

CYCLE 4

Assistant vocal :
Comment peut-il passer de la parole à l'action ?

**NIVEAU
TROISIÈME**

Présentation de la séquence

La séquence est la poursuite de la séquence : « **Assistant vocal : Comment fonctionne-t-il et quels sont ses effets sur l'utilisateur ?** », toutefois il est possible de mettre en place cette séquence sans la précédente.

La séquence est composée de trois séances. Elle permet aux élèves de comprendre comment un assistant vocal peut passer d'une phrase prononcée par l'utilisateur à une action réalisée par un objet connecté.

Les élèves comparent d'abord un assistant vocal programmé par règles avec une version utilisant un LLM (*large language model*) appelé communément « *IA générative* ». Ils identifient ce qu'apporte un modèle de langage comme « gpt » utilisé via Groq : produire une réponse, interpréter une intention et adapter son comportement à partir de règles données dans un prompt.

Ils modifient ensuite l'assistant vocal EduVox pour piloter un système via une commande Bluetooth. Ils ajoutent une nouvelle commande vocale, puis testent son fonctionnement sur l'objet connecté.

Enfin, ils analysent une application qui utilise un LLM pour transformer une intention en commande autorisée. Ils comprennent le rôle du prompt système, du prompt utilisateur, du prompt engineering, du filtre de sécurité et de la chaîne d'information complète.

Afin de s'approprier rapidement la séquence, chaque séance est accompagnée d'un document professeur détaillant les actions effectuées par l'enseignant.

Préconisation technique préalable :

- La séquence nécessite un réseau compatible avec MIT App Inventor et MIT AI2 Companion, un accès HTTPS aux API utilisées (*accès Internet + WiFi*) , un appareil mobile (*tablette*) avec micro et reconnaissance vocale, le Bluetooth et la communication avec la carte micro:bit. (voir document ressource professeur : *Seq2_preconisations_techniques.pdf*)
- Les fichiers App Inventor diffusés sont fournis sans clé API réelle.** Avant la séance, l'enseignant doit renseigner une clé API temporaire dédiée au projet EduVox dans les applications utilisant Groq / gpt-oss-20b. Cette clé ne doit pas apparaître dans les fiches élèves, les captures d'écran ou les documents partagés. (voir document ressource professeur : *Seq2_tutoriel_professeur_cle_API_Groq*)

Thème abordé : Les objets et les systèmes techniques : leurs usages et leurs interactions à découvrir et à analyser

Attendu de fin de cycle : Décrire les liens entre usages et évolutions technologiques des objets et des systèmes techniques

Compétences

Mettre en relation une découverte scientifique avec ses développements technologiques et leurs effets sur la société

Connaissances

Les grands types d'apprentissage des intelligences artificielles et leurs usages possibles (géolocalisation, identification, calcul, traduction, etc.).

Thème abordé : Structure, fonctionnement, comportement : des objets et des systèmes techniques à comprendre	
Attendu de fin de cycle : Décrire et caractériser l'organisation interne d'un objet ou d'un système technique et ses échanges avec son environnement (énergies, données)	
Compétences Décrire un OST en caractérisant sa chaîne d'information	Connaissances Les fonctions des constituants suivants : capteurs (température, présence, distance, etc.), microcontrôleur, composants d'une interface entre l'humain et la machine (IHM) : boutons, afficheurs, etc..
Attendu de fin de cycle : Comprendre et modifier un programme associé à une fonctionnalité d'un objet ou d'un système technique	
Compétences Déterminer les données utilisées et produites par un programme associé à une fonctionnalité en vue de le modifier.	Connaissances -entrées ou sorties d'un programme (données issues par exemple de capteurs IHM et sorties pouvant être en lien avec un actionneur, fichiers).
Modifier et tester le programme associé à une nouvelle fonctionnalité d'un OST	Algorithmique et programmation :

PROPOSITION DE DÉROULEMENT DE LA SÉQUENCE

Séance 1 : Qu'apporte un LLM par rapport à un assistant vocal programmé par règles ? (80 mn)

Problématique :

Qu'apporte un LLM par rapport à un assistant vocal programmé par règles ?

