



## Създайте чатбот в CodeSkool

### Необходими материали и инструменти

Софтуер за блоково програмиране CodeSkool (<https://ide.codeskool.cc/>). CodeSkool е особено подходящ за този проект, защото комбинира интеграция с отворен изкуствен интелект с разширения на телефонни сензори и възможности за преобразуване на текст в реч.



Има няколко алтернативи на CodeSkool. Scratch предлага разширения за геолокация, които осигуряват подобна функционалност в блокова среда. MIT App Inventor включва вградени компоненти за местоположение, специално проектирани за разработване на мобилни приложения. За класове с по-напреднал опит в програмирането, Python библиотеки като geocoder или geopy предлагат усъвършенствани услуги за местоположение, които могат да бъдат интегрирани в по-големи приложения.

### Кодиране и разбиране на вашата програма

1

#### Влезте в CodeSkool и съхранете GPS информация, използвайки променливи

За да внедрите GPS функционалността, започнете със създаването на нов проект и достъп до редактора в CodeSkool.

Преди да получите достъп до каквото и да е GPS данни, създайте местоположения в паметта на програмата, където ще се съхранява тази информация. Започнете, като отидете в секцията „Променливи“ на палитрата „Блокове“ и щракнете върху „Създаване на променлива“. Създайте променлива с име „Географска ширина“, за да съхранявате позицията на потребителя север-юг. Тази променлива ще действа като специален контейнер, който запомня тази специфична информация в цялата програма. След това се върнете в секцията „Променливи“ и създайте втора променлива с име „Географска дължина“, за да съхранявате позицията изток-запад. Заедно тези две променливи ще осигурят пълна географска позиция, която се запазва в паметта.

След като контейнерите за памет за данните за местоположението са създадени, достъпете GPS възможностите на устройството, за да попълните тези променливи с реални координати. От основния интерфейс достъпете бутона „Разширения“, разположен в самата добра част на лентата с инструменти. В библиотеката с разширения потърсете „Сензори за телефони“ и добавете това разширение към проекта, за да получите достъп до GPS функционалността.

