

18.09.2020

1. Marcin Jasiński
2. Miłosz Szkudlarek
3. Jakub Kusiowski
4. Nikodem Janaszak

Podstawy Teleinformatyki – Dokumentacja
Projektu

Movement Tracer

1.	Wstęp	3
1.1	Ogólna charakterystyka zadania (tematyki pracy)	3
1.2	Uzasadnienie podjęcia tematu.	3
2.	Cel i zakres pracy	3
2.1	Przeznaczenie i zadania (podstawowe funkcje) projektowanego systemu	3
2.2	Funkcjonalności:	3
2.3	Określenie uwzględnianych i pomijanych aspektów zagadnienia	4
2.4	Udział poszczególnych członków zespołu w realizacji zadania	5
3.	Metody konstruowania systemu	5
3.1	Metody pracy zespołowej	5
3.2	Powody wybrania wspomnianej metodyki:	6
3.4	Środki implementacji	6
4.	Budowa systemu	6
4.1	Ogólna architektura systemu	6
4.2	Opis poszczególnych modułów (struktura, funkcje).	7
4.3	Parametry	7
4.4	Testy	8
4.4.1	Wychodzenie kursorem poza ekran	8
4.4.2	Podanie złych parametrów	8
5.	Użytkowanie (Instrukcja)	9
5.1	Rady	9
5.2	Sterowanie	9
5.3	Dobieranie parametrów przy uruchomieniu programu	10
5.4	Rysowanie	11
5.5	Tryb pauzy	13
6.	Podsumowanie	14
6.1	Krótki opis celów zrealizowanych i niezrealizowanych	14
6.2	Wskazanie możliwych kierunków rozbudowy systemu	15
6.3	Wnioski i obserwacje dot. pracy zespołowej.	15
7.	Literatura	15

Kod dostępny jest pod adresem: <https://github.com/mjaelo/Movement-Tracer>

1. Wstęp

1.1 Ogólna charakterystyka zadania (tematyki pracy)

Program służy do rysowania ruchem przez kamerę komputera. Program szuka punktów najbardziej zbliżonych do pożądanego koloru i rysuje na nich wybranym kolorem. Kontrola rysowania jest przez wybrany kontroler o ustalonym kolorze. Przykładowo, przy szukaniu białego koloru, można użyć białego światła komórki. Rysowanie odbywa się z dopasowanymi parametrami.

1.2 Uzasadnienie podjęcia tematu.

Użytkowanie systemów wizyjnych i obróbka obrazów jest tematem, który nas wszystkich zainteresował. Widząc możliwość wykonania projektu, który wpasował się w nasze umiejętności, postanowiliśmy podjąć się wykonania owego tematu.

2. Cel i zakres pracy

2.1 Przeznaczenie i zadania (podstawowe funkcje) projektowanego systemu

Program jest przeznaczony do użytkowników zainteresowanych rysowaniem na obrazie kamery. Jako użytkownik, można dobrać sobie odpowiednie parametry i użyć wybranego przez siebie kontrolera o charakterystycznym kolorze tak, aby program go rozpoznał.

2.2 Funkcjonalności:

Wymagania funkcjonalne:

- Ustalanie początkowych parametrów, lub wybieranie domyślnych
- Otwieranie ekranu z obrazem z kamery, na którym można rysować. Program najpierw szuka na całym obrazie punktu najbardziej zbliżonego kolorem do poszukiwanego, a potem szuka punktów w ustalonej odległości od niego.

- W każdym momencie można zatrzymać rysowanie, naciskając pauzę. W konsoli można zmienić obecne parametry. program przedstawi obecne wartości parametrów przez (def=xxxx)
- Kolor rysowania można zmienić wchodząc w ustawienia, bądź najeżdżając kursorem na kolorowe prostokąty na górze ekranu
- Aby usunąć swój rysunek, należy najechać na skrzynkę "clear all" po lewej górnej części ekranu
- najnowszy punkt jest oznaczony przez kolor przeciwny do koloru rysowania
- gdy w obszarze poszukiwania (parametr search_area) nie znajduje się punkt wystarczająco zbliżony do poszukiwanego koloru(parametry tolerance oraz value), piksel pozostaje w bezruchu
- Dostępna jest alternatywna metoda rysowania, poprzez użycie klawiszy WSAD do poruszania kursorem. Jest to opcja przeznaczona, gdy chce się dokładnie zmienić pozycję kursora, bez używania kontrolera, bądź w przypadku wystąpienia problemów z nim. Jest to też metoda przydatna przy testowaniu oprogramowania.

