

# **ASK – LABORATORIUM**

## **SPRAWOZDANIE**

### **ZADANIE 3**

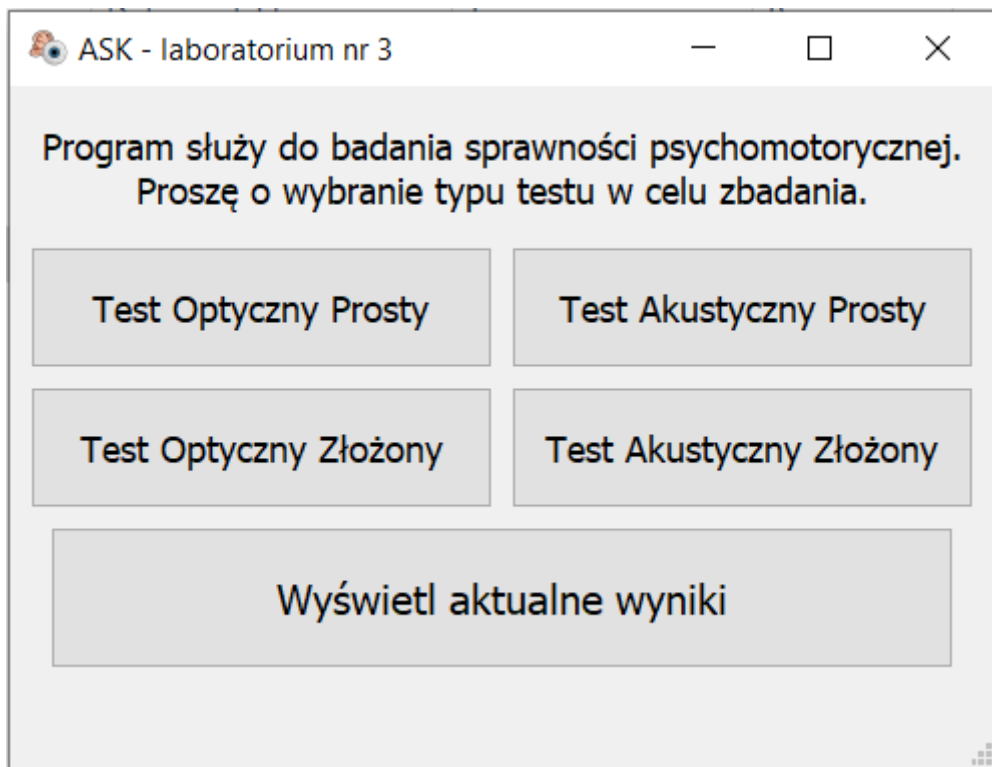
#### **1. Zadanie**

Tematem zadania trzeciego jest napisanie aplikacji uzależnionej od czasu. Jednym z zadań aplikacji ma być możliwie precyzyjny pomiar lub odmierzanie czasu. Wykorzystując dowolny język programowania (wykorzystano język python z biblioteką PyQt5) dla komputerów w standardzie PC napisać aplikację spełniającą funkcję testera sprawności psychomotorycznej np. kandydatów na kierowców. Na aplikację powinna się składać seria różnych testów badających prosty i złożony czas reakcji na bodźce optyczne i akustyczne. Każdy test właściwy powinna poprzedzać informacja o przebiegu testu oraz faza szkoleniowa, w trakcie której osoba badana wykona te same czynności co w trakcie testu, ale bez oceny. Po wykonaniu serii testów osoba poddana badaniom powinien zostać poinformowana o osiągniętych wynikach w formie syntetycznej i analitycznej z wykorzystaniem wartości liczbowych i reprezentacji graficznej.