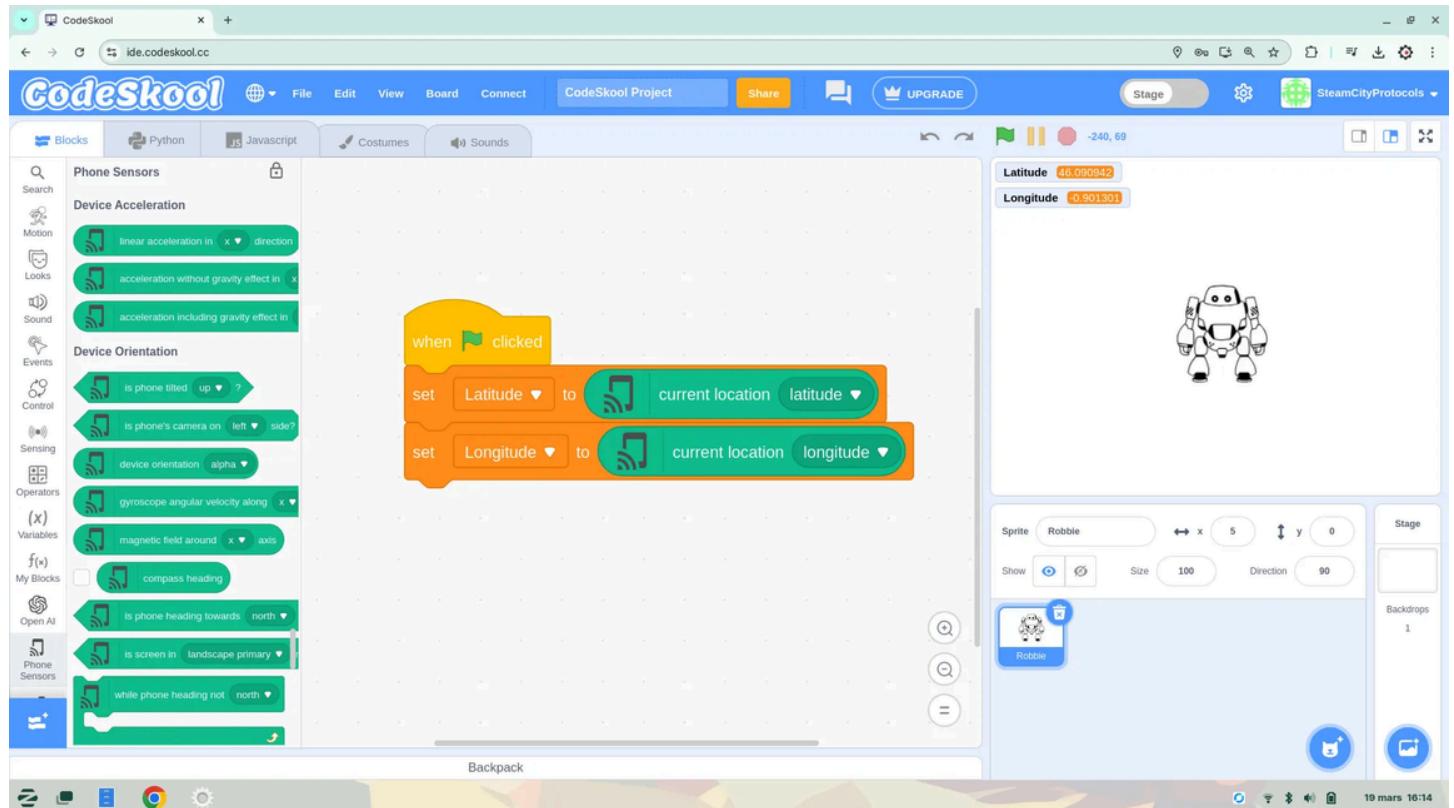


След добавяне на разширението, намерете блокове, свързани с GPS, в новата категория, която се появява в палитрата с блокове. Потърсете подкатегорията на блоковете, наречена „GPS местоположение“, в която да намерите блока „текущо местоположение“ за географска дължина и ширина.

Отсега нататък всички блокове и променливи, необходими за създаването на програмата, се идентифицират чрез прости стъпки:

1. За да задействате процеса на събиране на данни, добавете блок за събитие „При щракване върху бутон“ към работното пространство. Това ще служи като отправна точка за заснемане на местоположението на потребителя.

- От секцията „Променливи“ плъзнете два блока „задай [променлива] на [стойност]“ и ги свържете със събитието на бутона. Тези блокове са механизъмът, чрез който информацията преминава от сензорите към паметта на програмата.
- Конфигурирайте първия блок, за да дефинира „Географска ширина“, а втория, за да дефинира „Географска дължина“. Изборът на променлива показва, че програмата вече има специфични места в паметта, готови да съхраняват различни видове информация.
- Свържете съответните блокове за „текуща позиция“ от разширението Phone Sensors към всяка стойност на променлива. Изберете „географска ширина“ за първата променлива и „географска дължина“ за втората. Когато тези блокове се изпълнят, те ще заснемат показанията на сензорите и ще ги съхраняват в паметта на програмата.
- Щракнете върху зеления флаг в тестовия панел, за да проверите дали променливите успешно съхраняват информация.