➤ Mise en situation (5 mn)

La séance débute par un rappel de la séquence précédente. Les élèves ont déjà testé EduVox Niveau 1 et EduVox Niveau 2. Ils ont constaté qu'un assistant vocal simple peut reconnaître un mot-clé exact ou vérifier si une phrase contient un mot-clé.

Dans le cas où l'enseignant n'a pas traité la séquence, il faut prévoir une démonstration du fonctionnement des deux versions de l'assistant à la classe.

Le professeur propose maintenant de comparer ces deux versions avec une nouvelle application EduVox LLM.

➤ Description de la situation (5 mn)

L'enseignant utilise le nouvel assistant vocal à plusieurs reprises. Les élèves comprennent que les trois applications ont le même point de départ visible : l'utilisateur parle dans une application. Cependant, leur fonctionnement interne n'est pas le même. **EduVox N1 et N2 utilisent des règles prévues dans les blocs du programme**, alors qu'EduVox LLM envoie la question et des règles à un modèle de langage.

➤ Propositions / Hypothèses (5 mn)

Les élèves émettent des hypothèses sur ce que pourrait apporter un LLM. Ils peuvent proposer *qu'il comprend mieux les phrases, qu'il répond à davantage de questions, qu'il utilise une intelligence artificielle ou qu'il produit des réponses qui ne sont pas écrites à l'avance dans le programme*.

Ces hypothèses sont notées au tableau afin d'être confrontées aux tests.

➤ Investigations (35 mn)

Les élèves testent les trois versions d'EduVox : Niveau 1, Niveau 2 et EduVox LLM. Ils posent plusieurs questions : un mot-clé simple comme « bluetooth », une phrase complète comme « Peux-tu m'expliquer le bluetooth ? », « une demande sur les données personnelles », une question sur la « vie privée » et une question non prévue comme « Quelle est la capitale du Japon » ?

Ils complètent un tableau pour indiquer si chaque version répond correctement, ne comprend pas ou répond à une question non prévue.

Ils observent ensuite les blocs App Inventor d'EduVox LLM à l'aide de la fiche ressource (Seq2-s1_fiche_ressource.pdf). Ils identifient :

- le bouton Parler,
- le composant Speech, I
- a variable texte, le
- bloc Screen1.Initialize,
- l'envoi au LLM avec Web1,
- la réception de la réponse.

Pour chaque bloc identifié, ils doivent ensuite identifier son rôle parmi :

- A. Stocker la phrase reconnue.
- B. Récupérer, afficher et lire la réponse du LLM.
- C. Lancer l'écoute de la question.
- D. Envoyer la question et les règles à l'intelligence artificielle.
- E. Préparer l'application au démarrage.
- F. Transformer la voix en texte.

Les élèves doivent ensuite reconstruire le trajet de l'information en remettant dans l'ordre les étapes du fonctionnement :

- La réponse est affichée à l'écran et lue à voix haute.
- L'utilisateur clique sur le bouton Parler.
- Le composant Speech transforme la voix en texte.
- Le texte reconnu est stocké dans la variable texte.
- Le programme envoie la question et les règles au LLM avec Web1.
- Le LLM génère une réponse.

L'enseignant circule dans les groupes et amène les élèves à conclure sur le fait que grâce au LLM, l'assistant réponds à des questions alors que la réponse n'est pas dans la base de données de l'application.

➤ **Analyse et mise en perspective (15 mn)**

La mise en commun permet de comparer les trois versions.

EduVox N1 répond seulement quand la demande correspond exactement à un mot-clé.

EduVox N2 reconnaît davantage de phrases car il vérifie si la phrase contient un mot-clé.

EduVox LLM peut produire une réponse à partir de la question et des règles données dans le prompt.

Le professeur introduit simplement le vocabulaire : **un LLM est un modèle de langage**. Dans le projet EduVox, gpt-oss-20b est le modèle utilisé via Groq.

