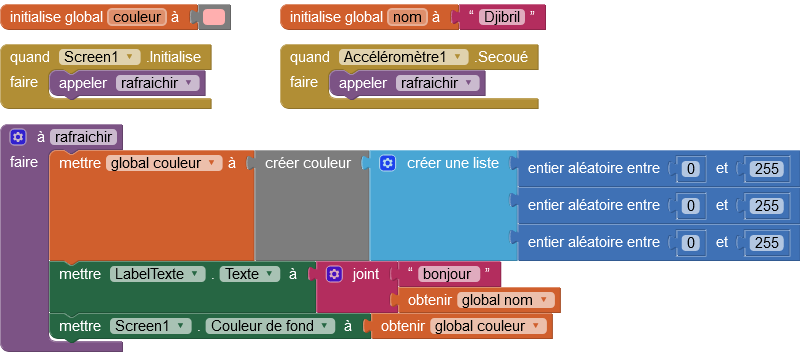
* [Video de présentation](http://onvaessayer.org/villon/videos/blocklyVillon.mp4)
* notions abordées :
  + algorithmes, décomposition, diagramme fonctionnel ou pseudo-code
* objectifs pédagogiques :
  + comprendre la notion d’algorithme comme méthode de résolution de problème,   
    et comprendre sa structure en séquences d’actions, boucles et conditions.
  + appréhender les étapes d’un développement : décomposition, passage du langage naturel au langage informatique (pseudo code), développement et test progressifs
* Déroulé : cf. [supports](http://onvaessayer.org/vaucanson/21-algorithmesProgrammes/1-algosProgrammes.php)
  + 35 mn : jeu blockly maze : [labyrinthe](https://blockly.games/maze?lang=fr&level=10&&skin=0) 
    - séquences (niveaux 1-2), boucles (niveaux 3-5), conditions (niveaux 6-8)

(pour info : ce [jeu similaire](https://blockly.games/turtle?lang=fr) est plus difficile, mais basé sur la géométrie)

* + 10 mn revue de définition d’un algorithme : [planches](http://onvaessayer.org/vaucanson/21-algorithmesProgrammes/1-algosProgrammes/assets/slides/algorithmes.pdf) (1a et1b planches 3 à 34)
  + 10 mn écrire un algorithme / décomposition, pseudo code, codage, traçage
    - écrire – en classe entière - le pseudo code du niveau 9 (ou 8 selon le niveau)
    - exercice de lecture/traçage de code :  
      <https://docs.google.com/forms/d/e/1FAIpQLSfqwk17gyssRa-ztWGRXQb_K8Jh48ZjgqkkvvrDujmVO4MqYw/viewform>
  + 5 mn debrief
    - qu’est-ce qu’un algorithme ?
    - comment écrire un algorithme : décomposer, écrire, tester
    - exercices à préparer le cours suivant : niveaux 9 et 10 ou niveau 10  
      <https://blockly.games/maze?lang=fr&level=9&&skin=0>
  + quiz sur les algorithmes (peut être fait au début du cours suivant
    - Quiz Google forms sur les algorithmes :  
      <https://docs.google.com/forms/d/e/1FAIpQLSf6v7jlArX32jfIS26okCdFk9tGQRMxlt8MH3J8SS5F1SYQhg/viewform>
    - exercice de lecture/traçage de code :  
      <https://docs.google.com/forms/d/e/1FAIpQLSfqwk17gyssRa-ztWGRXQb_K8Jh48ZjgqkkvvrDujmVO4MqYw/viewform>

## Séance 2 : Bonjour le monde, open food facts

Dans cette séance on commence par concevoir et réaliser ensemble une première application très simple qui consiste à afficher, "bonjour le monde", puis avec le nom de l’utilisateur, puis avec une couleur de fond aléatoire, puis en rafraichissant l’affichage en fonction d’un capteur. Elle permet parcourir des notions diverses.

