



BOT BUDDY ADVENTURE

Concevoir un ChatBot pour une meilleure accessibilité urbaine

thématique : intelligence artificielle et nouvelles technologies

sous-thème : engagement citoyen, gouvernance et données



Introduction

Un **agent conversationnel**, souvent appelé **chatbot** ou **assistant virtuel**, est un programme informatique conçu pour engager des conversations en langage naturel avec les utilisateurs. Il utilise divers **algorithmes et techniques de traitement du langage** pour comprendre et répondre aux requêtes ou commandes des utilisateurs, en fournissant des informations, de l'assistance, ou en effectuant des tâches telles que la **reconnaissance vocale** pour convertir les mots parlés en texte, le **traitement du langage naturel** pour comprendre l'intention de l'utilisateur, l'**analyse sémantique** pour extraire le sens du texte, la **gestion du dialogue** pour maintenir le flux de conversation ou la **synthèse vocale** pour les réponses orales. Pour développer un **agent conversationnel autonome**, il est essentiel de comprendre les différentes fonctions techniques requises et de concevoir les solutions appropriées pour chacune d'entre elles. Choisir la bonne combinaison de **reconnaissance vocale, traitement du langage naturel et synthèse vocale** permet aux utilisateurs d'interagir avec l'agent conversationnel de manière naturelle et intuitive. Dans ce protocole « **Bot Buddy Adventure** », les élèves exploreront le développement d'un agent conversationnel en identifiant et en implémentant les fonctionnalités techniques clés. Ils apprendront à utiliser la reconnaissance vocale pour convertir la voix en texte, à exploiter le traitement du langage naturel pour comprendre l'intention de l'utilisateur, et à mettre en œuvre la synthèse vocale pour la génération de réponses. Grâce à une expérience pratique avec ces technologies, les élèves acquerront une compréhension concrète de la façon dont l'intelligence artificielle permet une interaction naturelle entre l'homme et l'ordinateur.



Les solutions d'IA pour les défis urbains visent à rendre les villes plus intelligentes et accessibles. Le protocole Bot Buddy Adventure guide les élèves dans la création d'un agent conversationnel utilisant GPS, reconnaissance vocale et IA pour faciliter l'accès aux services essentiels, aux transports publics et aux ressources communautaires. Cette technologie aide à réduire la fracture numérique avec des interfaces vocales, rendant les informations urbaines accessibles à tous, y compris aux populations ayant des difficultés avec les interfaces textuelles. Le protocole encourage des discussions sur l'accessibilité urbaine et l'inclusion, en explorant comment l'IA peut aider divers groupes, comme les malvoyants et les résidents âgés, à naviguer dans leur environnement. Ces échanges permettent aux élèves de comprendre l'impact social de l'IA dans la création de villes inclusives.

Interdisciplinarité



technologie et ingénierie
art et design
physique et chimie

Objectifs de développement durable

4
ÉDUCATION
DE QUALITÉ



9
INDUSTRIE,
INNOVATION ET
INFRASTRUCTURE



10
INÉGALITÉS
RÉDUITES





L'activité en bref

Structure de l'activité

Ce protocole initie les élèves au développement d'un agent conversationnel autonome à travers une expérience d'apprentissage structurée et pratique. Le processus est divisé en quatre étapes complémentaires qui s'appuient les unes sur les autres pour créer une compréhension globale des systèmes d'IA conversationnelle.

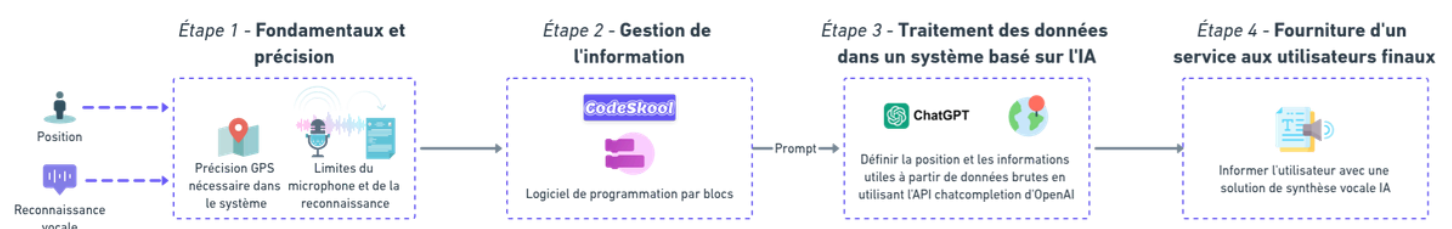
Étape 1 : Fondamentaux du système et précision : Cette phase initiale se concentre sur la compréhension des limites de précision des technologies centrales qui alimentent les agents conversationnels. Les élèves explorent les systèmes de géolocalisation et les technologies de reconnaissance vocale à travers des tests comparatifs sur différentes plateformes et environnements. En documentant systématiquement les variations de performance, ils développent une conscience critique des contraintes du monde réel qui éclaireront leurs décisions de conception, créant un état d'esprit scientifique essentiel pour un développement responsable de l'IA.

Étape 2 : Collecte de données et gestion de l'information : S'appuyant sur leur compréhension des limitations technologiques, les élèves implémentent leur premier prototype fonctionnel. Cette phase introduit le concept de variables comme la « mémoire » de leur programme, enseignant aux élèves comment capturer et stocker à la fois les coordonnées GPS et les entrées vocales. Grâce à une programmation pratique dans CodeSkool, les élèves créent la couche essentielle de collecte de données de leur agent conversationnel, expérimentant comment les concepts abstraits se traduisent en implémentation pratique.

Étape 3 - Traitement des données dans un système basé sur l'IA : Cette phase transforme les données brutes en informations significatives grâce à l'intégration de l'IA. Les élèves explorent les modèles de langage et découvrent l'art de l'ingénierie des prompts — apprenant à élaborer des instructions efficaces qui guident les réponses de l'IA. À travers des tests itératifs et un raffinement continu, ils convertissent les coordonnées techniques en noms de lieux lisibles par l'homme et développent des compétences pour créer des prompts contextuellement pertinents. Cet apprentissage expérientiel aide les élèves à comprendre à la fois les capacités et les limitations des modèles de langage.

Étape 4 : Diffusion d'informations accessibles aux utilisateurs finaux : La phase finale complète l'agent conversationnel en implémentant les capacités de sortie vocale et la conception conversationnelle. Les élèves configurent les paramètres de synthèse vocale, créent des flux de dialogue logiques et développent des prompts qui fournissent des informations pertinentes sur les environnements urbains. Cette activité culminante intègre tous les composants précédents en un assistant entièrement interactif vocalement qui peut engager une conversation naturelle, démontrant comment la technologie peut améliorer l'accessibilité et fournir des services géolocalisés précieux.

À travers cette progression, les élèves développent à la fois des compétences techniques et une réflexion critique sur les systèmes d'IA. Ils passent de la compréhension des limitations technologiques à l'implémentation, de la collecte de données de base au traitement intelligent, et finalement à la création d'expériences utilisateur accessibles. Chaque étape s'appuie naturellement sur la précédente, créant un parcours d'apprentissage cohérent qui reflète les processus de développement d'IA du monde réel tout en mettant l'accent sur les pratiques de conception responsables.



Pour bien démarrer

Durée : L'activité se déroulera sur quatre séances, chacune durant environ 30 à 60 minutes.

Niveau de difficulté : *Modéré. L'activité a été structurée en plusieurs étapes, mais ne suit pas un modèle clé en main. Elle doit être appropriée par l'enseignant.*

Matériel nécessaire : Ordinateur et appareil mobile (smartphone)

Ressources en ligne :

- Introduction au prompting : https://www.youtube.com/watch?v=8lQ9j_QoA3A
- Introduction aux modèles de langage : <https://youtu.be/K8gOvC8gyB4?si=EOvjZwRIWw9td9ww>



Glossaire

Mots-clés et concepts	Définitions
Agent conversationnel	Également connu sous le nom de chatbot, un agent conversationnel intelligent est un programme logiciel robotisé qui utilise une base de données pour communiquer avec un humain en langage naturel, soit verbalement soit par écrit.
Analyse sémantique	L'analyse sémantique permet aux ordinateurs d'interpréter le contexte correct des mots ou expressions ayant plusieurs significations, ce qui est essentiel pour la précision des applications de traitement du langage naturel basées sur le texte. En effet, plutôt que de simplement analyser les données, cette technologie va plus loin et identifie les relations entre les éléments de données.
Traitement du langage naturel	Le traitement automatique du langage naturel est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle. Il vise à créer des outils capables d'interpréter et de synthétiser du texte pour diverses applications.
Modèle de langage	Un système d'IA entraîné sur de grandes quantités de données textuelles pour comprendre et générer un langage similaire à celui des humains, capable d'effectuer des tâches comme la complétion de texte, la traduction et la réponse à des questions.
Ingénierie des prompts	La pratique consistant à élaborer des instructions d'entrée efficaces pour les modèles de langage afin d'obtenir les résultats souhaités, nécessitant une attention particulière au contexte, à la clarté et aux contraintes.
Synthèse vocale	Technologie qui convertit le texte écrit en mots parlés, utilisant l'intelligence artificielle pour générer une parole au son naturel avec un rythme et une intonation appropriés.



Protocole

Étape 1 : Fondamentaux du système et précision

Contexte et description du problème à résoudre à cette étape : Dans le protocole Bot Buddy Adventure, les élèves développent une application avec un agent conversationnel autonome pour aider à découvrir des activités locales. Ce projet vise à surmonter les défis d'accès à l'information urbaine grâce à des interfaces vocales. Lors de la première étape, « Fondamentaux du système et précision », ils analyseront les limites de précision des systèmes de positionnement géographique et de reconnaissance vocale. Les élèves apprendront à déterminer la localisation avec précision et à évaluer la fiabilité de l'interaction vocale par rapport à la saisie de texte, afin de concevoir une expérience accessible et performante.



Objectifs d'apprentissage : Comprendre les fondamentaux de la géolocalisation et du GPS. Comprendre les limites de précision et les principes fondamentaux des systèmes de géolocalisation et de la technologie GPS. Évaluer la fiabilité des solutions de reconnaissance vocale dans différents environnements et conditions d'utilisation. Développer des compétences de résolution de problèmes en travaillant sur des projets impliquant la géolocalisation et la reconnaissance vocale. Apprendre à identifier les problèmes potentiels et itérer sur la conception pour améliorer les performances et l'efficacité d'un système.