Le prompt système correspond aux règles générales données à l'IA, tandis que le prompt utilisateur correspond à la demande formulée par l'utilisateur.

Le prompt engineering désigne le travail qui consiste à rédiger, tester, comparer puis améliorer un prompt.

➤ **Bilan de mes recherches (10 mn)**

Un assistant vocal programmé par règles ne comprend pas réellement toutes les phrases possibles. Il applique des règles prévues dans le programme.

EduVox N1 compare la question à un mot-clé exact.

EduVox N2 vérifie si la phrase contient un mot-clé prévu.

EduVox LLM fonctionne différemment. La question de l'utilisateur est transformée en texte, puis envoyée à un LLM avec des règles appelées prompt.

Le LLM produit une réponse générée, qui n'est pas simplement choisie dans une liste écrite à l'avance dans les blocs. Cette réponse peut varier, ce qui impose de bien encadrer l'IA lorsque l'on veut piloter un objet réel.

➤ **Synthèse (5 mn)**

L'enseignant distribue le document de structuration (IA_LLM_usages) et l'explicite.

Ressources pour le professeur

Fiches :

Seq2-s1-eleve_correction.pdf (+docx)

Seq2-s1_fiche_ressource.docx

Ressource professeur :

Seq2-s1_fiche_professeur.pdf (+docx)

Seq2_tutoriel_professeur_cle_API_Groq.pdf

EduVox_N1.apk

EduVox_N2.apk

Structuration de connaissances :

Ressources pour les élèves

Fiches :

Seq2-s1-eleve.pdf (+docx)

Ressources :

Seq2_S1_EduVox_N1_corrige.aia

Seq2_S1_EduVox_N2_corrige.aia

Seq2_S1_EduVox_QR_LLM_sans_cle.aia

Seq2-s1_fiche_ressource.pdf

Structuration de connaissances :

Structuration_IA_LLM_usages.pdf

Séance 2 : Comment ajouter une nouvelle commande vocale pour déclencher une action sur un objet connecté ? (80 mn)

➤ Problématique :

Comment ajouter une nouvelle commande vocale pour déclencher une action sur un objet connecté ?

➤ Mise en situation (5 mn)

La séance débute par la projection de l'image « Situation_déclenchante ».



Le professeur fait verbaliser des exemples connus : allumer une lampe, fermer un volet roulant, verrouiller une porte ou déclencher une action par la voix avec un assistant comme Alexa ou Google Home.

Un point particulier est fait pour montrer l'intérêt sociétal de l'assistant pour l'inclusion. On expliquera le rôle d'une alerte visuelle pour remplacer une sonnette pour les malentendants.

Les élèves comprennent qu'une phrase prononcée doit devenir une commande exploitable par un objet technique.

Description de la situation (5 mn)

Le professeur présente le projet « EduVox Commande ». Pour comprendre le fonctionnement d'un assistant, les élèves manipulent un système didactique composé d'une carte microbit, d'une LED et d'un servomoteur.

L'application App Inventor transforme la voix en texte, recherche une commande connue dans la base de donnée TinyDB, puis envoie cette commande par Bluetooth à la carte micro:bit. La carte micro:bit exécute ensuite l'action sur une LED branchée sur P1 ou un servomoteur branché sur P0 du shield Grove.

Propositions / Hypothèses (5 mn)

Les élèves proposent des hypothèses sur ce qu'il faut modifier pour ajouter une nouvelle action. Certains peuvent penser qu'il suffit de modifier l'application. D'autres peuvent penser qu'il faut modifier le programme de la carte micro:bit.

Ces hypothèses permettent de faire émerger l'idée centrale : pour ajouter une action, il faut modifier à la fois ce que l'application connaît et ce que la carte micro:bit sait exécuter.

Investigations (35 mn)

Les élèves préparent le matériel : LED externe sur P1, servomoteur SG90 sur P0, carte micro:bit V2, shield Grove ou BitMaker, tablette ou smartphone avec MIT AI2 Companion.