* notions abordées :
  + l’environnement de développement App Inventor
  + les étapes de design et de programmation,
  + les blocs : forme, couleur et catégorie
  + les évènements, scripts, variables et procédures
* objectifs pédagogiques :
  + savoir réaliser une application mobile simple
  + comprendre
    - les étapes : design avec le choix des ressources, puis programmation et tests
    - la nature des blocs : get/set ou obtenir/mettre (correspondant à leur forme)
    - la structure évènementielle d’une application (event driven)
    - les catégories de blocs : variables, propriétés, procédures et fonctions, évènements, texte, booléens (associées à leur couleur)
    - la notion de variable
    - la notion de procédure
    - la notion de capteur (moyen de perception de l’environnement)
* déroulé sommaire
  + SMS game : réalisation d’une loterie, faite et expliquée par l’enseignant.   
    Les élèves jouent à la fin en envoyant un SMS, le gagnant tiré au sort reçoit un SMS  
    cf. [vidéo de développement de cette appli](http://onvaesayer.org/appinventor?video=smsgame)  
      
  + "bonjour le monde" réalisation d’une première application par les élèves
    - V1 : Bonjour le monde
    - V2 : Bonjour + nom utilisateur
    - V3 : Bonjour + nom utilisateur   
      + fond d’écran de couleur aléatoire
    - V4 : ajout du rafraichissement d’écran quand on secoue l’appareil
      * + déroulé détaillé : voir chapitre

## Séance 4 : Application web (open food facts)

Rédaction en cours

## Séance 5 : Application géolocalisée

Rédaction en cours

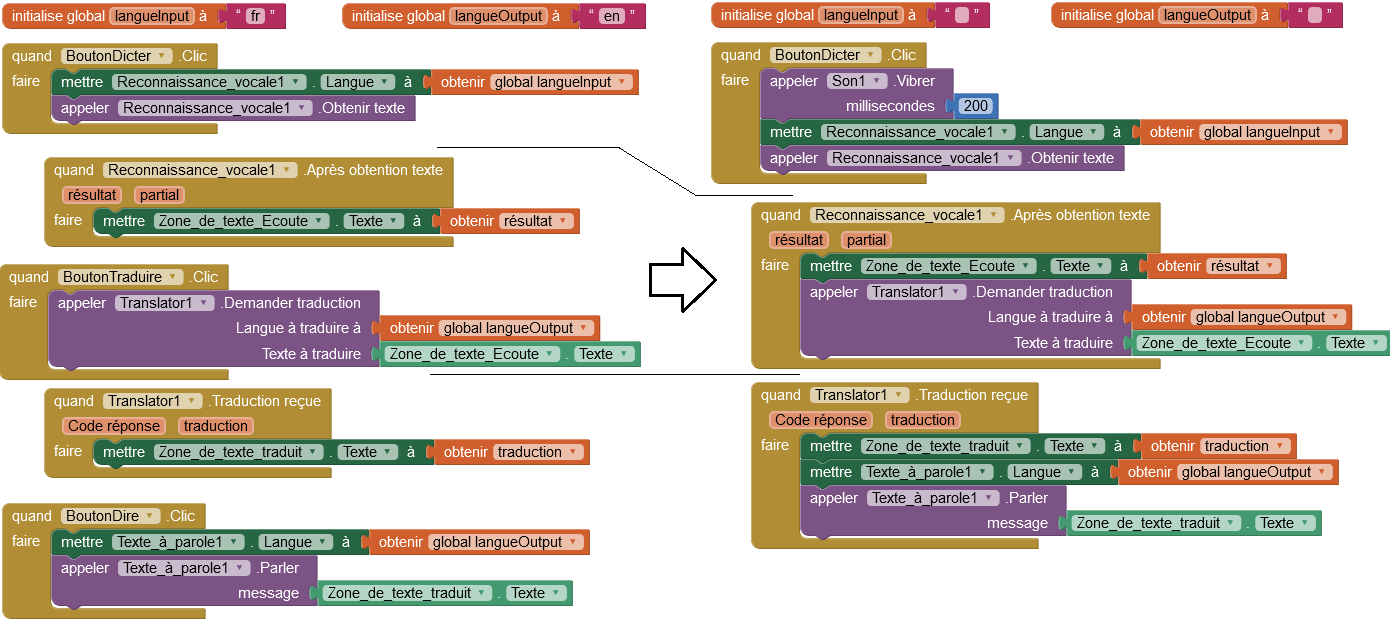
# Déroulés détaillés

* séance 2 – "bonjour le monde"