Conceptualisation

Dans cette première étape, les élèves explorent les fondements des services géolocalisés et des interfaces vocales, qui forment ensemble les **mécanismes d'entrée essentiels pour un assistant de navigation urbaine accessible**. Le projet vise à aider tous les utilisateurs à naviguer dans les environnements urbains, avec une attention particulière aux besoins d'accessibilité. Pour que cet assistant fonctionne efficacement, il doit maîtriser deux capacités fondamentales : **savoir où se trouve l'utilisateur et comprendre ce que l'utilisateur demande**. Ces capacités représentent la base d'entrée sur laquelle toutes les fonctionnalités ultérieures seront construites.

Question de recherche 1 : Conscience de la localisation

Comment peut-on déterminer avec précision la position physique d'un utilisateur dans un environnement urbain pour fournir des informations localisées et pertinentes ?

Hypothèse : En utilisant la technologie GPS, nous pouvons créer un système qui détermine de manière fiable la localisation de l'utilisateur dans les environnements urbains avec une précision suffisante pour fournir des informations locales pertinentes.

Concepts clés :

- **Fondamentaux du GPS** : Le GPS fonctionne grâce à un réseau de satellites qui transmettent des signaux contenant des informations temporelles précises. Les récepteurs calculent leur position en mesurant le temps que mettent les signaux à arriver de plusieurs satellites (triangulation).
- **Facteurs de précision du GPS** : La qualité du signal varie en fonction des conditions atmosphériques, des bâtiments environnants et de la qualité du récepteur, avec une précision typique des smartphones allant de plusieurs mètres dans des conditions optimales à des mesures moins précises dans des environnements urbains difficiles.
- **Contexte géospatial** : Comprendre comment les coordonnées brutes se rapportent à des lieux significatifs comme les rues, les quartiers et les points d'intérêt dans une ville.

Question de recherche 2 : Accessibilité vocale

Comment peut-on s'assurer que notre interface accepte les commandes vocales et les convertit efficacement en

requêtes qu'un système peut traiter ?

Hypothèse : En implémentant des directives robustes pour notre système de reconnaissance vocale, nous pouvons créer une interface vocale qui comprend de manière fiable les demandes des utilisateurs à travers divers modèles d'élocution, conditions environnementales et accents.

Concepts clés :

- **Traitement de la reconnaissance vocale** : L'entrée vocale passe par quatre étapes clés : capture audio, extraction de caractéristiques (identification des phonèmes), modélisation acoustique (association des sons aux mots) et modélisation linguistique (détermination des séquences de mots probables).
- **Considérations d'accessibilité** : Comment les interfaces vocales suppriment les barrières pour les utilisateurs ayant différentes capacités et besoins, y compris les déficiences visuelles, les limitations motrices ou les difficultés de lecture.
- **Défis environnementaux** : Comprendre comment le bruit de fond, les accents et les modèles d'élocution affectent la précision de reconnaissance et développer des stratégies pour améliorer la fiabilité.

En explorant ces deux questions en parallèle, les élèves acquièrent une compréhension plus approfondie de chaque technologie tout en voyant comment elles se complètent pour créer un service accessible et géolocalisé. Les activités d'investigation permettront aux élèves de tester leurs hypothèses à travers une expérimentation pratique avec les composants de services de localisation et de reconnaissance vocale.

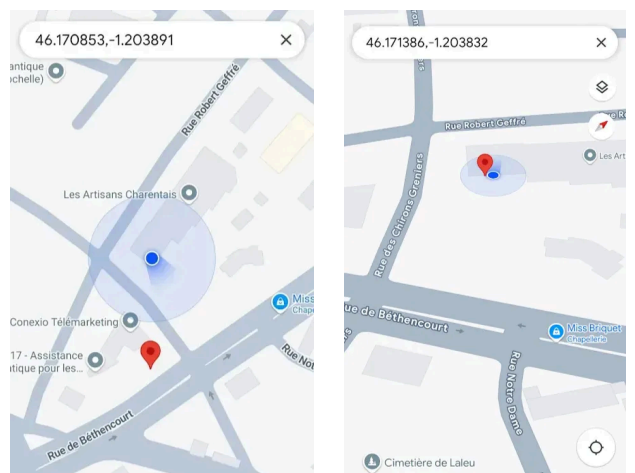
Investigation par les élèves

Cette phase d'investigation guide les élèves à travers une exploration pratique des technologies GPS et de reconnaissance vocale, les mécanismes fondamentaux du Bot Buddy Assistant Urbain.

Explorer les capacités du GPS

Dans cette première activité, les élèves exploreront comment intégrer la fonctionnalité GPS dans leur assistant virtuel en comparant différentes méthodes de géolocalisation et en considérant le niveau de précision nécessaire pour leur projet. Faites tester aux élèves plusieurs solutions GPS pour comprendre les capacités et les limites des services de localisation, indépendamment de la méthode d'implémentation spécifique.

Faites-leur accéder au GPS intégré de leur appareil via leur application de cartes (Google Maps, Apple Maps) pour comprendre comment fonctionnent les coordonnées géographiques en pratique. Demandez-leur d'enregistrer leur latitude et longitude actuelles pendant qu'ils sont en classe. Lorsque les élèves compareront leurs résultats, ils remarqueront de petites variations malgré le fait qu'ils se trouvent au même endroit.



Faites-leur expérimenter des méthodes GPS alternatives comme les services de cartographie en ligne, les applications GPS dédiées et les applications web qui utilisent l'API de géolocalisation du navigateur. Demandez-leur de comparer les résultats de localisation qu'ils obtiennent sur différents appareils. Les élèves peuvent remplir le tableau suivant avec leurs mesures et observer comment la précision de localisation varie selon les différents appareils et technologies. Cette comparaison les aidera à comprendre les considérations de fiabilité pour leur assistant virtuel.

Type d'appareil	Latitude	Longitude	Variation par rapport à la référence
GPS pour smartphone			
GPS pour tablette			
Localisation navigateur portable			
Appareil GPS dédié			

Pour la colonne « **Variation par rapport à la référence** », les élèves peuvent choisir une mesure comme point de référence (peut-être l'appareil le plus précis disponible) et calculer à quel point les autres mesures en diffèrent. Cela fournit un moyen concret de comprendre les variations de précision.

Animez une discussion sur ces variations, en vous concentrant sur leurs implications pour un agent conversationnel basé sur la localisation :

- Un assistant virtuel aidant les gens à découvrir des activités locales a-t-il besoin d'une précision millimétrique (à 1-2 mètres près) ou juste d'une conscience générale du quartier ?
- Comment la précision requise dépend-elle du cas d'usage spécifique ? Par exemple, trouver le restaurant le plus proche pourrait ne nécessiter qu'une localisation approximative, tandis que fournir des directions étape par étape nécessite une plus grande précision.
- Quels facteurs environnementaux semblent affecter la précision du GPS ? (Matériaux de construction, météo, densité urbaine)
- Comment ces limitations de précision pourraient-elles influencer la conception de votre assistant virtuel ?

Faites conclure les élèves en déterminant le niveau de précision nécessaire pour leur assistant local et en documentant les défis potentiels qu'ils pourraient devoir relever.

Explorer les limites de la reconnaissance vocale

De même, pour la reconnaissance vocale, les élèves peuvent expérimenter avec différents assistants vocaux (Siri, Google Assistant, Alexa...), la reconnaissance vocale basée sur navigateur, des logiciels de dictée dédiés, ou même les fonctionnalités de saisie vocale dans les traitements de texte. L'apprentissage important vient de l'observation de la performance de ces systèmes dans différentes conditions, pas de l'environnement de codage spécifique utilisé. Pour effectuer leur test, faites-leur créer un ensemble de phrases de test liées à la navigation urbaine que leur assistant pourrait avoir besoin de comprendre, telles que : « Où est le parc le plus proche ? », « Comment puis-je aller au musée ? », « Y a-t-il des cafés dans les environs ? », « À quelle heure ferme le cinéma ? ». **Expérimentez avec différentes conditions d'élocution** : parler clairement versus naturellement, différentes distances du microphone, avec et sans bruit de fond, avec différents locuteurs (camarades de classe avec des voix/accents variés). **Analysez les résultats pour identifier des modèles** : Quels types de phrases sont reconnus le plus précisément, quels mots ou sons sont fréquemment mal interprétés, quelles conditions d'élocution produisent les meilleurs résultats ? Enregistrez les résultats dans un tableau simple montrant :

Appareil	Phrase prononcée	Ton utilisé	Phrase reconnue par le système	Précision
<i>Précisez le système utilisé pour le test</i>	<i>Entrez la phrase exacte que vous avez prononcée</i>	<i>Notez si vous avez parlé lentement, rapidement, avec un accent</i>	<i>Écrivez ce que le système a entendu</i>	<i>Évaluez la précision (Bon/Correct/Médiocre)</i>
Siri				
Hé Google				
Alexa				
Cortana				
Logiciel de dictée				

Après avoir collecté les données, guidez les élèves dans l'analyse des résultats pour identifier des modèles : Quels types de phrases sont reconnus le plus précisément sur tous les systèmes ? Quels mots ou sons sont fréquemment mal interprétés ? Quelles conditions d'élocution produisent les meilleurs et les pires résultats ? Certains systèmes

fonctionnent-ils mieux pour des types spécifiques de phrases ? Comment les accents ou les modèles d'élocution affectent-ils la précision de reconnaissance ?

Animez une discussion sur les implications de ces découvertes pour concevoir des interfaces vocales :

- Comment pourriez-vous concevoir votre assistant pour minimiser les malentendus ?
- Quelles stratégies pourraient aider les utilisateurs à parler de manière que le système puisse comprendre ?
- Comment pourriez-vous gérer les résultats de reconnaissance incertains ?
- Quelles méthodes de secours pourraient aider lorsque la reconnaissance vocale échoue ?