#### **2. Założenia projektowe**

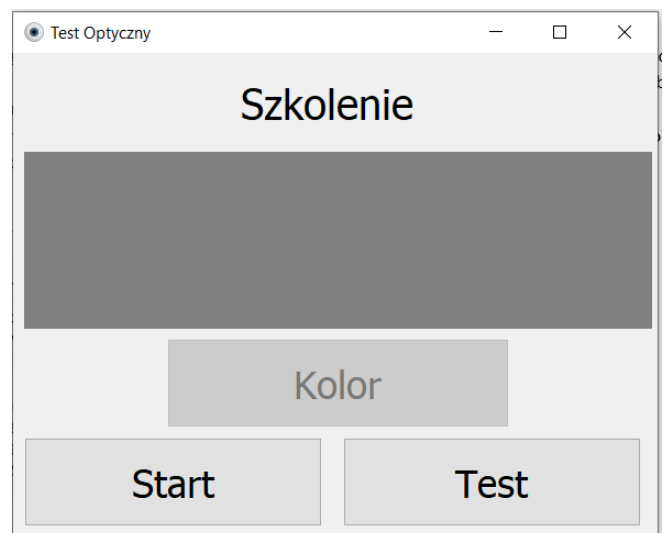
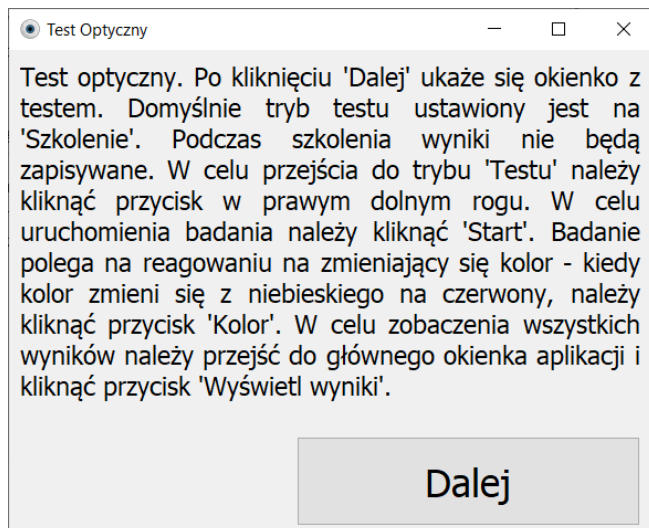
- stworzenie głównego okienka prowadzącego do 'podokienek' z poszczególnymi testami psychomotorycznymi
- stworzenie podokienka zawierającego wykresy pokazujące wyniki z poszczególnych testów ze średnim czasem reakcji użytkownika
- użycie plików tekstowych do zapisu wyników z konkretnych testów
- ponowne kliknięcie na przycisk otwierania okienka resetuje je jeśli jest już otwarte
- każdy test powinien posiadać tryb testowy i szkoleniowy
- podczas trybu szkoleniowego wyniki nie są zapisywane do plików tekstowych
- z testu można wyjść w każdym momencie zamykając okienko

### 3. Działanie programu



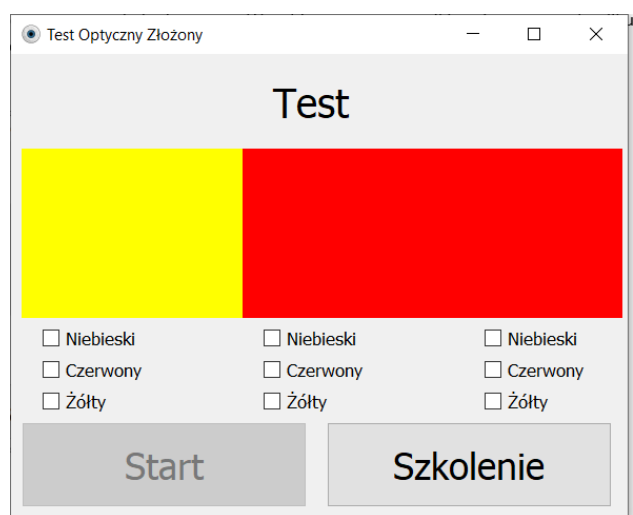
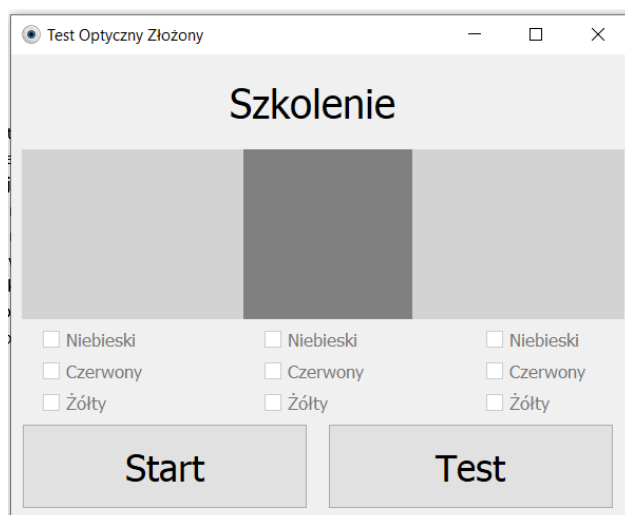
Rys.1 Okno programu

Na starcie programu zostaje inicjalizowane okienko z pięcioma przyciskami oraz tekstem – informacją o przeznaczeniu programu. Cztery górne guziki służą do przejścia (otworzenia okienka) do wybranego testu psychomotorycznego. Dolny przycisk otwiera okienko z wykresami przedstawiającymi osiągnięte wyniki w poszczególnym teście, jeśli brakuje wyników – użytkownik zostanie poproszony o wykonanie wszystkich testów.