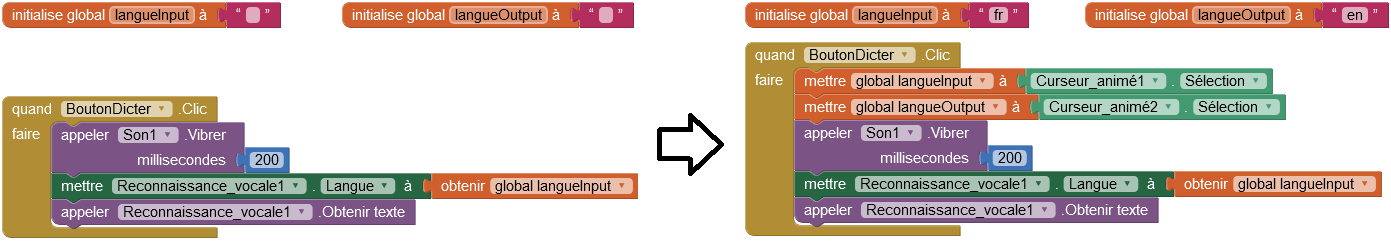
## Séance 3 : Application de traduction

Dans cette séance les élèves réalisent une application qui va écouter ce que dit l’utilisateur, le traduire dans une autre langue et le redire dans cette langue. L’application fait appel à des fonctions de reconnaissance vocale, de traduction et de diction (texte à parole).

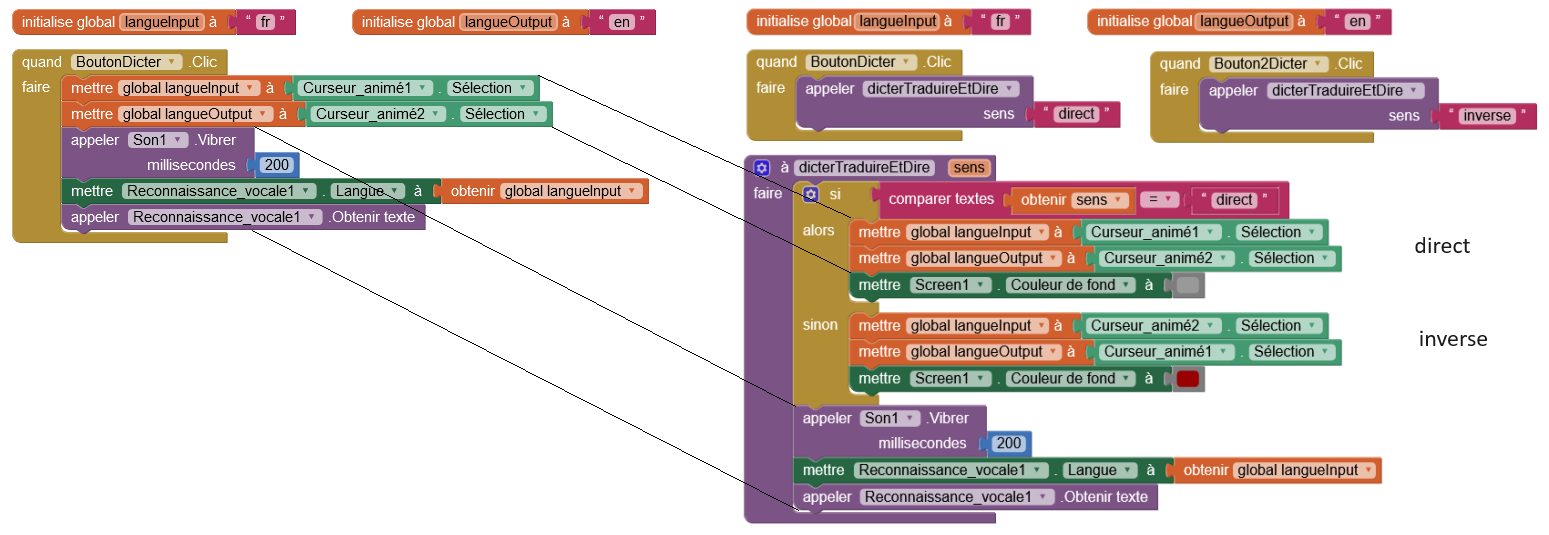
* notions abordées :
  + décomposition fonctionnelle : services et procédures asynchrones (on n’attend pas la réponse, son arrivée donne lieu à un évènement traité dans un autre script)
  + utilisations de services externes et/ou en ligne : reconnaissance de la parole, traduction, …
  + fonctions dites "asynchrones" : dans un script, le programme n’attend pas la réponse à une demande. L’arrivée de la réponse donnant lieu à un évènement traité par un nouveau script (programmation évènementielle).
* objectifs pédagogiques :
  + décomposer ce que doit faire l’application et trouver les composants pour le faire
    - 1. Convertir parole en texte, 2. Traduire texte, 3. dire le texte traduit
  + voir la différence entre une fonction qui donne la réponse tout de suite et une fonction (ou procédure) asynchrone où on n’attend pas la réponse.
  + utiliser une fonction avec des paramètres en entrée
  + modifier les propriétés d’une fonction
  + utiliser les données ou paramètres transmis par un évènement
  + utiliser une liste déroulante (spinner) pour choisir la langue
* déroulé sommaire (voir déroulé détaillé en annexe)
  + V1 : version initiale (un bouton par étape)
  + Design
    - 1 bouton pour écouter et convertir en texte ce que dit l‘utilisateur
    - 1 bouton pour traduire le texte dans une autre langue
    - 1 bouton pour lire à haute voix le texte traduit
    - 2 zones de texte pour afficher les textes reconnus et traduits
    - 3 composants : reconnaissance vocale, traduction et ‘texte à parole’
  + Codage/programmation : 5 scripts
    - Quand clic sur 1° bouton : lancer la reconnaissance vocale
    - Quand le texte a été reconnu : l’afficher
    - Quand clic sur 2° bouton : lancer la traduction
    - Quand le texte a été traduit : afficher ce texte traduit
    - Quand clic sur 3° bouton : convertir le texte traduit en parole
  + V2 : enchainement des opérations sur un bouton
    - Codage/programmation : 3 scripts
      * Quand clic sur 1° bouton : lancer la reconnaissance vocale
      * Quand le texte a été reconnu : l’afficher et lancer la traduction
      * Quand texte a été traduit : afficher traduction et convertir en parole