Cette activité de test aide les élèves à comprendre que la technologie de reconnaissance vocale a des limitations inhérentes qu'ils devront aborder lors de la conception de leur assistant virtuel. En identifiant les erreurs de reconnaissance communes sur toutes les plateformes, ils peuvent développer des stratégies pour améliorer la fiabilité, comme concevoir des étapes de confirmation pour les commandes ambiguës ou fournir un retour visuel sur ce que le système a compris.

Restitution et réflexion



- **Connaissances mobilisées :** Dans la première étape de l'aventure Bot Buddy, les élèves ont établi les bases de leur agent conversationnel en analysant la précision et les limites des technologies GPS et de reconnaissance vocale. Ils ont testé diverses technologies sur le terrain, découvrant que la précision géographique varie selon les appareils et l'environnement. Ils ont réalisé que des facteurs comme les matériaux de construction affectent la fiabilité du positionnement. De plus, leurs essais sur la reconnaissance vocale ont révélé l'impact de la diction, des bruits environnants et des accents sur la qualité de reconnaissance. En identifiant les erreurs d'interprétation, ils ont mieux compris les défis à relever pour concevoir des interfaces vocales efficaces, ce qui a guidé leurs choix de conception basés sur l'expérience pratique.
- **Réflexion sur la mise en œuvre en classe :** Cette étape réussit en associant exploration concrète et analyse comparative. Tester le GPS dans divers environnements rend les concepts abstraits tangibles et développe des compétences d'investigation scientifique. La collecte de données favorise l'apprentissage entre pairs. Les tests comparatifs incitent à réfléchir sur les limites technologiques, cultivant un regard critique. Pour l'évaluation, concentrez-vous sur la capacité des élèves à tirer des conclusions basées sur des preuves et à expliquer l'impact de ces découvertes sur leurs choix de conception.
- **Résultats d'apprentissage généraux :** Les élèves ont appris à concevoir des protocoles de test pour identifier les forces et faiblesses technologiques, développant ainsi des compétences d'analyse critique. Ils comprennent l'importance de la précision des données d'entrée et adoptent un état d'esprit d'ingénieur, reconnaissant les contraintes tout en cherchant des solutions optimales pour leur projet d'agent conversationnel.

Pour approfondir l'engagement des élèves avec les concepts abordés dans cette étape, envisagez d'explorer ces questions à travers des discussions ou des écrits de réflexion :

1. Comment l'environnement physique affecte-t-il la précision du GPS, et quelles sont les implications pour les services géolocalisés ?
2. Quels schémas émergent dans les performances de reconnaissance vocale selon différentes conditions ? Considérez des facteurs comme le bruit de fond, les accents et les modèles d'élocution, et discutez des solutions de conception potentielles pour améliorer l'accessibilité.
3. Comment les technologies GPS et de reconnaissance vocale peuvent-elles fonctionner ensemble efficacement tout en préservant la vie privée des utilisateurs ? Quelles considérations de sécurité doivent être prises en compte lors de la collecte et du stockage de ces données sensibles ?
4. Qui bénéficie le plus de la combinaison du GPS et des interfaces vocales, et comment pouvons-nous nous assurer que ces technologies restent accessibles à des groupes d'utilisateurs diversifiés ?

Étape 2 : Collecte de données et gestion de l'information



Contexte et description du problème à résoudre à cette étape : Grâce à l'étape 1, les élèves sont capables de comprendre les limites de précision des technologies GPS et de reconnaissance vocale. Maintenant, dans l'étape 2, ils vont implémenter un système de base pour collecter ces deux données d'entrée fondamentales. Les élèves vont appliquer leurs connaissances en créant un prototype fonctionnel dans CodeSkool qui capture les informations de localisation et les entrées vocales tout en tenant compte des limitations du monde réel qu'ils ont identifiées. Cette implémentation formera la couche essentielle de collecte de données de leur Bot Buddy Assistant Urbain, leur permettant de stocker les informations avant de les traiter dans les étapes ultérieures.

Objectifs d'apprentissage : Implémenter une fonctionnalité GPS de base pour récupérer et stocker les coordonnées de localisation de l'utilisateur. Créer une interface de reconnaissance vocale simple qui convertit la parole en texte. Comprendre les types de variables et leur rôle dans le stockage d'informations. Implémenter des flux de traitement de données qui relient la conscience de localisation aux requêtes des utilisateurs. Développer des compétences techniques pour travailler avec les capteurs d'appareils et les API externes.

Conceptualisation

Dans cette deuxième étape, les élèves passent de la compréhension à l'implémentation, créant un prototype fonctionnel qui collecte les deux entrées fondamentales de leur agent conversationnel. S'appuyant sur leur exploration des limitations technologiques de l'étape 1, ils vont maintenant créer un système opérationnel qui capture réellement la localisation de l'utilisateur et les entrées vocales.

Question de recherche 1 : Collecte de données de localisation

Comment peut-on implémenter un système qui capture et stocke efficacement la position physique d'un utilisateur pour des usages ultérieurs ?

Hypothèse : En utilisant l'extension Capteurs de Téléphone de CodeSkool et en implémentant des variables appropriées, les élèves peuvent créer un système qui capture et stocke les coordonnées de localisation avec une fiabilité suffisante pour un assistant de navigation urbaine.

Concepts clés :

- **Variables** : Constructions de programmation qui servent de conteneurs pour le stockage de données, similaires à des boîtes étiquetées dans la mémoire de l'ordinateur. Les variables permettent aux programmes de stocker temporairement et de rappeler des informations comme les coordonnées, les mesures ou le texte dans la RAM (Mémoire Vive). Tout comme la mémoire à court terme humaine retient temporairement les informations, la RAM stocke les données des variables uniquement pendant que l'ordinateur est allumé, contrairement au stockage permanent sur un disque dur.
- **Accès aux capteurs** : Méthodes pour accéder aux capacités GPS d'un appareil via des interfaces de programmation, y compris la gestion de la possible indisponibilité.
- **Retour utilisateur** : Techniques pour communiquer les données de localisation et leur fiabilité aux utilisateurs, les aidant à comprendre les limitations potentielles.

Question de recherche 2 : Collecte d'entrées vocales

Comment peut-on implémenter un système qui capture efficacement les entrées vocales et les convertit en texte tout en tenant compte des limitations de reconnaissance ?

Hypothèse : En utilisant l'extension Parole vers Texte de CodeSkool et en implémentant des mécanismes de retour appropriés, les élèves peuvent créer un système qui capture et stocke de manière fiable les requêtes des utilisateurs dans un format adapté au traitement ultérieur.

Concepts clés :

- **Capture de parole** : Méthodes pour activer et traiter les entrées vocales via des interfaces de programmation, incluant la reconnaissance de commandes vocales et la transcription.
- **Stockage et analyse de texte** : Approches pour stocker la parole transcrite dans des variables et identifier les mots-clés ou phrases pour déterminer l'intention de l'utilisateur. Par exemple, rechercher des termes spécifiques comme « où », « comment » ou « quoi » peut aider à catégoriser le type de demande.
- **Traitement des requêtes utilisateur** : Techniques pour analyser et catégoriser les requêtes des utilisateurs en identifiant les types de commandes (par exemple, demandes de localisation, requêtes d'information) et en extrayant les mots-clés pertinents pour générer des réponses appropriées.
- **Retour de reconnaissance** : Techniques pour confirmer la compréhension, stocker les informations pertinentes dans des variables pour un usage futur, et gérer les erreurs potentielles de reconnaissance à travers des invites de vérification.

En implémentant ces deux mécanismes de collecte de données, les élèves créent la base essentielle de leur agent conversationnel. La capacité à capturer et stocker à la fois la localisation et les entrées vocales permet tout le traitement, l'analyse et la génération de réponses ultérieurs. Ces composants forment l'interface critique entre le monde physique et l'assistant numérique, déterminant quelles informations sont disponibles pour aider les utilisateurs à naviguer dans leur environnement urbain.

Investigation par les élèves

Dans cette étape, les élèves vont implémenter les capacités essentielles de collecte de données nécessaires pour leur Bot Buddy Assistant Urbain. Dans notre exemple, nous avons utilisé le logiciel de programmation par blocs CodeSkool (<https://ide.codeskool.cc/>). CodeSkool est particulièrement adapté à ce projet car il combine l'intégration d'Open AI avec des extensions de capteurs de téléphone et des capacités de synthèse vocale.



Plusieurs alternatives existent à CodeSkool. Scratch propose des extensions de géolocalisation qui fournissent des fonctionnalités similaires dans un environnement par blocs. MIT App Inventor inclut des composants de localisation intégrés spécialement conçus pour le développement d'applications mobiles. Pour les classes ayant une expérience de programmation plus avancée, les bibliothèques Python comme geocoder ou geopy offrent des services de localisation sophistiqués qui peuvent être intégrés dans des applications plus larges.

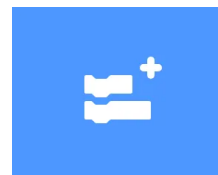
Entrer dans CodeSkool et stocker les informations GPS à l'aide de variables

La capacité à stocker et mémoriser des informations est l'un des concepts les plus fondamentaux en programmation. Les variables agissent comme la « **mémoire** » de votre programme, lui permettant de **retenir des données importantes** comme la localisation de l'utilisateur qui peuvent être utilisées dans différentes parties de votre application.

Dans cette première activité, les élèves apprendront comment **créer des variables pour stocker les coordonnées GPS**, donnant à leur agent conversationnel la capacité de se « **souvenir** » de l'endroit où se trouve l'utilisateur.

- Pour implémenter la fonctionnalité GPS, les élèves doivent commencer par **créer un nouveau projet et accéder à l'éditeur** dans CodeSkool.
- Avant d'accéder à toute donnée GPS, ils doivent établir des **emplacements mémoire** dans leur programme où cette information sera stockée. Ils doivent commencer par aller dans la section « **Variables** » de la palette de blocs et cliquer sur « **Créer une variable** ». Les élèves doivent créer une variable nommée « **Latitude** » pour stocker la position nord-sud de l'utilisateur. Cette variable agira comme un conteneur dédié qui mémorise cette information spécifique tout au long de leur programme. Puis, ils doivent retourner dans la section « **Variables** » et créer une seconde variable nommée « **Longitude** » pour stocker la position est-ouest. Ensemble, ces deux variables fourniront une position géographique complète qui persiste en mémoire.