Le travail peut être réparti dans l'îlot ou conduit en intégralité.

A l'aide des fiches ressources ([Seq2-s2_fiche_ressource_app_inventor.pdf](#) et [Seq2-s2_fiche_ressource_makecode.pdf](#)), les élèves transfèrent le programme micro:bit fourni ([Seq2-s2_EduVox_Mode_Commande_eleve.hex](#)), puis connectent l'application « EduVox Commande » ([Seq2-s2_EduVox_Mode_Commande_eleve.aia](#)) à la carte micro:bit en Bluetooth. Ils testent d'abord les commandes déjà prévues : **allume, eteins, deverrouille et verrouille**. Ils constatent ensuite que la commande « **clignote** » ne fonctionne pas encore. **Cette situation permet d'introduire le besoin de modifier le système.**

Dans App Inventor, les élèves ajoutent une nouvelle commande dans la base de données TinyDB. TinyDB est présenté comme la mémoire de l'application. Chaque information est enregistrée avec un tag et une valeur à stocker. Le tag est le mot ou la phrase que l'application doit reconnaître. La valeur à stocker est la commande envoyée à la carte micro:bit. **Pour une version simple, les élèves ajoutent tag : clignote et valeur à stocker : clignote.**

Dans MakeCode, les élèves ajoutent une condition « **sinon si** » le message comprend « **clignote** ». La carte micro:bit doit alors faire **clignoter 5 fois la LED branchée sur P1** : écrire P1 à 1, pause 300 ms, écrire P1 à 0, pause 300 ms, le tout répété 5 fois. Ils transfèrent ensuite le programme modifié et testent la commande.

Analyse et mise en perspective (15 mn)

La mise en commun permet de **distinguer les deux rôles**. App Inventor prépare et envoie la commande. MakeCode reçoit la commande et réalise l'action. **Si l'un des deux éléments n'est pas modifié, la nouvelle fonctionnalité ne peut pas fonctionner correctement.**

Le professeur attire l'attention des élèves sur un point important : les mots utilisés dans l'application App Inventor et dans le programme MakeCode doivent être exactement les mêmes. Par exemple, si l'application envoie la commande « **allume** », le programme de la carte micro:bit doit tester le mot « **allume** ». **Si les deux programmes n'utilisent pas le même mot, la carte micro:bit ne reconnaît pas la commande et l'action ne se déclenche pas.**

Un prolongement peut être proposé aux élèves rapides : comparer les mots utilisés pour le servomoteur. Si l'application envoie « **ouvre** », mais que MakeCode teste « **ouvrir** », la commande risque de ne pas être reconnue. De la même manière, si l'application envoie « **ferme** », mais que MakeCode teste « **fermer** », l'action peut échouer. Les élèves doivent donc corriger le programme pour que les mots soient identiques des deux côtés.

Bilan de mes recherches (10 mn)

Dans une maison connectée, un assistant vocal peut commander des objets techniques. Dans EduVox, l'application transforme la voix en texte, puis recherche une commande dans la base de données. Cette commande est envoyée par Bluetooth à la carte micro:bit.

Pour ajouter une nouvelle action, il faut intervenir à deux endroits : dans la base de données pour que l'application connaisse la commande, puis dans MakeCode pour que la carte micro:bit réalise l'action. La commande envoyée doit être simple, cohérente et écrite de la même façon dans l'application et dans le programme de la carte.

➤ Synthèse (5 mn)

L'enseignant distribue le document de structuration (*programme_donnees_commandes*) et l'explique.