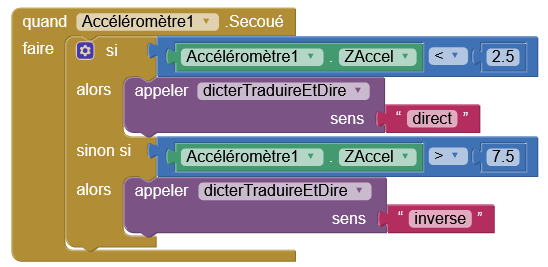


* + V3 : choisir les langues en entrée et en sortie
    - Design
      * Ajout de 2 spinners (curseurs animés) pour choisir chaque langue dans une liste. On définit la liste dans les propriétés et la langue sélectionnée au départ.
    - Codage
      * Quand l’utilisateur a sélectionné une nouvelle langue, mise à jour de la variable qui porte cette langue : langueInput ou langueOutput



* + V4 : traduction bidirectionnelle, selon l’orientation du smartphone
    - Design
      * ajout d’un bouton pour dicter dans l’autre sens
      * accéléromètre pour détecter la position du smartphone
    - Codage
      * Quand clic sur bouton1 Dicter
        + langueInput  langue sélectionnée par le curseur 1
        + langueOutput langue sélectionnée par le curseur 2
        + appeler la procédure dicter
      * Quand clic sur bouton1 Dicter
        + langueInput  langue sélectionnée par le curseur 2
        + langueOutput langue sélectionnée par le curseur 1
        + appeler la procédure dicter





Bonus avec l’accéléromètre

* Quand téléphone secoué
  + si (acc z >7.5)

appeler la procédure dicter (direct)