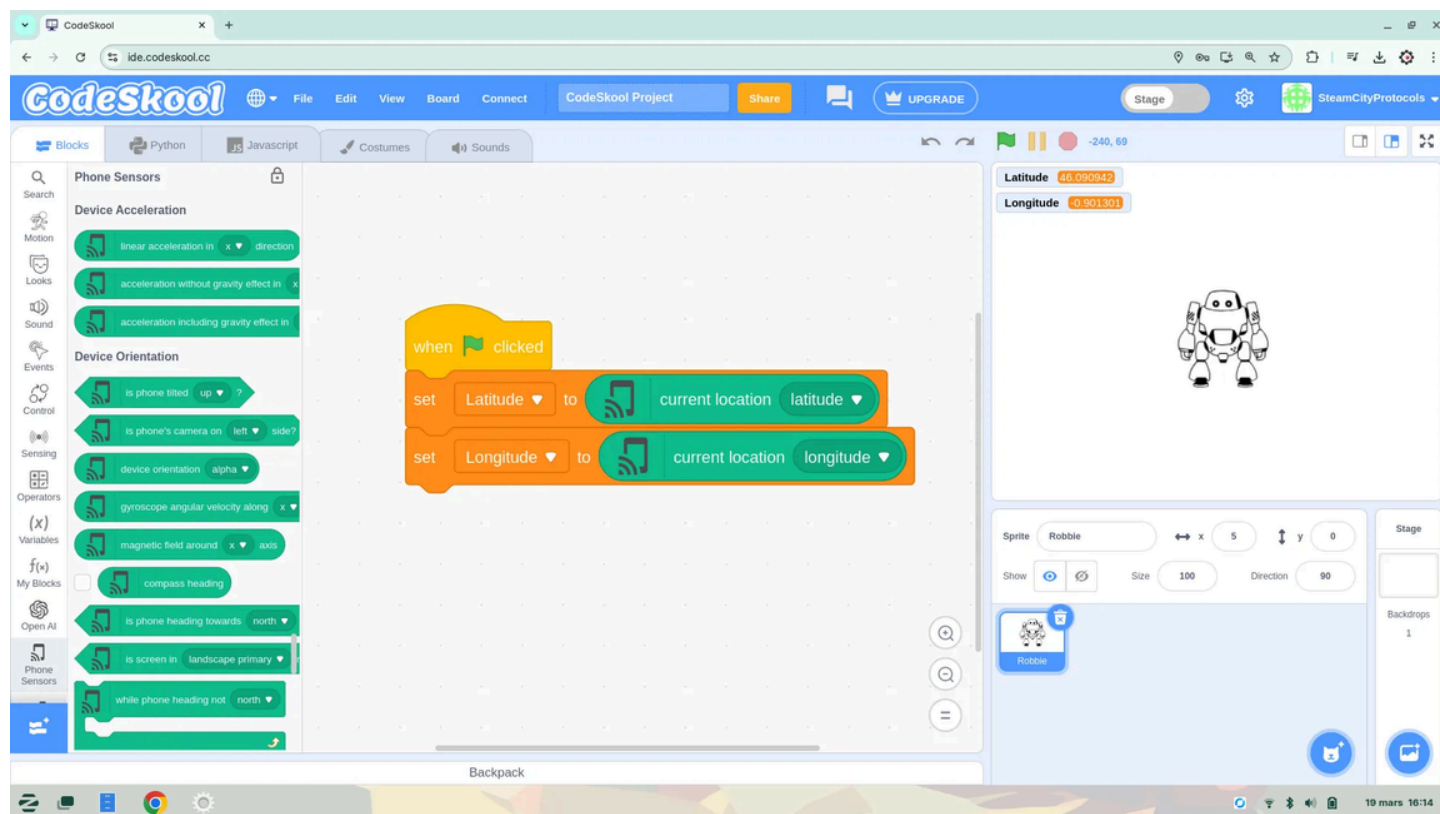
- Maintenant que les élèves ont créé les conteneurs mémoire pour leurs données de localisation, ils doivent **accéder aux capacités GPS de l'appareil** pour remplir ces variables avec de vraies coordonnées. Depuis l'interface principale, ils doivent accéder au bouton « **Extensions** » situé tout à la fin de la barre d'outils. Dans la bibliothèque d'extensions, rechercher « **Phone sensors** » et ajouter cette extension à leur projet pour accéder à la fonctionnalité GPS.



- Après avoir ajouté l'extension, ils peuvent trouver des **blocs liés au GPS** dans la nouvelle catégorie qui apparaît dans leur palette de blocs. Ils peuvent rechercher la sous-catégorie de blocs appelée « **GPS location** » dans laquelle ils devraient trouver le bloc « **current location** » pour la longitude et la latitude.

À partir de maintenant, les élèves ont identifié tous les blocs et variables nécessaires pour créer le programme, grâce à des étapes simples :

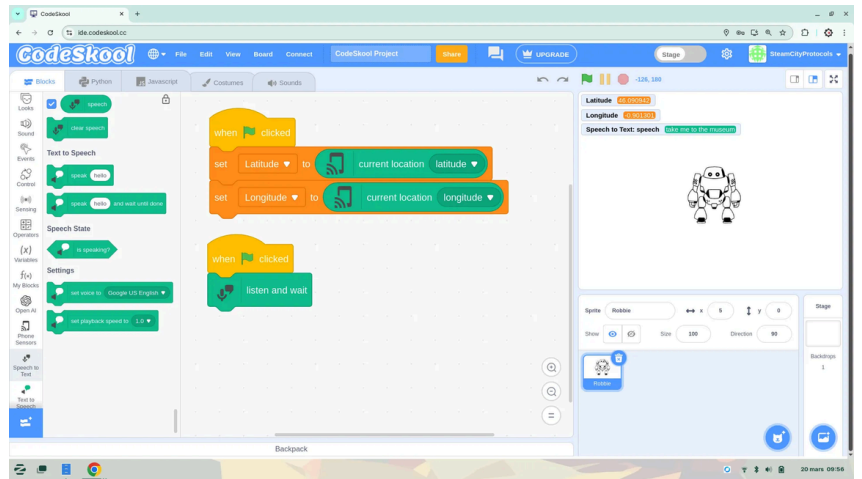
1. Pour déclencher le processus de collecte de données, ajouter un bloc d'événement « **Quand le bouton est cliqué** » à votre espace de travail. Ceci servira de point de départ pour capturer la localisation de l'utilisateur.
2. Depuis la section « **Variables** », faire glisser deux blocs « **mettre [variable] à [valeur]** » et les connecter à votre événement bouton. Ces blocs sont le mécanisme par lequel l'information passe des capteurs vers la mémoire de votre programme.
3. Configurer le premier bloc pour définir « **Latitude** » et le second pour définir « **Longitude** ». La sélection de variable démontre que votre programme a maintenant des emplacements mémoire spécifiques prêts à contenir différents types d'informations.
4. Connecter les blocs « **position actuelle** » appropriés de l'extension Capteurs de Téléphone à chaque **valeur de variable**. Sélectionner « **latitude** » pour la première variable et « **longitude** » pour la seconde. Quand ces blocs s'exécutent, ils captureront les lectures de capteurs et les stockeront dans la mémoire de votre programme.
5. Cliquer sur le **drapeau vert dans le panneau de test** pour vérifier que vos variables stockent avec succès l'information.



Ce programme fonctionnera soit sur l'ordinateur en utilisant la localisation de la machine, soit via un appareil mobile grâce à l'extension Capteurs de Téléphone.

Implémenter et tester la reconnaissance vocale

En s'appuyant sur le programme créé, les élèves devront implémenter la seconde fonctionnalité pour la reconnaissance vocale. Sur le même éditeur, les élèves doivent retourner au bouton « **Extensions** » à la fin de la barre d'outils et chercher « **Speech to Text** » dans la bibliothèque d'extensions pour l'ajouter au projet.



Après avoir ajouté l'extension, ils peuvent trouver des **blocs liés à la reconnaissance vocale** dans la nouvelle catégorie qui apparaît dans leur palette de blocs. Ils doivent chercher le bloc « **Écouter et attendre** ». Pour plus de clarté dans les blocs, faire ajouter aux élèves un second bloc « **Quand le bouton est cliqué** » pour identifier les deux fonctionnalités de base du système.

Une fois que les élèves ont implémenté la fonctionnalité de reconnaissance vocale de base, ils peuvent réutiliser la même méthode de test qu'à l'étape 1 pour évaluer systématiquement sa précision afin de comprendre ses capacités et ses limitations. Ils peuvent réutiliser les phrases de test et compléter leur tableau avec les capacités de leur propre système :

Phrase prononcée	Reconnaissance du système	Précision
<i>La phrase formulée par les élèves</i>	<i>Ce que le système transcrit de votre discours</i>	<i>Évaluation de la compréhension (Bonne / Passable / Mauvaise)</i>
« Où est le parc le plus proche ? »		
« Emmène-moi au musée »		
« Y a-t-il un café à proximité ? »		
« À quelle heure est-il pour le cinéma ? »		

Restitution et réflexion



- **Connaissances mobilisées :** Dans cette deuxième étape du protocole Bot Buddy Adventure, les élèves sont passés de la compréhension à la mise en pratique, créant un prototype fonctionnel qui collecte les deux entrées fondamentales de leur agent conversationnel : les données de localisation et les requêtes vocales. Ils ont appris à accéder aux capteurs d'appareil via les extensions de CodeSkool, à stocker les informations dans des variables, et à effectuer une analyse de base des demandes des utilisateurs. Les élèves ont acquis une expérience pratique en travaillant avec les variables comme conteneurs de stockage pour différents types d'informations. Ils comprennent comment ces variables permettent à un programme de se souvenir des saisies utilisateur et des relevés de capteurs, créant une continuité dans l'expérience utilisateur. Le concept de variables comme « mémoire » de leur programme fournit une base pour une gestion de données plus complexe dans les étapes futures.
- **Réflexion sur la mise en œuvre en classe :** Cette étape d'implémentation fonctionne mieux avec une approche pratique et exploratoire où les élèves peuvent tester leur code fréquemment. Envisagez de faire travailler les élèves en binômes, avec un élève qui énonce des requêtes pendant que l'autre observe comment le système répond. Pour résoudre les problèmes techniques courants, vous pouvez créer un document partagé où les élèves peuvent noter les difficultés qu'ils rencontrent et les solutions qu'ils découvrent - cela peut également servir pour les étapes suivantes du protocole. Cette approche collaborative de débogage aide toute la classe à surmonter les obstacles plus efficacement.
- **Résultats d'apprentissage :** À travers cette étape, les élèves ont développé plusieurs compétences et connaissances importantes. Ils ont appris à accéder aux capteurs d'appareil par programmation, capturant des données du monde réel comme la localisation et les entrées vocales. Ils comprennent comment stocker différents types d'informations dans des variables, créant un programme doté d'une « mémoire » qui peut utiliser ces informations dans différentes fonctions.