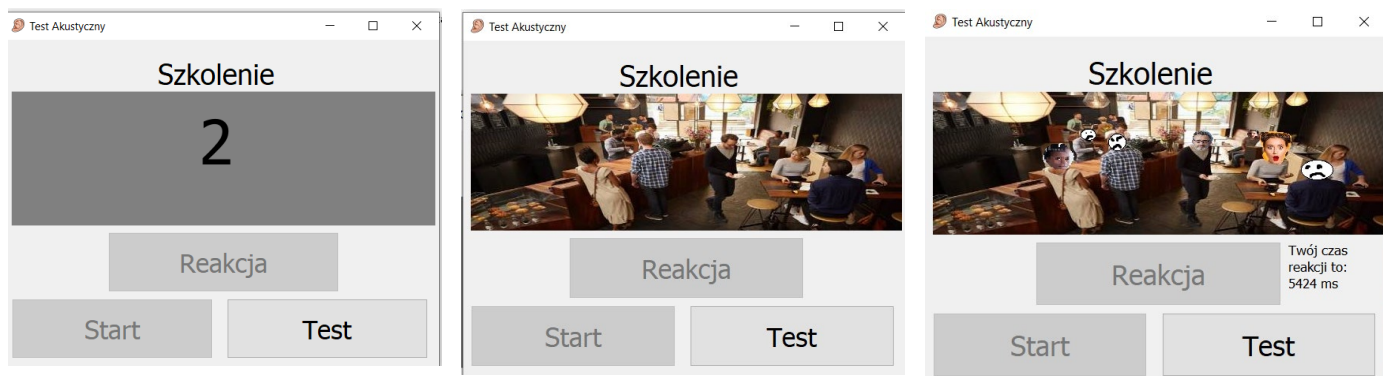
Rys.2 Okno prostego testu optycznego

Pierwszy test (prosty test optyczny) polega na reagowaniu na zmieniający się kolor na środku okienka. Przed testem użytkownik dostaje informacje o jego przebiegu (to dotyczy wszystkich testów), po kliknięciu przycisku 'Dalej' znajduje się już właściwe okno z testem. Po kliknięciu 'Start' test się zaczyna – środek okienka zmienia kolor na niebieski, w tle startuje też timer, który po losowym czasie od 2 do 10 sekund zmienia kolor okienka na czerwony. Wtedy też przycisk 'Kolor' zostaje odblokowany, po kliknięciu przez użytkownika wymienionego przycisku zostaje zmierzony czas reakcji użytkownika i wyświetlony obok przycisku 'Kolor'. Test następnie zaczyna się od nowa. W każdej chwili użytkownik może zmienić tryb testu optycznego ze szkolenia na test. W trybie testowym wyniki zapisywane są do pliku testowego 'optic1.txt'.



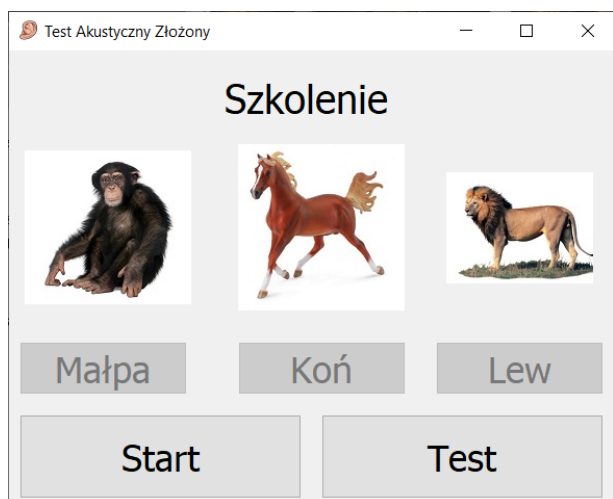
Rys.3 Okno testu optycznego złożonego

Test optyczny złożony – po kliknięciu przycisku 'Start' kolory na środku okienka zmieniają się na losowo wybrane z listy: żółty, czerwony lub niebieski. Zadaniem użytkownika jest zaznaczenie odpowiadających checkboxów pod kolorami. Po odpowiednim zaznaczeniu użytkownik dostaje informację o swoim czasie reakcji i test zaczyna się od nowa.