* + sinon sii (acc z <2.5)

appeler la procédure dicter (inverse)



## Séance 4 : Application web (open food facts)

Rédaction en cours

## Séance 5 : Application géolocalisée

Rédaction en cours

# Déroulés détaillés

## séance 2 – "bonjour le monde"

### Introduction avec SMS game :

L’animateur réalise une application qui attend et enregistre les SMS, puis - quand on clique sur un bouton – tire un numéro au hasard et lui envoie un SMS "vous avez gagné".

|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions** |
| Design conception | Design / décomposition:   * Dessiner l’interface et écrire une phrase qui décrit l’application * Identifier * les ressources nécessaires :  bouton + composant sms * les évènements * et données manipulées (liste numéros) | maquette  évènements  composants visibles (bouton) et cachés : SMS  variable de type liste |
| Design réalisation |  | Ajout et paramétrage des composants,  Mise en page |
| Pseudo code | * déclarer et initialiser la liste des numéros * quand application démarre   + activer SMS * quand SMS reçu   + ajouter numéro à liste numéros * quand bouton cliqué   + choisir un numéro au hasard   + envoyer un sms à ce numéro | Pseudo code de chaque évènement |
| codage |  | codage  vérification, traitement des erreurs |
| Création exécutable | * Installation d’AI2 companion depuis le playstore * Construire l’exécutable      * Télécharger l’application sur le smartphone | création de l’exécutable (.apk)  transfert sur smartphone  essais |

### Application "bonjour le monde" :

|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions vues** |
| version 1 | Préparation : installer l’appli AI2 companion sur le smartphone  Description : au démarrage, l’application affiche ‘bonjour le monde’  Design :  Ajout d’un composant label : label texte réglage des propriétés à droite :  Centrage verticale et horizontal de Screen1  Réglage de la taille et de la couleur des caractères de labelTexte  Evènements à traiter:  quand l’application démarre (screen.initialize)  Pseudo code :  *quand Screen1. Initialize*  *mettre le texte de labelTexte à "bonjour le monde"*  Codage :    Construire l’exécutable :    Télécharger sur le smartphone :    Tester | Réalisation d’une application complète minimale  Design :   * Choix et ajout d’un composant label pour afficher un texte   Mise en page :   * modification des propriétés de Screen1 : centrage horizontal et vertical, choix de la couleur de fond * modification des propriétés de labelTexte, taille et couleur des caractères   compilation du code, transfert smartphone, exécution et tests |
| Version 2 | Description : au démarrage, afficher ‘bonjour’ + le nom de l’utilisateur  Design : sans changement  Evènements : sans changement  Pseudo code :  *Initialiser la variable nom avec une valeur*  *quand screen1. Initialize*  *mettre le texte de labelTexte à "bonjour le monde" + nom*  Codage :    Construire l’exécutable :  Télécharger sur le smartphone :  Tester | Variable de type texte  Bloc de jointure de textes |
| Version 3 | Description :  comme précédent + choisir une couleur aléatoire pour le fond d’écran couleur du fond d’écran de façon  Design : sans changement  Evènements : sans changement  Pseudo code :  *Initialiser la variable couleur  Initialiser la variable nom avec une valeur*  *quand screen1. Initialize*  *mettre le texte de labelTexte à "bonjour le monde" + nom*  *mettre dans la variable couleur, une dose aléatoire de rouge, vert et bleu affecter la couleur à la propriété couleur de fond d’écran*  Codage :    Construire l’exécutable, Télécharger sur le smartphone, Tester | Définition d’une couleur depuis une liste de base  Définition d’une couleur par ses composantes RVB entre 0 et 255  Fonction qui renvoie un entier aléatoire  Modification d’une propriété de Screen1 |
| Version 4 | Description : idem + rafraichir quand on secoue l’appareil  Design : Ajouter un composant non visible accéléromètre  Evènements : Nouvel évènement quand accéléromètre secoué  Pseudo code :  *Initialiser la variable couleur  Initialiser la variable nom avec une valeur*  *quand screen1. Initialize*  *appeler la fonction rafraichir*  *quand accéléromètre. secoué*  *appeler la fonction rafraichir*  ***procédure rafraichir***  *mettre labelTexte.texte à "bonjour le monde" + nom*  *mettre dans la variable couleur, une couleur qui contient une dose aléatoire de rouge, de vert et de bleu affecter la couleur à la propriété couleur de fond d’écran*  Codage :    Construire l’exécutable, Télécharger sur le smartphone, Tester | Utilisation d’un composant accéléromètre  Utilisation d’une procédure pour :  1. éviter les doublons 2. Généraliser / abstraire  nota :  on commence par dupliquer le code, puis on crée la procédure pour éviter les doublons |