Pour approfondir l'engagement des élèves avec les concepts abordés dans cette étape, envisagez d'explorer ces questions à travers des discussions ou des écrits de réflexion :

1. Comment les variables permettent-elles à votre programme de « se souvenir » d'informations, et quelles données supplémentaires pourriez-vous avoir besoin de stocker pour rendre votre agent conversationnel plus utile et sensible au contexte ?
2. Quels défis avez-vous rencontrés lors de l'implémentation du GPS et de la reconnaissance vocale, et comment votre compréhension de l'étape 1 vous a-t-elle aidé à relever ces défis ?
3. Comment votre analyse des requêtes pourrait-elle être améliorée pour mieux comprendre les intentions des utilisateurs et gérer les situations où le GPS ou la reconnaissance vocale fonctionnent mal ?
4. Comment la capacité à stocker des informations dans des variables crée-t-elle de nouvelles possibilités pour les types d'assistance que votre agent conversationnel pourrait fournir ?

Étape 3 - Traitement des données dans un système basé sur l'IA



Contexte et description du problème à résoudre à cette étape : Grâce aux étapes 1 et 2, les élèves ont déjà exploré les limites de précision des technologies de localisation et vocales, et implémenté un système de base qui collecte et stocke ces données à l'aide de variables. Maintenant, à l'étape 3, ils font face à un nouveau défi : comment traiter ces données brutes pour générer des réponses significatives. Bien qu'ils aient des coordonnées de latitude et longitude et des requêtes vocales transcrites, celles-ci seules ne fournissent pas les informations contextuelles dont les utilisateurs ont besoin. Dans cette étape, les élèves exploreront comment l'intelligence artificielle, spécifiquement les modèles de langage, peut transformer ces données brutes en informations personnalisées, localisées et actualisées.

Objectifs d'apprentissage : Comprendre les principes fondamentaux des modèles de langage et comment ils traitent l'information. Apprendre les bases de l'ingénierie des prompts pour communiquer efficacement avec les systèmes d'IA. Développer des compétences pour intégrer les données de géolocalisation et les requêtes utilisateur dans des prompts bien structurés. Implémenter un système qui relie leur code de collecte de données aux capacités d'IA. Évaluer de manière critique les réponses d'IA pour leur précision et leur pertinence par rapport aux besoins des utilisateurs.

Conceptualisation

Dans cette troisième étape, les élèves passent de la collecte de données à l'analyse des données et à l'intégration de l'intelligence. Ayant établi les entrées fondamentales de conscience de localisation et de compréhension vocale, ils explorent maintenant comment l'intelligence artificielle peut transformer ces données brutes en informations significatives et contextuelles pour les utilisateurs. Maintenant que leur assistant peut capturer la localisation et la voix, il doit développer **l'intelligence pour fournir des réponses pertinentes et utiles**. Cette intégration d'IA représente le « cerveau » qui traitera les entrées et générera des sorties appropriées.

Question de recherche 1 : Intégration de l'IA

Comment peut-on utiliser les modèles de langage pour transformer les données brutes de localisation et de requête en informations personnalisées et contextuellement pertinentes sur les environnements urbains ?

Hypothèse : En élaborant des prompts bien structurés qui incorporent à la fois le contexte de localisation de l'utilisateur et les spécificités de sa requête, nous pouvons exploiter les modèles de langage pour générer des réponses pertinentes et précises sur les activités locales et les points d'intérêt. Basé sur l'analyse de données historiques et les retours des utilisateurs, l'agent conversationnel peut également personnaliser ses recommandations au fil du temps.

Concepts clés :

- **Modèles de langage** : Systèmes d'IA entraînés sur de vastes ensembles de données textuelles qui peuvent comprendre et générer du texte similaire à celui des humains basé sur les entrées qu'ils reçoivent.
- **Ingénierie des prompts** : La pratique d'élaborer des textes d'entrée (prompts) qui guident efficacement les modèles de langage pour produire les sorties désirées.
- **Compréhension contextuelle** : Comment les modèles de langage interprètent l'information basée sur le contexte environnant fourni dans le prompt.

Question de recherche 2 : Conception de prompts

Comment peut-on formuler des prompts efficaces qui communiquent clairement à la fois le contexte de localisation et l'intention de l'utilisateur aux modèles de langage ?

Hypothèse : En développant un format de prompt structuré qui inclut systématiquement des informations spécifiques, nous pouvons créer des prompts qui produisent de manière cohérente des réponses pertinentes et

géolocalisées de la part des modèles de langage.

Concepts clés :

- **Structure de prompt** : Organiser l'information dans les prompts selon un format clair et cohérent que les modèles de langage peuvent interpréter de manière fiable.
- **Intégration du contexte géographique** : Méthodes pour incorporer l'information de localisation d'une manière qui aide l'IA à comprendre les relations spatiales.
- **Clarification d'intention** : S'assurer que l'IA interprète correctement la demande de l'utilisateur dans le contexte géographique.

En explorant ces questions, les élèves développeront une compréhension de comment l'intelligence artificielle peut combler le fossé entre la collecte de données brutes et l'assistance significative aux utilisateurs. Le modèle de langage devient la couche de traitement cruciale qui transforme les coordonnées et les requêtes vocales en informations utiles sur les environnements urbains, créant un agent conversationnel véritablement intelligent.

Investigation par les élèves

Dans cette phase d'investigation, les élèves exploreront comment intégrer les capacités d'IA dans leur Bot Buddy Assistant Urbain. S'appuyant sur le système de collecte de données qu'ils ont créé à l'étape 2, ils vont maintenant ajouter la couche d'intelligence qui traite cette information pour générer des réponses pertinentes.

Comprendre les grands modèles de langage

Avant d'implémenter l'intégration d'IA, les élèves doivent développer une **compréhension de base du fonctionnement des modèles de langage**. Commencez par une brève introduction aux modèles de langage comme systèmes d'IA entraînés sur de vastes ensembles de données textuelles qui peuvent comprendre et générer du texte similaire à celui des humains. Faites mener aux élèves une recherche rapide sur les modèles de langage en leur fournissant des ressources recommandées telles qu'une vidéo suggérée (par exemple [Comment fonctionnent les modèles de langage](#)), des articles simples expliquant les principes des modèles de langage ou des exemples d'interactions avec des modèles de langage. Cette base aidera les élèves à comprendre pourquoi l'ingénierie des prompts est importante et comment elle influence la qualité des réponses d'IA.

Première intégration d'IA - Conversion de localisation en ville

Dans cette étape, les élèves vont implémenter leur première intégration d'IA en transformant des coordonnées GPS brutes en un nom de ville significatif. Cette fonctionnalité sert un objectif important : elle convertit des données techniques (coordonnées numériques) en informations lisibles par l'homme qui sont plus intuitives et utiles tant pour le système que pour ses utilisateurs. Cette conversion servira également de base pour des prompts plus complexes dans les activités suivantes, car inclure le nom de ville dans les futurs prompts aidera l'IA à fournir des informations plus pertinentes contextuellement.

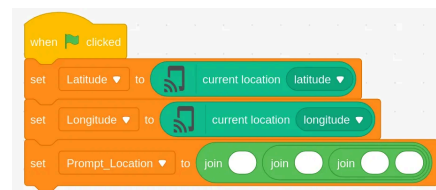
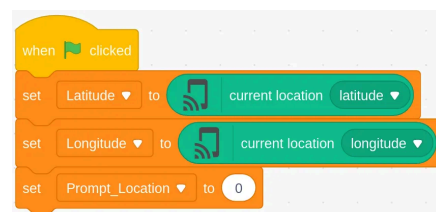
1. Préparer les variables

Pour implémenter cette conversion coordonnées-vers-ville dans CodeSkool, nous encourageons les élèves à préparer leur programme en utilisant diverses **variables** qu'ils peuvent ensuite activer quand ils en ont besoin. Ils peuvent d'abord créer une nouvelle variable appelée « **Location** » qui va **stocker le nom de ville** correspondant aux coordonnées GPS et une seconde appelée « **Prompt_Location** », qui contiendra la **structure de la question à poser au modèle d'IA**.

