

Raport Projektu

Patrycja Flicińska, Zuzanna Gumienik

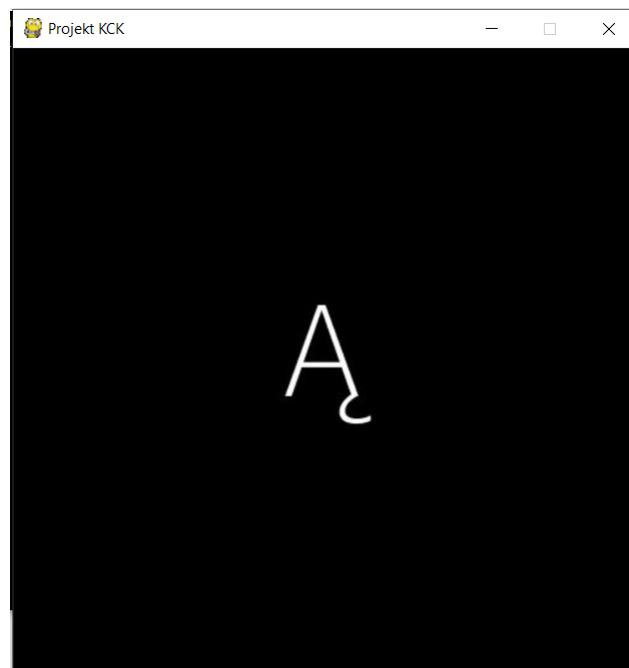
09.01.2022

1. Cel projektu

Celem projektu było stworzenie programu, który wyświetla litery alfabetu, zebranie danych poprzez mruganie, a także napisanie kodu, który pozwoli na odczytanie wymruganych liter, które razem tworzą słowo. Projekt pozwala zaznajomić się z możliwym zastosowaniem interfejsu człowiek-komputer. Jego zastosowanie może pomóc w nawiązywaniu kontaktu z osobami z zespołem zamknięcia, czyli stanem, w którym osoba przy zachowaniu świadomości i przytomności jest niemal w pełni sparalizowana. W zespole tym pacjent traci zdolność kontroli ruchów kończyn, twarzy, języka, gardła i poziomych ruchów oczu. Zachowane zostają pionowe ruchy gałek ocznych i możliwość kontroli powiek.

2. Program wyświetlający litery

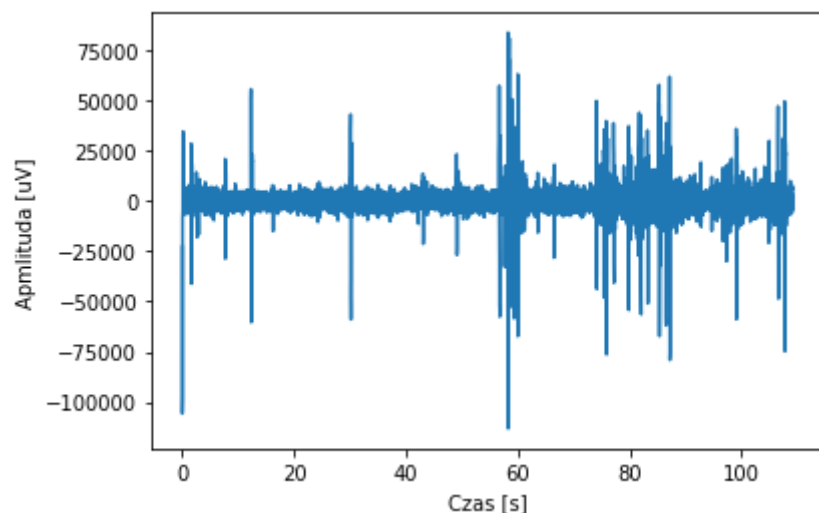
Program wyświetlający litery został stworzony przy pomocy języka programowania Python w bibliotece PyGame. Po uruchomieniu programu wyświetla się okienko z czarnym tłem, a na środku pojawiają się białe litery. Wygląd jest zatem prosty, nie ma żadnych zbędnych dystraktorów. Alfabet zawiera polskie znaki. W pętli wyświetlają się 33 litery (polski alfabet oraz litera "X"), zmieniające się co sekundę. Jedno pełne odtworzenie programu zajmuje zatem 33 sekundy, a po tym czasie pętla zaczyna się wyświetlać od nowa.