## séance 3 – "application de traduction"

### V1 : écouter, traduire et dire dans une autre langue

|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions** |
| Description | l’application doit permettre d’écouter ce que dit l’utilisateur, le traduire dans une autre langue et le redire dans cette langue |  |
| Décomposition | * "dicter" et convertir en texte à afficher (reconnaissance vocale) * "traduire" et afficher traduction qui est affichée * "lire » le texte traduit à haute voix | Analyse, décomposition fonctionnelle |
| Design réalisation | Composants visibles   * 3 boutons : "écouter", "traduire", "lire" * une zone de texte pour le texte reconnu * une zone de texte pour le texte écouté traduit   Composants invisibles   * reconnaissance vocale, translator, texte à parole   Propriétés   * ajouter une image au format png (transparent) en fond d’écran * choisir le nom et la taille des boutons | Ajout et paramétrage des composants,  Mise en page |
| Données et variables | Données et variables de l’application   * langueInput : la langue à écouter en entrée * langueOutput : la langue de traduction en sortie * texte écouté (propriété du composant texte) * texte traduit (propriété du composant texte) |  |
| Pseudo code | * définir/initialiser langueInput et langueOutput * quand clic sur bouton dicter   + mettre la propriété langue à écouter à langueInput   + appeler la fonction de reconnaissance vocale * quand texte reconnu   + afficher le texte dans la zone de texte ecoute * quand clic sur traduction   + appeler la fonction de traduction avec en paramètres, le texte à traduire et la langue (langueOutput) * quand traduction obtenue   + afficher la traduction dans la zone de texte trad. * quand clic sur lire   + mettre la propriété langue à lire à langueOutput   + appeler la fonction texte à parole avec le texte traduit |  |
| codage |  |  |

### V2 : automatiser (tout enchainer sur un seul clic)

|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions** |
| Description | Dans cette version, on va enchainer les actions après avoir déclenché la reconnaissance vocale avec le bouton dicter, donc lancer la traduction dès que le texte est reconnu et lancer la lecture dès que la traduction est disponible. Les boutons traduire et lire deviennent inutiles. On va aussi améliorer l’ergonomie en faisant vibrer le smartphone quand on demande la reconnaissance vocale | Analyse, décomposition fonctionnelle |
| Description Décomposition | * Quand l’utilisateur clique sur le bouton « écouter » écouter ce que dit l’utilisateur, * quand le texte est reconnu   afficher le résultat et demander la traduction   * quand la traduction arrive, l’afficher et demander la lecture à haute voix | Analyse, décomposition fonctionnelle |
| Design | * Ajouter un composant ‘son’ pour pouvoir faire vibrer * On n’utilisera plus les boutons traduire et lire,  mais avant de les supprimer, récupérer les lignes de code |  |
| Pseudo code | * Quand l’utilisateur clique sur le bouton "dicter"   + faire vibrer   + et demander la reconnaissance vocale * quand le texte est reconnu   + afficher le résultat   + demander la traduction * quand la traduction arrive,   + afficher la traduction   + demander la lecture à haute voix |  |
| codage |  |  |