Ressources pour le professeur

Fiches :

Seq2-s2-eleve_correction.pdf (+docx)
Seq2-s2_fiche_ressource_app_inventor.docx
Seq2-s2_fiche_ressource_makecode.docx

Ressources :

Situation_declenchante.png
Seq2-
s2_EduVox_Mode_Commande_corrige.aia
Seq2-
s2_EduVox_Mode_Commande_corrige.hex

Ressource professeur :

Seq2-s2_fiche_professeur.pdf (+docx)
Seq2_maquette_serrure_servo.pdf (+docx)

Structuration de connaissances :

Structuration_programme_donnees_commandes.docx

Ressources pour les élèves

Fiches :

Seq2-s2-eleve.pdf (+docx)

Ressources :

Seq2-s2_fiche_ressource_app_inventor.pdf
Seq2-s2_fiche_ressource_makecode.pdf
Seq2-
s2_EduVox_Mode_Commande_eleve.aia
Seq2-
s2_EduVox_Mode_Commande_eleve.hex

Structuration de connaissances :

Structuration_programme_donnees_commandes.pdf

Séance 3 : Comment un assistant vocal avec IA peut-il comprendre une intention et envoyer une commande correcte à un objet connecté ? (80 mn)

Problématique :

Comment un assistant vocal avec IA peut-il comprendre une intention et envoyer une commande correcte à un objet connecté ?

Mise en situation (5 mn)

La séance débute par un rappel de la séance 2. Les élèves ont ajouté une commande vocale et compris qu'une application pouvait envoyer une commande à une carte micro:bit.

Le professeur pose alors une nouvelle question : **que se passe-t-il si l'utilisateur ne donne pas une commande directe, mais exprime une intention** comme « *Il fait sombre dans la pièce* » ou « *Je vais dormir* » ?

Description de la situation (5 mn)

Les élèves découvrent « *EduVox IA Bluetooth* ». Cette application utilise un LLM pour interpréter la phrase prononcée. Le LLM ne doit pas répondre librement : il doit choisir un seul mot parmi une liste de commandes autorisées. L'application vérifie ensuite la réponse avec un filtre avant d'envoyer la commande par Bluetooth à la micro:bit.

Propositions / Hypothèses (5 mn)

Les élèves émettent des hypothèses sur le rôle du LLM et sur les risques possibles. **Ils peuvent proposer que le LLM devine l'intention, qu'il transforme *Il fait sombre* en *allume*, ou qu'il peut parfois se tromper.**

Le professeur fait émerger l'idée qu'un objet réel ne doit pas être piloté directement par une réponse libre de l'IA : **il faut encadrer et vérifier la commande.**

Investigations (35 mn)

Les élèves préparent le matériel : LED externe sur P1, servomoteur SG90 sur P0, carte micro:bit V2, shield Grove ou BitMaker, tablette ou smartphone avec MIT AI2 Companion.

L'enseignant fournit les cartes micro:bit avec le programme (*Seq2_s3_EduVox_Mode_Commande_corrige.hex*) préchargé.

A l'aide de la fiche ressource (*Seq2_s3_fiche_ressource_EduVox_IA_Bluetooth.pdf*), les élèves connectent l'application « EduVox_IA_Bluetooth » (*Seq2_s3_EduVox_IA_Bluetooth_sans_cle.aia*) à la carte micro:bit en Bluetooth. Puis ils testent plusieurs phrases : « Il fait sombre », « Éteins la lumière », « Fais clignoter la LED », « Je vais dormir », « Je veux entrer », « Raconte-moi une blague ». Ils observent la réponse du LLM, la commande envoyée et l'action réalisée par la LED ou le servomoteur.

Ils reconstituent ensuite la chaîne d'information complète en huit étapes : l'utilisateur prononce une phrase, Speech transforme la voix en texte, l'application envoie la phrase au LLM, le LLM interprète l'intention, l'application vérifie que la réponse correspond à une commande autorisée, SendText envoie la commande par Bluetooth, la micro:bit reçoit la commande, puis la LED ou le servomoteur réalise l'action.

Les élèves analysent le prompt système. Ce prompt impose au LLM de répondre par un seul mot parmi « allume, éteins, clignote, verrouille, déverrouille ou inconnu ».

Il contient des règles comme :

- « sombre ou manque de lumière » = « allume »,
- « dormir, partir ou sécuriser » = « verrouille »,
- « entrer ou ouvrir » = « déverrouille »,
- « éteindre la lumière » = « éteins »,
- « faire clignoter » = « clignote ».

