

Inżynieria Oprogramowania

Projekt “Aurora – smart home assistant”

Opis:

Sieć zdecentralizowanych węzłów odpowiadających za zarządzaniem domem. Głosowy asystent rozpoznaje proste komendy i wydaje na ich podstawie polecenia odpowiednim węzłom, które następnie wykonują zlecone zadanie.

Funkcjonalność:

- ◆ Interfejs do zarządzania węzłem
 - Prosty CLI z komendami connect, send command itp.
 - GUI o podobnej funkcjonalności
- ◆ Zdecentralizowana sieć połączeń
 - Dodawanie, znajdowanie i łączenie z węzłami
 - Wydawanie komend węzłom z innych węzłów
- ◆ Rozpoznawanie predefiniowanych słów-komend i ich wykonywanie
 - Użytkownik może zdefiniować słowo, do którego zostaje wytrenowany model rozpoznający, by mogło być ono później używane jako komenda.
- ◆ Wysoka konfigurowalność
 - Użytkownik sam konfiguruje zachowania węzłów
 - Config węzła głosowego, np. [“light” — “send bulb1 turn_on”]
 - Config węzła roboczego, np. [“turn_on” — “./turn_on.py”]
- ◆ Bezpieczeństwo dzięki szyfrowaniu i certyfikatom

Użytkownicy:

Osoby, które poszukują open source’owych lub bardziej elastycznych alternatyw dla produktów typu Google Home czy Alexa. Dzięki interfejsowi GUI produkt będzie przystępny dla każdego użytkownika. Bardziej zaawansowani użytkownicy mogą skorzystać z nieco bardziej elastycznego interfejsu CLI.

Technologie:

- ◆ Python, Flask
- ◆ HTML, CSS
- ◆ TLS
- ◆ speechrecognition, CMU Sphinx, Google Cloud STT

Narzędzia:

- ◆ Gitlab
- ◆ Sphinx do dokumentacji
- ◆ Facebook (Messenger) do komunikacji

Zespół:

Jakub Bednarz, Jan Ciołek, Gerard Drózdź, Artur Matyjasek

Harmonogram

Sprint #0 – 6 marca 2020:

- ◆ Wizja i harmonogram, ogólne planowanie projektu

Sprint #1 – 27 marca 2020:

- ◆ Podstawowa łączność między węzłami
 - Łączenie węzłów w sieć
 - Przesyłanie komend do węzłów
- ◆ Podstawowy interfejs CLI
 - Czytanie i wykonywanie podstawowych komend
 - `send <nazwa węzła> <komenda>` — wysyła komendę do węzła o tej nazwie
- ◆ Odbieranie i wykonywanie komend
 - Węzeł odbiera komendę z innego węzła, czyta w configu co powinien zrobić i wykonuje to
- ◆ Testowanie różnych bibliotek i rozwiązań w zakresie rozpoznawania mowy

Sprint #2 – 8 kwietnia 2020:

- ◆ Podstawowe rozpoznawanie mowy
 - Rozpoznawanie kilku predefiniowanych wyrazów i wysyłanie komend na ich podstawie

Sprint #3 – 4 maja 2020:

- ◆ Podstawowe GUI
- ◆ Integracja rozpoznawania mowy z interfejsem sieciowym
- ◆ Udoskonalanie i poprawianie protokołu sieciowego

Sprint #4 – 22 maja 2020:

- ◆ Rozbudowanie GUI
- ◆ Opracowanie i ustalenie protokołu kryptograficznego

Sprint #5 – 5 czerwca 2020:

- ◆ Implementacja protokołu kryptograficznego
- ◆ Przygotowanie przykładowych skryptów różnych akcji wykonywanych przez węzły (np. włączanie muzyki)