Obraz 2.1. Wygląd programu wyświetlającego litery

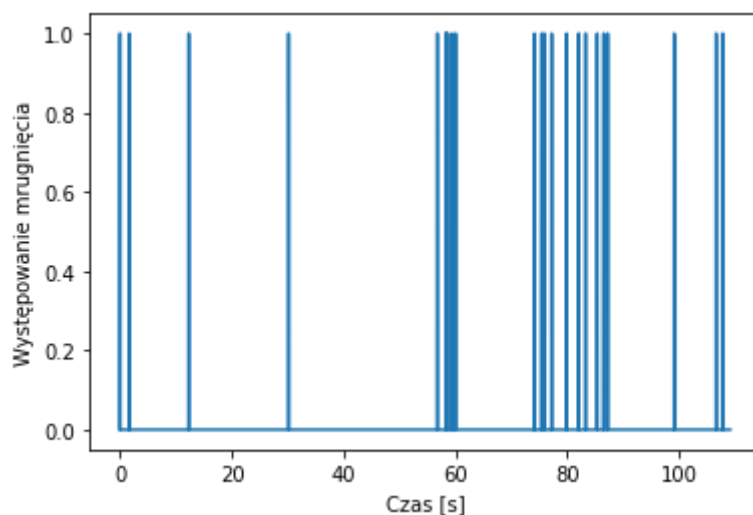
3. Analiza danych i wyniki

Na pierwszym etapie analizy danych zamieniono czas wyrażony HH:MM:SS na sekundy. Pozwoliło to ujednolicić czas zebranych danych, aby na kolejnych etapach odszyfrować w której sekundzie wymrugana została dana litera. Następnie od każdej wartości odjęto wartość początkową plus jeden, aby zamiast czasu rzeczywistego wyświetlała się długość zbierania danych.



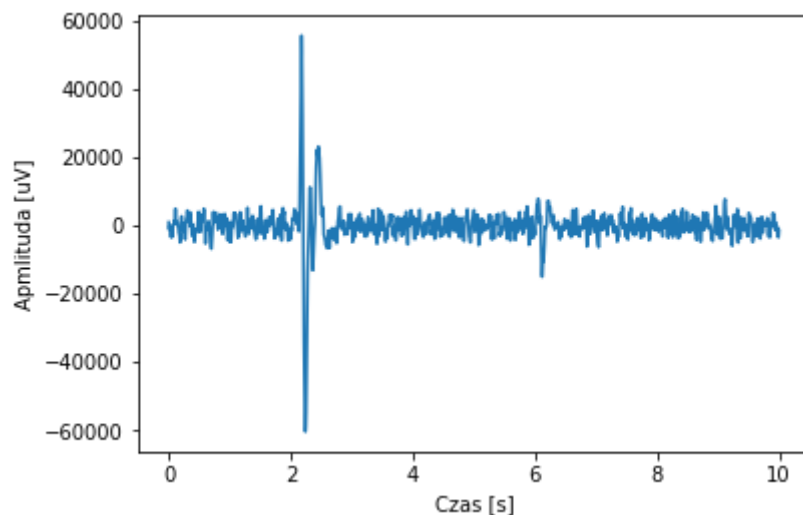
Wykres 3.1. Zebrany sygnał

Następnie stworzono kod, który wychwytyje momenty, w których amplituda jest mniejsza niż -40000 oraz wyświetlono je na wykresie.