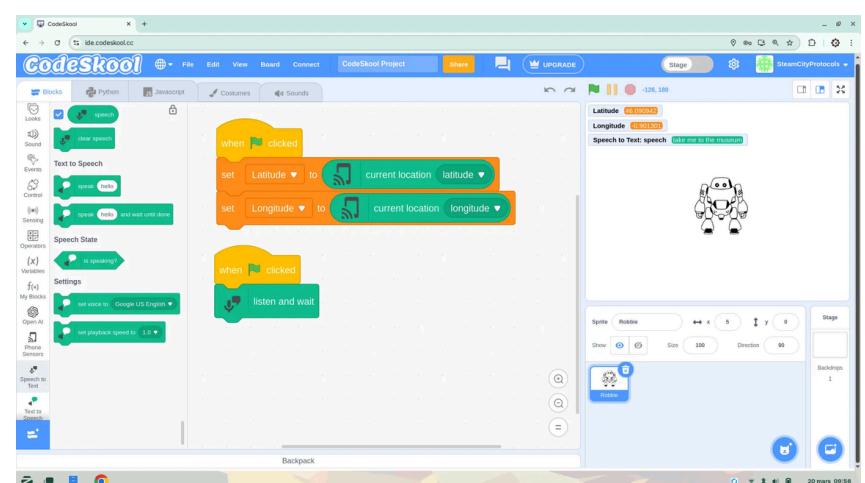
Тази програма ще работи или на компютъра, използвайки местоположението на машината, или чрез мобилно устройство, използвайки разширението Phone Sensors.

## 2

## Внедряване и тестване на разпознаване на реч

Въз основа на създадената програма, имплементирайте втората функционалност за разпознаване на глас.

В същия редактор се върнете към бутона „Разширения“ в края на лентата с инструменти и потърсете „Реч в текст“ в библиотеката с разширения, за да го добавите към проекта.



След добавяне на разширението, намерете блокове, свързани с разпознаване на реч, в новата категория, която се появява в палитрата с блокове. Потърсете блока „Слушай и чакай“.

За по-голяма яснота в блоковете, добавете втори блок „При щракване върху бутон“, за да идентифицирате двете основни функционалности на системата.

След като основната функционалност за разпознаване на реч е внедрена, използвайте повторно същия метод за тестване, както в стъпка 1, за да оцените систематично неговата точност и да разберете неговите възможности и ограничения. Използвайте повторно тестовите изречения и попълнете таблицата с възможностите на системата:

Произнесена присъда	Разпознаване на системата	Прецизност
Изречението, формулирано от учениците	Какво системата транскрибира от вашата реч	Оценка на разбирането (Добро / Задоволително / Слабо)
„Къде е най-близкият парк?“		
„Заведи ме в музея“		
„Има ли кафене наблизо?“		
„Колко е часът за кино?“		

### 3

## Преобразуване на местоположение в град

Внедрете първата интеграция с изкуствен интелект, като трансформирате сировите GPS координати в смислено име на град. Тази функция служи на важна цел: тя преобразува техническите данни (цифрови координати) в информация, четлива за човек, която е по-интуитивна и полезна както за системата, така и за нейните потребители. Това преобразуване ще послужи и като основа за по-сложни подки в последващи дейности, тъй като включването на името на града в бъдещи подки ще помогне на изкуствения интелект да предоставя по-контекстуално релевантна информация.

### 1. Подгответе променливите

За да се реализира това преобразуване от координати в град в CodeSkool, подгответе програмата, използвайки различни променливи, които след това могат да се активират при необходимост. Първо, създайте нова променлива, наречена „Location“, която ще съхранява името на града, съответстващо на GPS координатите, и втора, наречена „Prompt\_Location“, която ще съдържа структурата на въпроса, който да се зададе на AI модела.

### 2. Създайте структурата на подканата

Променливата „Prompt\_Location“ ще трябвало да ви позволи да трансформирате променливите за географска ширина и дължина, вече събрани от програмата, в имена на градове. За да структурирате ясно тази подкана:

- В същия блок за събитие „При щракване върху бутон“, където се извличат GPS координатите, добавете блок „set [LocationPrompt] to [value]“, който ще се използва за дефиниране на структурата на подканата.
- След това дефинирайте структурата на подканата в полето [стойност]. Започнете, като формулирате заявката естествено, като предлагате подканите на глас. Това може да бъде например „Кажете ми кой град се намира на тази географска ширина и дължина“. След като сте формулирали няколко идеи, тествайте ги в програмата.
- За да комбинирате текст и променливи в цялостен подканващ елемент, използвайте блоковете „join [стойност] [стойност]“ в секцията „Оператори“. Тъй като две променливи (географска ширина и дължина) трябва да бъдат интегрирани с текст, използвайте вложени блокове за свързване:

“

[Кажи ми кой град е на тази географска ширина] + [променлива „Географска ширина“] + [и географска дължина] + [променлива „Географска дължина“]

”

Това изискава три блока "група [стойност] [стойност]", вложени заедно. Тази структура на влагане е необходима, защото блоковете за свързване могат да комбинират само два елемента едновременно. Чрез влагането им се изгражда цяло изречение, което включва множество променливи и текстови сегменти:

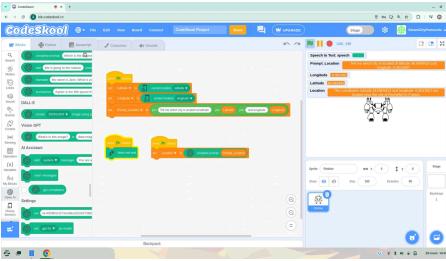
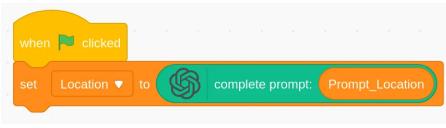
- **Първи блок за свързване: Стойност 1: Текстът „Кажи ми кой град е на географска ширина“ (обърнете внимание на интервала в края) / Стойност 2: Вторият блок за свързване**
- **Втори блок за свързване: Стойност 1: Променливата „Latitude“ / Стойност 2: Третият блок за свързване**
- **Трети блок за свързване: Стойност 1: Текстът "и географска дължина" (обърнете внимание на интервалите преди и след) / Стойност 2: Променливата "Географска дължина"**



### 3. Внедрете интеграция с изкуствен интелект

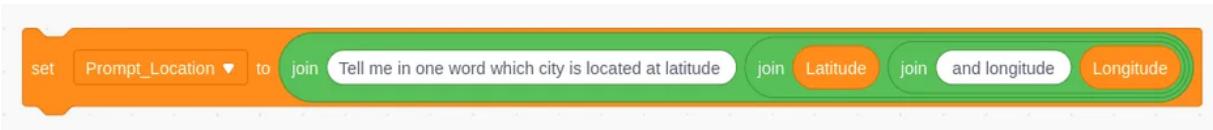
Тествайте подканата. Първо, създайте нов блок „При щракване върху бутон“ за по-голяма яснота и организация в програмата. Това разделя функционалността за събиране на данни от обработката с изкуствен интелект. В този блок:

- Добавете блок „задаване на [променлива] на [стойност]“
- Задайте променливата на „Местоположение“
- Върнете се в раздела „Разширения“ и добавете разширението „Open AI“ към проекта си
- Намерете блока „пълен подкаст: [подканата]“ в разширението Open AI
- Задайте този блок като стойност на „задай местоположението на [стойност]“
- Заменете стойността по подразбиране „prompt“ с тяхната променлива „Prompt\_Location“

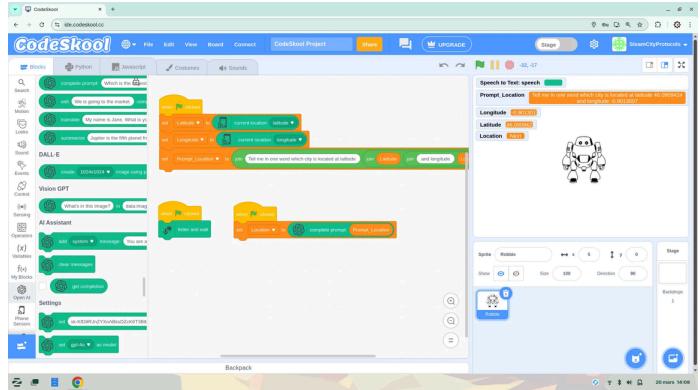


Щракнете върху зеления флаг, за да тествате програмата. Обърнете внимание, че изкуственият интелект обикновено предоставя подробни отговори като „Градът, разположен на 48.8566 географска ширина и 2.3522 географска дължина, е Париж, Франция. Париж е столицата и най-населеният град на Франция.“ Тази подробна информация, макар и информативна, може да не е идеална за чатбота, който се нуждае само от името на града като компонент за последващи подканни.