Rys.4 Okno testu akustycznego prostego

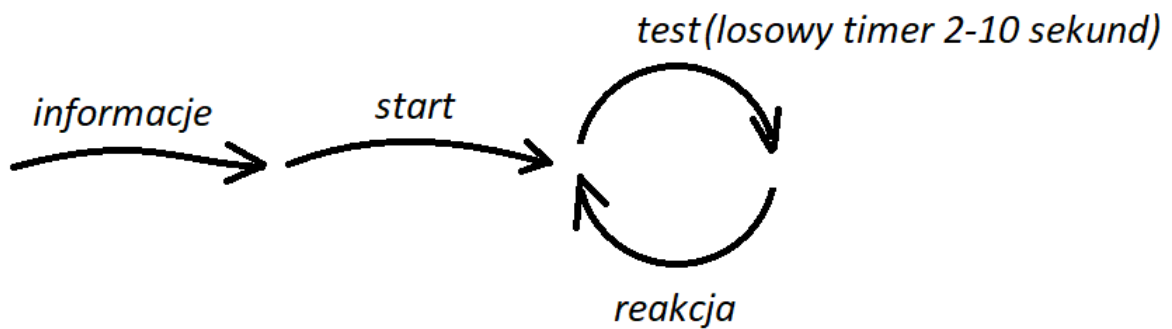
Test akustyczny prosty – celem testu jest zbadanie czasu reakcji użytkownika na dźwięk. Użytkownik wciela się w tłum osób z kawiarni, po pewnym losowym czasie (od 2 do 10 sekund) słysząc dźwięk tłuczonej szklanki, po kliknięciu 'Reakcja' mierzony jest czas reakcji oraz miny osób z kawiarni zmieniają się na zdegrustowane. Następnie odlicza się czas do kolejnej iteracji testu, wyświetlane przez '3,2,1,start!' na ekranie i test zaczyna się od nowa.



Rys.5 Okno testu akustycznego złożonego

Test akustyczny złożony – po kliknięciu przez użytkownika przycisku 'Start', zostają wykonane 4 sygnały głosowe odliczające start testu, po 4 sygnale startuje timer, który po losowym czasie od 2 do 10 sekund puszcza dźwięk jednego z 3 zwierząt podanych na ekranie. Zadaniem użytkownika jest wybranie odpowiedniego zwierzęcia po usłyszeniu jego odgłosu poprzez kliknięcie odpowiedniego przycisku pod zdjęciem.