Les élèves testent ensuite les limites de l'interprétation avec des phrases plus vagues. Ils discutent les réponses possibles et comprennent qu'une intention peut être ambiguë.

En fin de séance, ils utilisent une application didactique « *EduVox Prompt Système* » (*Seq2_s3_EduVox_Prompt_Systeme_sans_cle.aia*). Cette application n'est pas connectée en Bluetooth à la micro:bit et ne commande aucun objet. Elle sert uniquement à observer l'effet du prompt système sur la réponse du LLM. En s'aidant du document ressource (*Seq2_s3_fiche_ressource_prompt_systeme.pdf*) les élèves conservent le même prompt utilisateur, comparent plusieurs prompts systèmes, puis améliorent le prompt pour obtenir une réponse plus adaptée. Ils découvrent ainsi simplement la démarche de prompt engineering : rédiger, tester, comparer puis améliorer un prompt.

Analyse et mise en perspective (15 mn)

La mise en commun permet de formaliser la différence entre une commande directe et une intention. Avec une commande directe, l'utilisateur prononce explicitement l'action attendue. Avec une intention, le LLM doit interpréter le sens de la phrase.

Le professeur insiste sur trois points de sécurité. Premièrement, le prompt système encadre le comportement du LLM. Deuxièmement, la réponse du LLM doit être filtrée pour ne laisser passer que les commandes autorisées. Troisièmement, il ne faut pas envoyer directement toute la réponse du LLM à la micro:bit, car elle pourrait être longue, imprévisible ou non comprise par le programme.

La distinction est formulée clairement : le prompt système est la consigne de haut niveau qui cadre le comportement de l'IA ; le prompt utilisateur est la demande formulée par l'utilisateur ; le prompt engineering est le travail de conception qui consiste à rédiger, tester et améliorer cette consigne. Dans « *EduVox IA Bluetooth* », le prompt système est le résultat d'un travail de prompt engineering.

Bilan de mes recherches (10 mn)

Dans « *EduVox IA Bluetooth* », la phrase de l'utilisateur est envoyée à un LLM qui interprète l'intention de l'utilisateur. Le LLM répond par une commande simple. Cette réponse est vérifiée par un filtre afin de s'assurer qu'elle fait partie des commandes autorisées. La commande est ensuite envoyée à la carte micro:bit par Bluetooth.

Le prompt système joue le rôle d'une routine intelligente : il transforme des intentions en commandes autorisées. « *EduVox IA Bluetooth* » commande une seule chose à la fois parce que le prompt impose un seul mot. Pour piloter un objet réel, il faut garder le contrôle humain et technique sur ce qui est envoyé à la machine.

➤ Synthèse (5 mn)

L'enseignant distribue le document de structuration (*chaîne d'information IA*) et l'explicite.

Ressources pour le professeur

Fiches :

Seq2-s3-eleve_correction.pdf (+docx)
Seq2_s3_fiche_ressource_EduVox_IA_Bluetooth.docx
Seq2_s3_fiche_ressource_prompt_systeme.docx

Ressources :

Seq2_s3_EduVox_IA_Bluetooth_sans_cle.aia
Seq2_s3_EduVox_Mode_Commande_corrige.hex
Seq2_s3_EduVox_Prompt_Systeme_sans_cle.aia

Ressource professeur :

Seq2-s3_fiche_professeur.pdf (+docx)
Seq2_maquette_serrure_servo.pdf (+docx)

Structuration de connaissances :

Structuration_chaine_information_IA.docx

Ressources pour les élèves

Fiches :

Seq2-s3-eleve.pdf (+docx)

Ressources :

Seq2_s3_fiche_ressource_EduVox_IA_Bluetooth.pdf
Seq2_s3_EduVox_IA_Bluetooth_sans_cle.aia
Seq2_s3_EduVox_Mode_Commande_corrige.hex
Seq2_s3_EduVox_Prompt_Systeme_sans_cle.aia

Structuration de connaissances :

Structuration_chaine_information_IA.pdf