Сега прецизирайте подканата, за да изиска само конкретната необходима информация. Променете Prompt\_Location, за да бъде по-конкретен: „Кажете ми с една дума кой град се намира на географска ширина [променлива за географска ширина] и географска дължина [променлива за географска дължина].“



Чрез тестване и усъвършенстване на подканата, научете важен принцип на инженерството на подканите: бъдете конкретни както по отношение на исканата информация, така и на желания формат на отговора. Този итеративен процес демонстрира как подканите могат да бъдат оптимизирани, за да генерират точно необходимата информация, като учи, че ефективното взаимодействие с изкуствен интелект често изиска множество усъвършенствания, за да се постигнат желаните резултати.

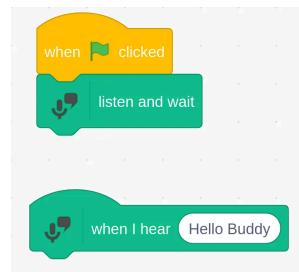


# 4

## Създайте чатбот с гласово активиране

Установете основната структура на гласово активирания асистент, като внедрите задействаща фраза и поздравително съобщение.

За да имплементирате това в CodeSkool, уверете се, че разширенията „Текст към реч“ и „Текст към реч“ са добавени към проекта. В блоковете „Текст към реч“ изберете тригерния блок „Когато чуя [стойност]“, за да активирате чатбота. По този начин, всеки път, когато се щакне върху зеления флаг, благодарение на блока „Слушай и чакай“, добавен в началото на протокола, чатботът автоматично ще реагира, ако чуе тригерната фраза (в примера „Здравей, приятелю“).



След като това е направено, конфигурирайте гласовите настройки за чатбота. Оптимизирането на гласовите настройки е от съществено значение за естествена и ефективна комуникация. Правилните настройки за скорост и тип глас оказват значително влияние върху начина, по който потребителите разбират и взаимодействат с чатбота. Следователно, за дефиниране на тези настройки трябва да се използват два блока, достъпни директно в секцията „Текст в реч“, в категорията настройки:

- „**задаване на скорост на възпроизвеждане на [стойност]**“: Този блок контролира колко бързо говори синтетичният глас. Стойността обикновено варира от 0 (много бавно) до 2 (много бързо), като 1 е нормална скорост
- „**изберете глас [стойност]**“: Този блок ви позволява да изберете различни гласови опции за изхода от текст в реч. Различните гласове могат да се различават по пол, възраст или акцент.

●



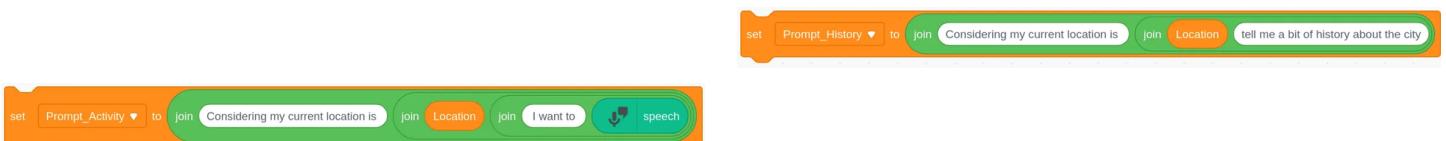
# 5

## Проектиране на потоци на разговор с подкани

След като конфигурирате гласа, създайте структурирани потоци от разговори, които да насочват взаимодействията на потребителите с асистента. Тази логическа блкова структура служи като пример. Експериментирайте с различни организационни подходи, като се учате от начина, по който асистентът реагира, за да изградите по-добър код и подкани.