2. Créer la structure du prompt

La variable « **Prompt_Location** » devrait permettre de transformer les variables latitude et longitude déjà collectées par le programme en nom de ville. Pour structurer clairement ce prompt :

- Dans le même bloc d'événement « **Quand le bouton est cliqué** » où ils récupèrent les coordonnées GPS, les élèves doivent ajouter un bloc « **mettre [LocationPrompt] à [valeur]** » qui sera utilisé pour définir la structure du prompt.
- Ils doivent ensuite définir dans la zone [valeur], la structure du prompt. Commencez par demander aux élèves comment ils **formuleraient naturellement leur demande** en proposant des prompts à voix haute. Cela peut être par exemple « **Dites-moi quelle ville se trouve à cette latitude et longitude** ». Une fois qu'ils ont plusieurs idées, testons-les dans le programme.
- Pour combiner texte et variables en un prompt complet, les élèves auront besoin d'utiliser les blocs « **joindre [valeur] [valeur]** » de la section « Opérateurs ». Parce que nous devons intégrer deux variables (latitude et longitude) avec du texte, les élèves auront besoin d'utiliser des blocs de jointure imbriqués :

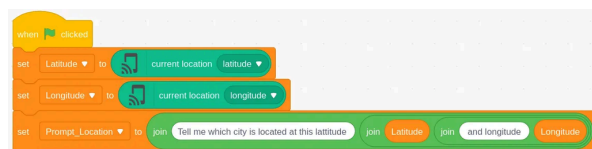
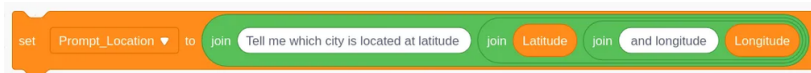


[Dis-moi quelle ville se trouve à cette latitude] + [variable « Latitude »] + [et longitude] + [variable « Longitude »]



Cela nécessite trois blocs « **regrouper [valeur] [valeur]** » imbriqués ensemble. Cette structure d'imbrication est nécessaire car les blocs de jointure ne peuvent combiner que deux éléments à la fois. En les imbriquant, nous pouvons construire une phrase complète qui incorpore plusieurs variables et segments de texte :

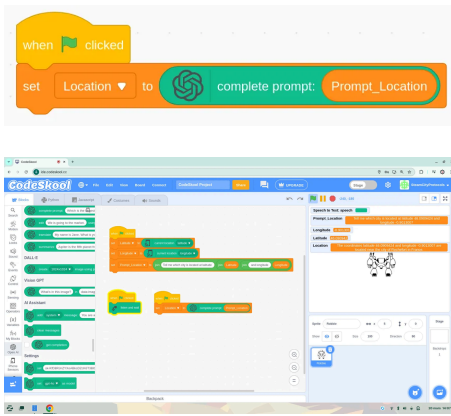
- **Premier bloc de jointure** : Valeur 1 : Le texte « Dis-moi quelle ville se trouve à la latitude » (notez l'espace à la fin) / Valeur 2 : Le deuxième bloc de jointure
- **Deuxième bloc de jointure** : Valeur 1 : La variable « Latitude » / Valeur 2 : Le troisième bloc de jointure
- **Troisième bloc de jointure** : Valeur 1 : Le texte « et longitude » (notez les espaces avant et après) / Valeur 2 : La variable « Longitude »



3. Implémenter l'intégration d'IA

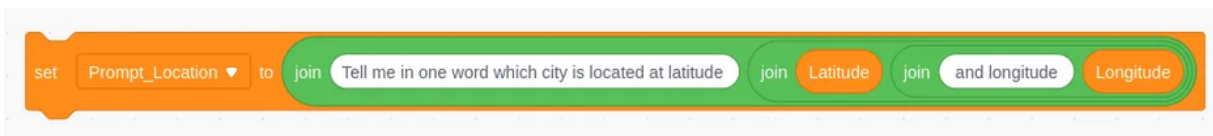
Laissez les élèves tester leur prompt. Ils doivent d'abord créer un nouveau bloc « **Quand le bouton est cliqué** » pour plus de clarté et d'organisation dans leur programme. Cela sépare la fonctionnalité de collecte de données du traitement d'IA. Dans ce bloc, ils doivent :

- Ajouter un bloc « **mettre [variable] à [valeur]** »
- Définir la variable à « **Location** »
- Retourner à l'onglet « **Extensions** » et ajouter l'extension « **Open AI** » à leur projet
- Trouver le bloc « **compléter prompt : [prompt]** » dans l'extension Open AI
- Placer ce bloc **comme valeur** pour « mettre Location à [valeur] »
- Remplacer la valeur « prompt » par défaut par leur variable « **Prompt_Location** »

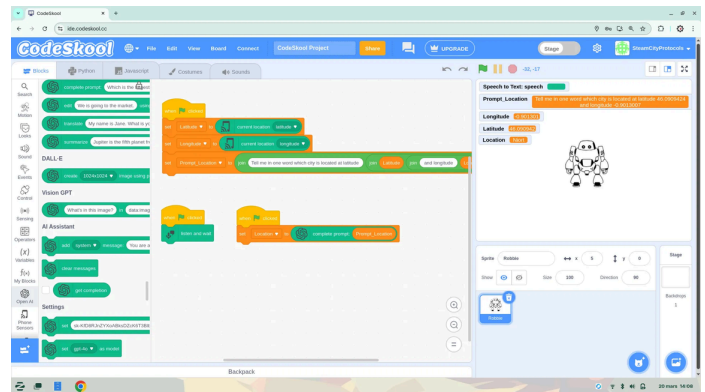


Faites cliquer les élèves sur le drapeau vert pour tester leur programme. Ils observeront que l'IA fournit typiquement des réponses verbeuses comme « La ville située à la latitude 48,8566 et longitude 2,3522 est Paris, France. Paris est la capitale et la ville la plus peuplée de France. » Cette verbosité, bien qu'informative, peut ne pas être idéale pour leur agent conversationnel qui a besoin juste du nom de ville comme composant pour les prompts suivants.

Les élèves doivent maintenant affiner leur prompt pour demander seulement l'information spécifique nécessaire. Ils peuvent modifier Prompt_Location pour être plus précis : « **Dis-moi en un mot quelle ville se trouve à la latitude [variable Latitude] et longitude [variable Longitude]** ».



En testant et affinant leur prompt, les élèves apprennent un principe important de l'ingénierie des prompts : **être spécifique à la fois sur l'information demandée et le format de réponse désiré**. Ce processus itératif démontre comment les prompts peuvent être optimisés pour produire précisément l'information nécessaire, enseignant aux élèves que l'interaction efficace avec l'IA nécessite souvent plusieurs raffinements pour atteindre les résultats souhaités.



Comprendre les limites de l'IA dans l'identification de localisation

Lors des tests sur ordinateurs plutôt que sur appareils mobiles, les élèves pourraient remarquer que les localisations retournées ne sont pas toujours précises ou exactes. Cela présente un excellent moment d'enseignement sur les limitations de l'IA et de la géolocalisation. Les navigateurs de bureau fournissent souvent des localisations approximatives basées sur les adresses IP ou les informations réseau plutôt que le GPS, résultant en des coordonnées moins précises. Le modèle d'IA pourrait alors identifier des villes proches ou des zones métropolitaines plutôt que des localisations exactes. De plus, certaines coordonnées peuvent tomber dans des zones entre villes ou dans des régions où le modèle d'IA a moins de données d'entraînement. Dans ces cas, le modèle pourrait fournir la localisation reconnue la plus proche, un nom de région au lieu d'une ville, ou occasionnellement même faire une détermination incorrecte. Faites comparer aux élèves les résultats des tests sur ordinateur versus les tests sur mobile pour comprendre ces différences. Cela les aide à apprécier à la fois les limitations technologiques (géolocalisation de navigateur vs GPS) et les limitations d'IA (couverture des données d'entraînement, granularité géographique). Ces observations renforcent l'importance de la pensée critique lors du travail avec les systèmes d'IA et le besoin de vérifier les informations importantes, spécialement quand des données de localisation précises sont cruciales pour la fonctionnalité de l'application.



Restitution et réflexion



- **Connaissances mobilisées** : Dans cette troisième étape, les élèves sont passés de la collecte de données de base au traitement intelligent des données en utilisant l'IA. Ils ont appris à transformer des coordonnées brutes en noms de lieux significatifs grâce aux modèles de langage, expérimentant à la fois les forces et les limitations des systèmes d'IA. Les élèves ont découvert les atouts clés des modèles de langage : leur capacité à analyser le contexte, exploiter des données d'entraînement diversifiées, générer un raisonnement de base, et fournir des solutions créatives aux problèmes. Ils ont également rencontré des limitations telles que les biais potentiels, les exigences de calcul, et les « hallucinations » ou inexactitudes occasionnelles. Plus important encore, les élèves ont développé des compétences pratiques en ingénierie des prompts - apprenant que la communication efficace avec l'IA nécessite spécificité, clarté et structure. Ils ont vu comment différentes formulations de prompts produisent des résultats radicalement différents, renforçant l'importance d'une conception soignée dans les interactions avec l'IA.
- **Réflexion sur la mise en œuvre en classe** : Cette étape peut être mise en œuvre comme une exploration progressive, en commençant par des interactions d'IA simples avant de construire vers des implémentations plus complexes. L'approche itérative de raffinement des prompts encourage naturellement l'expérimentation et la pensée critique. Les discussions de groupe comparant différentes formulations de prompts et leurs résultats peuvent être particulièrement enrichissantes. Pour l'évaluation, concentrez-vous sur le raisonnement des élèves concernant leurs choix de conception de prompts plutôt que sur la seule fonctionnalité, car cela démontre une compréhension plus profonde des concepts impliqués.
- **Résultats d'apprentissage** : À travers cette investigation, les élèves ont appris que la communication avec les systèmes d'IA nécessite des approches structurées même en utilisant le langage naturel. Ils ont expérimenté l'impact de la conception itérative en révisant les prompts pour améliorer les réponses, et ont commencé à développer des compétences d'évaluation critique des sorties d'IA. Ces compétences préparent les élèves non seulement pour les étapes suivantes du développement de leur agent conversationnel, mais aussi pour un engagement réfléchi avec les technologies d'IA dans de nombreux contextes futurs.

Pour approfondir l'engagement des élèves avec les concepts abordés dans cette étape, envisagez d'explorer ces questions directrices à travers des discussions ou des écrits de réflexion :

1. **Évolution de la conception des prompts** : En quoi votre prompt initial différait-il de votre prompt final optimisé ? Quels changements spécifiques avez-vous apportés, et comment chaque changement a-t-il affecté la réponse de l'IA ? Si vous conceviez un prompt pour obtenir des informations sur les attractions locales près de la ville identifiée, comment le structureriez-vous pour vous assurer de recevoir des recommandations utiles et spécifiques ?
2. **Avantages de l'intégration d'IA** : De quelles manières l'ajout de capacités d'IA transforme-t-il ce que votre Bot Buddy peut faire par rapport aux approches de programmation traditionnelles seules ? Quels types de questions votre assistant peut-il maintenant répondre qui auraient été difficiles ou impossibles à programmer manuellement ? Comment cette combinaison de données GPS et de réponse d'IA pourrait-elle changer la façon dont les utilisateurs interagissent avec les services géolocalisés ?
3. **Considérations éthiques** : Quelles préoccupations de confidentialité pourraient surgir lorsqu'une application collecte des données de localisation et les traite via des systèmes d'IA ? Comment les services de localisation alimentés par l'IA pourraient-ils affecter différemment diverses communautés ou quartiers ? Quelles responsabilités ont les développeurs lors de la création de systèmes qui fournissent des recommandations sur les lieux à visiter ou à éviter dans une ville ?
4. **Limitations techniques et solutions** : Quels défis avez-vous observés lors du test de votre conversion localisation-vers-ville ? Comment votre agent conversationnel pourrait-il gérer les situations où les signaux GPS sont faibles, les coordonnées imprécises, ou l'IA fournit des informations de localisation incorrectes ? Quels systèmes de secours ou méthodes de vérification pourraient rendre votre assistant plus fiable dans des conditions réelles ?