Wykres 3.2. Występowanie mrugnięcia w danej sekundzie programu

Jak można zauważyć, wychwyconych momentów jest więcej niż rzeczywista liczba mrugnięć. Słowo, które próbowano przedstawić to “bizon”. Litera “B” pojawia się w drugiej sekundzie, “I” w trzynastej, a powinno być w jedenastej, “Z” w z jednosekundowym opóźnieniem w trzydziestej pierwszej, “O” powinno być w pięćdziesiątej trzeciej, jednak wygląda na to że ta reakcja była opóźniona do pięćdziesiątej siódmej sekundy oraz “N” w osiemdziesiątej trzeciej sekundzie.



Wykres 3.3. Dziesięciosekundowy fragment - litera "B"

Podjęto próbę napisania kodu, który będzie konwertował wyżej przedstawione dane na wymrugane słowo. Stworzono słownik, którego hasłami było 0 lub 1 (informacja, czy amplituda jest mniejsza niż -40000 czy nie) oraz czas. Następnie usunięto wszystkie hasła zawierające "0", i tym sposobem otrzymaliśmy słownik z parami "czas:mrugnięcie".

Następnym krokiem było stworzenie pętli, która pozwoli na przeinterpretowanie wyżej wspomnianych par na odpowiednie litery.

```
#pętla do odczytywania słowa
ostateczne_slowo = []
for key, value in list(slownik.items()):
    if key > 2 and key < 3:
        if "B" not in ostateczne_slowo:
            ostateczne_slowo.append("B")
    elif key > 13 and key < 14:
        if "I" not in ostateczne_slowo:
            ostateczne_slowo.append("I")
    elif key > 31 and key < 32:
        if "Z" not in ostateczne_slowo:
            ostateczne_slowo.append("Z")
    elif key > 57 and key < 58:
        if "O" not in ostateczne_slowo:
            ostateczne_slowo.append("O")
    elif key > 83 and key < 84:
        if "N" not in ostateczne_slowo:
            ostateczne_slowo.append("N")
print(ostateczne_slowo)
```

['B', 'I', 'Z', 'O', 'N']

Obraz 3.1. Propozycja pętli do interpretacji słowa

Wyżej przedstawiona pętla jest jedynie umowna, ponieważ ostateczny czas mrugania nie pokrywa się z szukanim słowem. Na potrzeby zaprezentowania efektu dostosowano niektóre wartości czasowe tak, aby ukazywały inne litery niż teoretycznie powinny.

4. Dyskusja

Z powodzeniem udało się wyświetlić zebrane wyniki w formie wykresu wraz z odpowiadającym im czasem. Wymruganego słowa nie można jednak odczytać ze stuprocentową

pewnością. Istnieje możliwość, że osoba mrugająca działała zbyt wolno, lub elektrody z jakiegoś powodu za późno zapisywały mrugnięcia.

Pojawiają się także błędnie wychwycone momenty spadku amplitudy. Wynikać to może z ustalenia, że jedynie podwójne mrugnięcia liczą się jako sygnalizujące daną literę. Jak można zauważyć, nie udało się do tego jednak zastosować. Być może drugie mrugnięcie od razu po pierwszym nie było wykrywalne przez elektrody. W przypadku powtórzenia projektu, z pewnością należałoby pomyśleć o innym sposobie pozwolenia osobie mrugającej na odpoczynek - na przykład kilku sekundowej przerwie między każdą iteracją alfabetu.

Jako że ostatnia istotna litera została zasygnalizowana w osiemdziesiątej trzeciej sekundzie, dalsza część sygnału może być traktowana jako nieistotna. Są to jedynie zakłócenia powstałe przed zakończeniem pracy programu.

Pętla mająca odczytywać ostateczne słowo powinna rzecz jasna zawierać wszystkie litery wyświetlane podczas badania, a także brać pod uwagę kolejne iteracje alfabetu. W przypadku dalszej pracy nad programem, warto również wziąć pod uwagę słowa, które zawierają tę samą literę kilka razy.