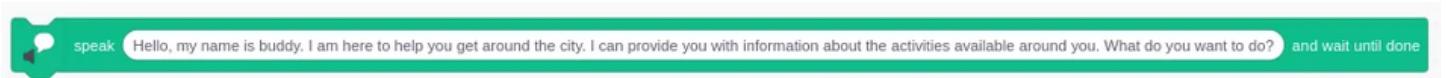
**Пример - Здравей, приятелю! Създай променливи, за да съхраняваш различни AI подкани за конкретни теми, както е направено преди това за локализация. Нашият агент трябва:**

- Предложение за дейности наблизо: Създадохме променливата „Prompt\_Activity“, използвайки 3 блока за свързване, като например: „Като се има предвид, че текущото ми местоположение е [променлива за местоположение], искам да [говоря]“
- Предоставете историческа информация за текущия град: Създадохме променливата „Prompt\_History“, използвайки 2 блока за свързване, като например: „Като се има предвид, че текущото ми местоположение е [променлива за местоположение], разкажете ми малко за историята на града“

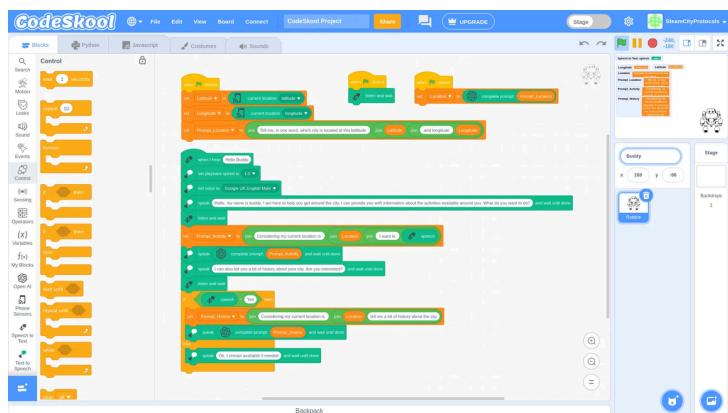


Имплементирайте просто диалогово дърво, което реагира на гласово въвеждане:

- След като потребителят каже задействащата фраза „Здравей, приятелю“, Бъди отговаря с първо предложение за дейности наблизо, използвайки следното съобщение (с блок за преобразуване на текст в реч): „Здравей, казвам се Бъди. Тук съм, за да ти помогна да се ориентираш в града. Мога да ти предоставя информация за дейностите, предлагани около теб. Какво би искал да правиш?“



- След като Бъди приключи с говоренето, използваме блок „слушане и чакане“, който позволява на потребителя да укаже желаната активност чрез блока „set Prompt\_Activity“.
- Въз основа на отговора на потребителя, като например „яж пица“, Buddy използва комбинация от блокове „говор“ и Open AI „пълен подкаст“, за да предостави подробен отговор на заявката.
- След като този разговор приключи, Бъди продължава с разказване на информация за историята на града. Разговорът продължава с въпроса на Бъди: „Мога да ви разкажа и малко за историята на вашия град. Интересувате ли се?“
- За разлика от взаимодействието с активността, което използваше отворен въпрос, тук имплементираме затворен въпрос, изискващ контролен блок „ако [ ] тогава“. Потребителят може да отговори само с „да“ или „не“, което задейства различни отговори от бота.
  - Ако потребителят каже „да“, Бъди представя историята на града.
  - Ако потребителят каже „не“, Бъди учтиво прекратява разговора: „Добре. На разположение съм, ако е необходимо.“



Ето нашия пълен пример, който е базиран на всички програмни дейности, извършени в предишните стъпки, достъпни на тази връзка: <https://share.codeskool.cc/cqZu>

## Пример

Ето друга версия на програмата, базирана на същите блокове, добавяща функции и възможности за разказване на истории, използвани множество props и променливи. За да я отворите, отидете на [ide.codeskool.cc](http://ide.codeskool.cc) и изберете „open project from link“.

Можете да използвате линка за споделяне: <https://share.codeskool.cc/eT4U>

Ето демонстрационно видео (на френски): [https://drive.google.com/file/d/19cR5d5\\_sKCEtvqY\\_\\_fR5\\_TJbMsap-Knd/view?usp=drivesdk](https://drive.google.com/file/d/19cR5d5_sKCEtvqY__fR5_TJbMsap-Knd/view?usp=drivesdk)