### V3 : choisir la langue à écouter et à traduire dans une liste



|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions** |
| Description | Dans cette version, l’utilisateur peut choisir la langue en entrée et la langue en sortie |  |
| Décomposition | On va utiliser un curseur animé (spinner ou liste déroulante) pour choisir chacune des langues. La liste des langues sera définie dans les propriétés des 2 curseurs animés (on peut aussi les définir dans le programme, quand Screen1.initialize) |  |
| Design/ composants | * Composants visibles : ajouts   + un arrangement horizontal (largeur remplir parent)   + avec 2 curseurs animés (spinner) In et Out     - sélection ‘fr’ pour in, ‘en’ pour out     - element : fr,ar,de,en,es,it pour les deux   + séparés par un label (largeur fill parent) * invisibles : inchangés |  |
| Design |  |  |
| Pseudo code | * Après sélection dans la liste des langues à écouter :   + Mettre la langue sélectionnée comme langue en entrée * Après sélection dans la liste des langues de traduction :   + Mettre la langue sélectionnée comme langue de traduction |  |
| code |  |  |

### V4 : traduction bidirectionnelle

|  |  |  |
| --- | --- | --- |
| **étape** | **Description** | **Notions** |
| Description | * l’application doit traduire dans un sens quand c’est l’utilisateur qui parle et dans l’autre quand c’est l’interlocuteur qui répond. |  |
| Design / analyse | * utiliser l’orientation du smartphone (vertical ou tendu vers l’utilisateur) pour déterminer qui parle. * à chaque changement changer la couleur de l’écran |  |
| Design / composants | * visibles : sans changement * invisibles : ajout d’un composant accéléromètre |  |
| Design |  |  |
| Pseudo code |  |  |
| codage |  |  |
| Bonus accéléro | * quand smartphone secoué change   + si accélération z < 2.5     - appeler dicterTraduireEtDire(direct)   + sinon accélération z < 2.5     - appeler dicterTraduireEtDire(inverse) |  |

# Atelier sur le son

## Principe :

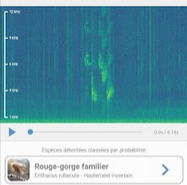
* le son est abordé comme une vibration qui se propage à une vitesse qui dépend du milieu,  
  puis on expérimente la notion de fréquence, de graves et d‘aigus, d’infra et ultra-sons  
  puis la notion de niveau sonore mesuré en décibels, avec les risques pour la santé,
* on voit ensuite à la représentation du son en fonction du temps, puis en fonction du temps et de la fréquence avec le spectrogramme (on évite la notion de spectre),
* enfin on applique ces notions à des cas concrets : reconnaissance des oiseaux par leur chant (birdNet), mesure d’une distance par l’écho, ou de la réverbération dans une salle.

## Environnement matériel et logiciel :

Matériel

* un PC avec [Libre office](https://fr.libreoffice.org/download/telecharger-libreoffice/), une enceinte BT
* un smartphone avec un sonomètre ou une appli [sonomètre](https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=fr) calibrée   
  (prévoir un logiciel comme Vysor qui permet de partager l’écran du smartphone)
* en option des petits matériels (bougie et entonnoir).

Logiciel et supports :

* Récupérer les planches de présentation et les sons dans le fichier compressé à [cette adresse](http://onvaessayer.org/villonCycle4?res=atelierSonZip), décompresser (unzip) ce fichier et vérifier que la présentation fonctionne en cliquant sur les liens qui jouent les sons.
* Si vous voulez faire des analyses en live, installer le logiciel [audacity](https://www.audacityteam.org/)  sur le PC
* et si vous voulez que les élèves expérimentent eux-mêmes, prévoir un smartphone par groupe de 3 ou 4 avec [sound meter](https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=fr), [Birdnet](https://play.google.com/store/search?q=birdnet&c=apps&hl=fr) ou [Merlin bird](https://play.google.com/store/search?q=merlin%20bird&c=apps&hl=fr)et [audio spectrum analyzer](http://onvaessayer.org/?res=atelierSonApks) ou équivalent ; en option [function generator](http://onvaessayer.org?res=atelierSonApks) ou [frequency sound generator](https://play.google.com/store/apps/details?id=com.luxdelux.frequencygenerator&hl=fr) (voir descriptif dans les [vidéos](http://onvaessayer.org/villonCycle4?video=atelierSon1)),

## Déroulé :

voir planches et media (dans [atelierson.zip](http://onvaessayer.org?app=atelierson)) en support et résumés : [vidéoSon1](http://onvaessayer.org?video=atelierson1) et [vidéoSon2](http://onvaessayer.org?video=atelierson2)