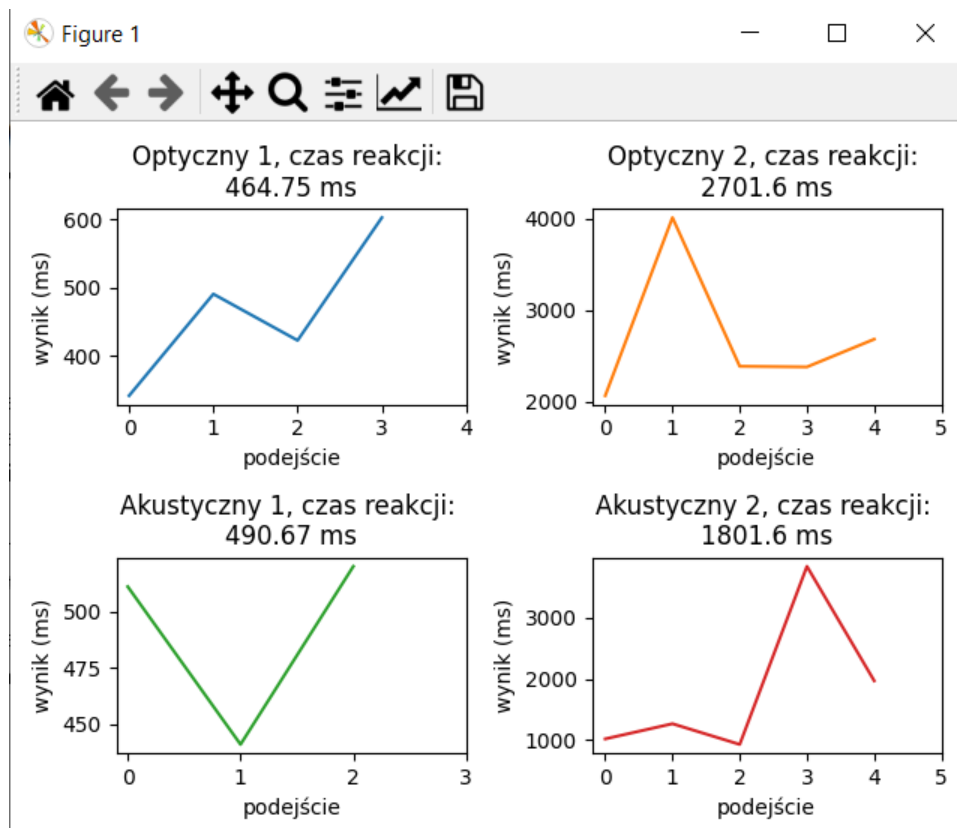
#### 4. Podsumowanie



Rys.6 ilustracja pętli okienek testowych

Wszystkie testy polegają na tej samej pętli programu. Po przejściu przez okienko informacyjne, a następnie kliknięcie przycisku 'Start' w danym teście, zostaje wykonana 'pętla testu', w której losowy timer zmienia pewne działanie w okienku, na które użytkownik powinien zareagować. Jeśli wybierze złą opcję (o ile taka w danym teście występuje) to zostanie o tym poinformowany. W okienkach akustycznych występuje również timer pomocniczy, służący do odliczania czasu do testu na zasadzie '3, 2, 1, Start!'.

Do odmierzania czasu reakcji w każdym teście jest używany dodatkowy timer, który uruchamia się przy zmianie właściwości okienka i resetuje przy każdej iteracji testu.



Rys.7 ilustracja wykresów z wynikami z poszczególnych testów (czas odczytany w ms)

## 5. Dyskusja osiągniętych wyników

Aplikacja spełnia założenia przyjęte w punkcie nr 1 sprawozdania. Symulacja działania testów psychomotorycznych została zrealizowana. Do utworzenia okienek aplikacji została użyta biblioteka PyQt5, Qt5 jest standardem dla wielu nowoczesnych aplikacji i posiada wsparcie dla wielu języków programowania.

Aplikacja jest prosta i przejrzysta, na bieżąco informuje użytkownika o swoim działaniu.

Pomijając test akustyczny prosty, testy w aplikacji są zdecydowanie kreatywne.

Okienka mogłyby być trochę większe, zdecydowanie checkboxy w teście optycznym złożonym są zbyt małe (z drugiej strony każdy użytkownik posiada to utrudnienie, więc wydaje się to sprawiedliwe).

Niektóre testy w aplikacji można 'oszukać':

- w teście optycznym prostym oraz teście akustycznym prostym użytkownik może 'spamować klikanie' na przycisk, który jest nieaktywny i w ten sposób kliknąć na niego od razu kiedy się uaktywni i w ten sposób uzyskać lepszy, niemal natychmiastowy wynik
- w teście akustycznym złożonym przed sygnałem dźwiękowym użytkownik może ustawić położenie kursora myszy, żeby być bliżej przycisków

#### **4. Wnioski**

Aplikacja spełnia założenia przyjęte w punkcie nr 1 sprawozdania. Symulacja działania przesyłu danych przez RS232 została zrealizowana. Do utworzenia okienek aplikacji została użyta biblioteka PyQt5, Qt5 jest standardem dla wielu nowoczesnych aplikacji i posiada wsparcie dla wielu języków programowania.

Aplikacja nie bierze pod uwagę problemów, które mogłyby wystąpić w rzeczywistym przesyśle danych przez RS232.

Kodowanie danych nie jest optymalne – zdecydowanie można by znaleźć sposób, żeby znaki ASCII, które składają się jedynie z 8 bitów nie musiały być konwertowane niepotrzebnie na 2 ciągi po 8 bitów.

Aplikacja jest prosta i przejrzysta, a nielegalne wyrazy zostają wykryte nawet jeśli słowa są ze sobą połączone.

W celu optymalizacji można by używać biblioteki numpy do niektórych operacji na listach, ponieważ biblioteka numpy jest napisana w C.