Étape 4 : Diffusion d'informations accessibles aux utilisateurs finaux



Contexte et description du problème à résoudre à cette étape : Dans les étapes précédentes, les élèves ont construit avec succès une solution qui peut automatiquement déterminer la localisation de l'utilisateur grâce aux requêtes vocales, générant des réponses pertinentes en utilisant l'IA. Ils peuvent réutiliser ce code pour créer un véritable agent conversationnel qui peut communiquer ses réponses par la voix plutôt que seulement par le texte. Dans le contexte actuel, un nombre croissant de personnes utilisent des assistants vocaux pour interagir avec des appareils et obtenir des informations. Cela inclut les applications d'assistance à la conduite, les systèmes de réponse téléphonique, les assistants personnels tels que Siri ou Google Assistant, et bien d'autres. L'un des défis majeurs de ces systèmes est de fournir une expérience utilisateur fluide et naturelle grâce à une synthèse vocale de haute qualité. Cette étape finale se concentrera sur l'intégration des capacités de synthèse vocale, complétant la boucle d'interaction complète du Bot Buddy Assistant Urbain.

Objectifs d'apprentissage : Comprendre les principes fondamentaux de la synthèse vocale, y compris les techniques et modèles utilisés pour générer la parole artificielle. Apprendre comment implémenter la fonctionnalité de synthèse vocale dans CodeSkool. Explorer les défis et enjeux liés à la qualité de la synthèse vocale, tels que le naturel, l'intelligibilité, l'adaptabilité et la personnalisation. Développer des compétences pour configurer les paramètres vocaux afin d'optimiser l'expérience utilisateur. Créer un agent conversationnel complet qui collecte, traite et diffuse les informations à travers la parole naturelle.

Conceptualisation

Dans cette quatrième et dernière étape, les élèves complètent leur agent conversationnel en ajoutant des capacités de sortie vocale et des prompts supplémentaires pour offrir un véritable service aux utilisateurs. Cela transforme leur assistant d'un système basé sur le texte en une application entièrement interactive vocalement qui à la fois écoute les utilisateurs et énonce des réponses, créant une expérience plus naturelle et accessible.

Question de recherche 1 : Implémentation de la synthèse vocale

Comment peut-on convertir efficacement les réponses d'IA basées sur le texte en sortie vocale qui sonne naturelle et est facilement comprise par les utilisateurs ?

Hypothèse : En implémentant la fonctionnalité de synthèse vocale avec une configuration appropriée des paramètres vocaux, les élèves peuvent créer un système vocal qui communique efficacement les informations générées par l'IA avec naturel.

Concepts clés :

- **Synthèse vocale** : Technologie qui convertit le texte écrit en sortie vocale parlée en utilisant des algorithmes qui génèrent la parole artificielle.
- **Paramètres vocaux** : Aspects configurables de la parole synthétisée incluant la sélection de voix, le débit de parole, la hauteur et le volume.
- **Naturel de la parole** : Facteurs qui influencent à quel point la parole synthétique sonne humaine pour les auditeurs.

Question de recherche 2 : Créer une expérience vocale interactive cohérente

Comment peut-on intégrer l'entrée vocale, le traitement d'IA et la sortie vocale pour créer un agent conversationnel cohérent et fiable qui délivre des informations géolocalisées précieuses ?

Hypothèse : En concevant des flux de conversation structurés et des prompts sensibles au contexte, les élèves peuvent créer un système vocal interactif cohérent qui délivre de manière fiable des informations pertinentes sur les environnements urbains à travers une conversation naturelle.

Concepts clés :

- **Conception de conversation** : Principes pour créer des flux de dialogue logiques et intuitifs qui guident les utilisateurs à travers les interactions.
- **Préservation du contexte** : Méthodes pour maintenir la conscience de localisation et l'historique de conversation tout au long des interactions.
- **Diffusion d'information** : Techniques pour présenter des informations complexes par la voix de manière accessible.

En abordant ces questions, les élèves créeront un agent conversationnel entièrement accessible qui complète le cycle entrée-traitement-sortie à travers l'interaction vocale naturelle. Ce composant final transforme leur projet d'une démonstration technique en une application pratique, centrée sur l'utilisateur avec une utilité réelle.

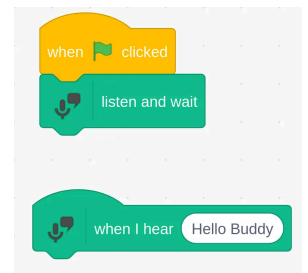
Investigation par les élèves

Dans cette phase d'investigation, les élèves vont implémenter les capacités de sortie vocale et créer un flux conversationnel pour leur Bot Buddy Assistant Urbain, en faisant une expérience véritablement interactive basée sur la voix.

Créer un agent conversationnel activé par la voix

Les élèves vont mettre en place la structure de base de leur assistant activé par la voix en implémentant une phrase déclencheur et un message d'accueil.

Pour implémenter cela dans CodeSkool, les élèves doivent s'assurer que les extensions « **Text to speech** » et « **Synthèse vocale** » sont ajoutées au projet. Dans les blocs « **Synthèse vocale** », ils peuvent sélectionner le **bloc déclencheur** « **Quand j'entends [valeur]** » pour activer l'agent conversationnel. Ainsi, chaque fois que le drapeau vert sera cliqué, grâce au bloc « **Écouter et attendre** » ajouté au début du protocole, l'agent conversationnel réagira automatiquement s'il entend la phrase déclencheur (dans l'exemple « Bonjour Buddy »).



Une fois cela fait, les élèves doivent configurer les **paramètres vocaux** pour l'agent conversationnel. Optimiser les paramètres vocaux est essentiel pour une **communication naturelle et efficace**. Les bons réglages pour la **vitesse et le type de voix** impactent significativement la façon dont les utilisateurs comprennent et interagissent avec l'agent conversationnel. Ainsi, deux blocs doivent être utilisés pour définir ces paramètres disponibles directement dans la section « Text to Speech », dans la catégorie réglages :

- « **set playback speed to [valeur]** » : Ce bloc contrôle à quelle vitesse la voix synthétique parle. La valeur va typiquement de 0 (très lent) à 2 (très rapide), avec 1 étant la vitesse normale
- « **choisir la voix [valeur]** » : Ce bloc vous permet de sélectionner différentes options de voix pour la sortie de synthèse vocale. Différentes voix peuvent varier en genre, âge ou caractéristiques d'accent

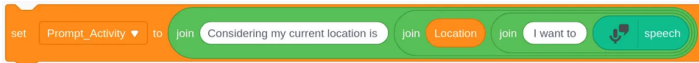
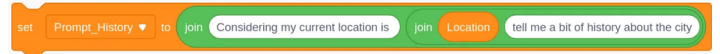


Concevoir des flux de conversation avec des prompts

Après avoir configuré la voix, les élèves créent des flux de conversation structurés pour guider les interactions des utilisateurs avec leur assistant. Nous fournissons des conseils pour créer une structure de blocs logique, bien que ceci ne serve que d'exemple. Les élèves sont encouragés à expérimenter avec différentes approches d'organisation, apprenant de la façon dont l'assistant répond pour construire un meilleur code et de meilleurs prompts.

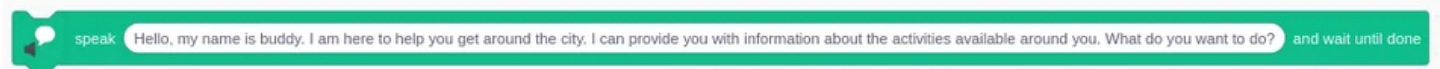
Exemple - Bonjour Buddy ! Créer des variables pour stocker différents prompts d'IA pour des sujets spécifiques, comme fait précédemment pour la localisation. Notre agent doit :

- **Proposer des activités à proximité** : nous avons créé la variable « **Prompt_Activity** » en utilisant 3 blocs de jointure comme : « **Considérant que ma localisation actuelle est [variable Location], je veux [speech]** »
- **Fournir des informations historiques sur la ville actuelle** : nous avons créé la variable « **Prompt_History** » en utilisant 2 blocs de jointure comme : « **Considérant que ma localisation actuelle est [variable Location], raconte-moi un peu l'histoire de la ville** »

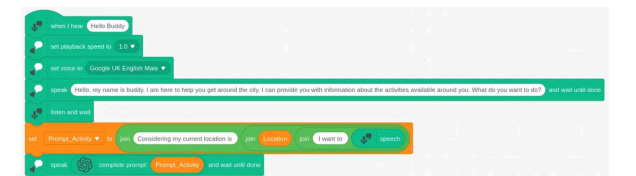
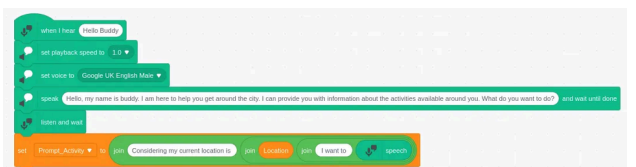


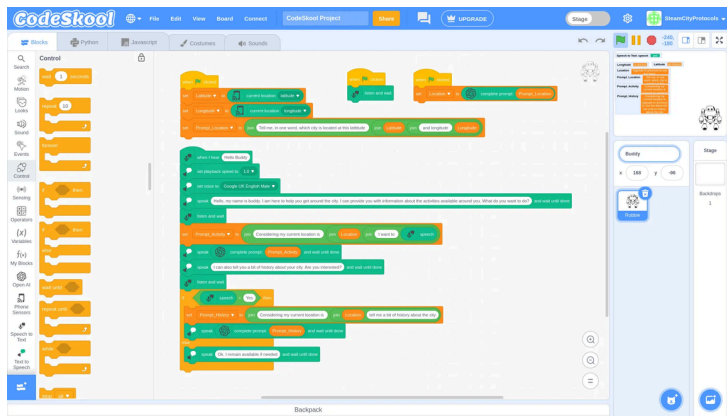
Implémenter un arbre de dialogue simple qui répond aux **entrées vocales** :

- Une fois que l'utilisateur dit la phrase déclencheur « Bonjour Buddy », Buddy répond avec une première proposition sur les activités à proximité en utilisant le message suivant (avec un bloc synthèse vocale) : « **Bonjour, je m'appelle Buddy. Je suis là pour vous aider à vous déplacer dans la ville. Je peux vous fournir des informations sur les activités disponibles autour de vous. Que voulez-vous faire ?** »



- Après que Buddy ait fini de parler, nous utilisons un bloc « écouter et attendre » qui permet à l'utilisateur de spécifier l'activité désirée à travers le bloc « **définir Prompt_Activity** ».
- Basé sur la réponse de l'utilisateur, comme « **manger une pizza** », Buddy utilise une **combinaison de blocs** « parler » et **Open AI** « **compléter prompt** » pour fournir une réponse détaillée à la demande.
- Après la fin de cette discussion, Buddy passe à proposer des informations sur l'histoire de la ville. L'interaction continue avec Buddy demandant « **Je peux aussi vous raconter un peu l'histoire de votre ville. Êtes-vous intéressé ?** ».
- Contrairement à l'interaction d'activité qui utilisait une **question ouverte**, ici nous implémentons une **question fermée** nécessitant un **bloc de contrôle** « **si [] alors** ». L'utilisateur ne peut répondre que par oui ou non, déclenchant différentes réponses du bot.
 - Si l'utilisateur dit **oui**, Buddy présente l'histoire de la ville.
 - Si l'utilisateur dit **non**, Buddy termine poliment la conversation : « **D'accord. Je reste disponible si besoin** »





Voici ci-contre notre exemple complet qui s'appuie sur toutes les activités de programmation réalisées dans les étapes précédentes disponible à ce lien : <https://share.codeskool.cc/cqZu>

Cette investigation démontre que créer un agent conversationnel de base est relativement simple avec les bons outils et la bonne approche. Cependant, le défi clé réside dans la **conception d'un flux de conversation logique et engageant** qui apporte de la valeur aux utilisateurs. Bien que notre exemple utilise une **structure simple et linéaire avec des réponses prédéterminées**, cela ne représente qu'une seule approche d'implémentation possible.

Les élèves peuvent exercer leur créativité en développant des **modèles d'interaction plus sophistiqués** qui :

- **Gèrent plusieurs variations d'entrée utilisateur grâce à une logique conditionnelle étendue**
- **Implémentent des chemins de conversation ramifiés basés sur les préférences des utilisateurs**
- **Créent des flux de dialogue plus naturels avec des questions de suivi**
- **Incluent la gestion d'erreurs pour des réponses utilisateur inattendues**

En expérimentant avec différentes structures de conversation et une logique si-alors étendue, les élèves peuvent créer des agents de plus en plus sophistiqués et réactifs tout en apprenant des leçons précieuses sur la conception d'interaction utilisateur et la gestion des flux de conversation.

Restitution et réflexion

- **Connaissances mobilisées** : Dans la dernière étape du Bot Buddy Adventure, les élèves ont créé un agent conversationnel interactif vocalement, intégrant la reconnaissance vocale et la synthèse vocale. Ils ont appris à ajuster le débit de parole et la voix pour améliorer l'expérience utilisateur. En mettant en œuvre des phrases déclencheurs et des flux conversationnels, ils ont développé des compétences en conception conversationnelle, en offrant des choix et des informations contextuelles pour rendre l'assistant plus naturel et utile. L'intégration de la synthèse vocale a permis d'explorer divers domaines de connaissance.
 - **Technologie de synthèse vocale** : Comprendre les principes fondamentaux de conversion du texte écrit en langage parlé et comment différents paramètres affectent la qualité de sortie.
 - **Traitement du langage naturel** : Explorer comment les ordinateurs peuvent comprendre et répondre au langage humain dans des contextes conversationnels.
 - **Conception d'accessibilité** : Apprendre comment les interfaces vocales rendent la technologie plus accessible à des utilisateurs diversifiés et comment les choix de conception peuvent améliorer ou entraver l'accessibilité.
- **Réflexion sur la mise en œuvre en classe** : Cette étape fonctionne mieux lorsque les élèves ont de multiples occasions de tester et affiner leurs implémentations. La qualité de l'interaction vocale s'améliore souvent considérablement avec plusieurs itérations de test et d'ajustement. Les tests entre pairs sont particulièrement précieux, car différentes voix, accents et styles d'élocution peuvent révéler des forces et faiblesses dans la reconnaissance vocale qui pourraient ne pas être apparentes lors de tests par un seul utilisateur. De même, avoir différents auditeurs évaluer la sortie vocale peut fournir des perspectives diverses sur la qualité de la voix et la compréhensibilité. Pour l'évaluation, envisagez de faire démontrer leurs agents conversationnels à la classe, montrant à la fois l'implémentation technique et l'expérience utilisateur. Les discussions sur les choix de conception et le raisonnement derrière des implémentations spécifiques aident les élèves à articuler leur apprentissage et partager leurs découvertes avec leurs pairs.
- **Résultats d'apprentissage** : À travers cette investigation finale, les élèves ont développé plusieurs compétences importantes. Ils comprennent comment implémenter des déclencheurs activés par la voix et des flux conversationnels qui guident les utilisateurs à travers les interactions. Ils ont acquis de l'expérience en configurant les paramètres vocaux pour optimiser l'expérience d'écoute et en concevant des structures de conversation qui semblent naturelles et utiles. Les élèves ont également développé une appréciation de la nature itérative de la conception d'interfaces vocales, expérimentant comment les tests et le raffinement sont essentiels pour créer des expériences vocales efficaces. Ils comprennent que de petits détails dans l'implémentation — de la vitesse de la voix à la formulation conversationnelle — peuvent impacter significativement la façon dont les utilisateurs perçoivent et interagissent avec les systèmes basés sur la voix. Ces compétences et perspectives préparent les élèves à réfléchir de manière critique sur les technologies vocales qu'ils rencontrent dans la vie quotidienne et à concevoir des technologies plus accessibles et centrées sur l'utilisateur à l'avenir.



Pour approfondir l'engagement des élèves avec les concepts abordés dans cette étape, envisagez d'explorer ces questions à travers des discussions ou des écrits de réflexion :

1. Comment une interface basée sur la voix change-t-elle la façon dont les utilisateurs interagissent avec la technologie par rapport aux interfaces traditionnelles basées sur l'écran ?
2. Quels défis peuvent surgir lors de la conception de flux de conversation, et comment peut-on les surmonter ?
3. Comment les assistants vocaux pourraient-ils être davantage personnalisés pour répondre aux besoins de différents utilisateurs ?
4. Quelles considérations éthiques émergent lors de la création de systèmes qui utilisent des voix au son de plus en plus naturel ?



Teste une version modifiée du Bot Buddy Assistant Urbain

Voici une autre version du programme basée sur les mêmes blocs, ajoutant des fonctions et capacités pour raconter des histoires en utilisant des accessoires et des variables multiples. Pour l'ouvrir, allez sur ide.codeskool.cc et sélectionnez « ouvrir un projet depuis un lien ».

Vous pouvez utiliser le **lien de partage** : <https://share.codeskool.cc/eT4U>

Voici une vidéo de démonstration (en français) : https://drive.google.com/file/d/19cR5d5_sKCEtvqY_fr5_TJbMsap-Knd/view?usp=drivesdk



Inspirations du monde réel

Les assistants d'IA vocaux et les outils d'accessibilité contribuent à transformer la façon dont les gens interagissent avec les services numériques. Bien qu'encore en évolution, ces technologies montrent des perspectives prometteuses pour accompagner les personnes en situation de handicap et répondre à divers défis de société. Voici quelques exemples de la façon dont les technologies vocales sont utilisées pour rendre les solutions plus inclusives et accessibles.



1. Be My Eyes - Assistance visuelle renforcée par l'IA

Be My Eyes a commencé comme une application mettant en relation des utilisateurs aveugles et malvoyants avec des bénévoles voyants par appels vidéo pour une assistance visuelle. En 2023, ils ont intégré « Be My AI », un assistant d'IA alimenté par GPT-4 qui peut décrire des images et fournir une assistance visuelle immédiate sans attendre un bénévole humain. Les utilisateurs peuvent simplement prendre une photo et l'IA décrit ce qui se trouve dans l'image, lit le texte, identifie les produits ou fournit des conseils de navigation.

Cette initiative démontre comment l'IA vocale peut considérablement améliorer l'autonomie des personnes malvoyantes en fournissant des informations immédiates et précises sur leur environnement. La combinaison d'interaction vocale, de reconnaissance d'images et de génération de langage naturel crée une interface accessible qui fonctionne avec, plutôt qu'à la place de, l'assistance humaine.

<https://www.bemyeyes.com>



2. NaviLens - Système de navigation par codes couleur

NaviLens est un système innovant développé en Espagne qui utilise des codes colorés semblables à des QR codes combinés à une application smartphone pour aider les personnes malvoyantes à naviguer de manière autonome dans les environnements urbains. Contrairement aux QR codes traditionnels, les marqueurs NaviLens peuvent être détectés à des distances allant jusqu'à 12 mètres et sous des angles allant jusqu'à 160 degrés, permettant aux utilisateurs de scanner leur environnement en balayant simplement leur smartphone.

Ce qui rend NaviLens particulièrement remarquable, c'est son interface vocale qui lit à voix haute les informations sur les marqueurs détectés, fournissant le contexte de localisation, des conseils directionnels et des informations détaillées sur l'environnement. Le système a été implémenté dans les réseaux de transport public à Barcelone, Madrid et d'autres villes à travers l'Europe, aidant les voyageurs malvoyants à naviguer de manière autonome dans des systèmes de transport complexes.

<https://www.navilens.com>