Wymagania pozafunkcjonalne:

- Bezpieczeństwo: Przez brak połączenia z internetem, użytkownik nie musi martwić się o narażenie swoich danych
- Użyteczność: możliwość rysowania ruchem kontrolera po obrazie kamery z ustalonymi parametrami
- Wydajność: zabezpieczono utratę pamięci operacyjnej poprzez ograniczenie rozmiaru tablicy przechowującej rysowane piksele do rozmiaru obrazu
- Niezawodność: Zabezpieczono użytkownika przed wychodzeniem kontrolerem poza ekran

2.3 Określenie uwzględnianych i pomijanych aspektów zagadnienia

Jeśli chodzi o uwzględniane aspekty zagadnienia - większość została wymieniona w funkcjonalnościach. Dodatkowo, jak najwierniej staraliśmy się odwzorować doświadczenie rysowania, poprzez zastosowanie odpowiednich rozwiązań wizualnych.

Aspekty pominięte to głównie zaprogramowanie gumki, usuwającej narysowane punkty. Ponadto, planowaliśmy dodać opcję wyłączania skrzynek do zmiany kolorów, jednak nie byliśmy pewni, co do słuszności tej koncepcji.

2.4 Udział poszczególnych członków zespołu w realizacji zadania

Udział w przygotowaniu aplikacji został podzielony tak jak przedstawiono poniżej, każdy z członków zespołu miał do zrealizowania zadanie, w przypadku problemów mógł liczyć na pomoc. Ostatecznie aplikacja powstała w trochę innym czasie niż zakładaliśmy, ponieważ napotkaliśmy nieoczekiwane trudności związane z technologią, którą wykorzystywaliśmy, ale ostatecznie udało nam się zrealizować projekt.

Nikodem	Miłosz	Jakub	Marcin
Interfejs graficzny	Funkcje rysowania	Obsługa kontrolera	Śledzenie kontrolera na obrazie kamery

3. Metody konstruowania systemu

3.1 Metody pracy zespołowej

Wybraliśmy metodykę typu Scrum. Jest ona metodyką zwinną, która polega na dzieleniu rozwoju produktu na sprinty. Po każdym sprincie zespół powinien być w stanie dostarczyć w pełni działającą wersję programu. Sprinty nie powinny trwać dłużej niż miesiąc. W założeniach metodyki Scrum są również codzienne spotkania trwające 15 minut i konsultacje pomiędzy programistami.

Ze względu na akademicki charakter projektu oraz epidemię, pełne spełnienie założeń metodyki Scrum nie było możliwe. Spotkania odbywały się co jakiś czas, ale trwały dłużej niż kwadrans. Podczas nich ustalono priorytetowe zadania do wykonania. Ponadto przyjęto regularną komunikację pomiędzy programistami.

3.2 Powody wybrania wspomnianej metodyki:

- Możliwość bieżącego nanoszenia zmian w projekcie w czasie jego realizacji
- Możliwość bieżącej regulacji zakresu projektu i zamknięcia go w dowolnym momencie
- Bieżący monitoring i ocena powstającego produktu
- Formalności i biurokracja zmniejszone do minimum
- Ponoszenie wydatków wyłącznie za wykonaną pracę.
- Szybsza i skuteczniejsza realizacja projektu

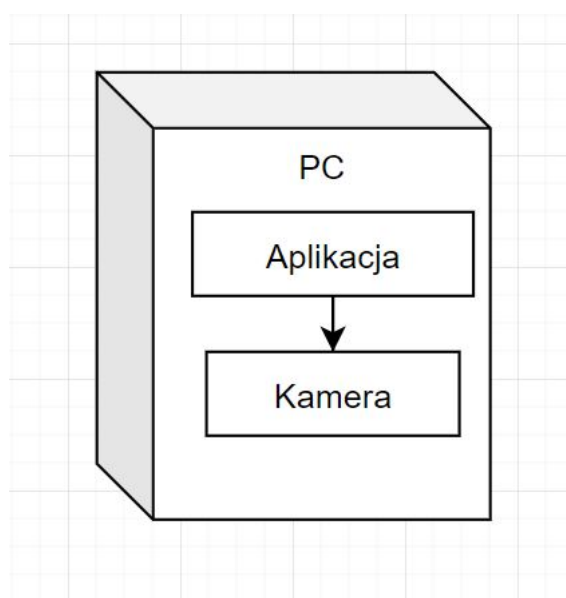
3.4 Środki implementacji

- Python 3.7 jako język programowania
- Biblioteka OpenCV jako podstawa działania przetwarzania obrazów
- Biblioteka PyInstaller do stworzenia exe

4. Budowa systemu

4.1 Ogólna architektura systemu

Aplikacja jest z założenia nie jest aplikacją internetową, więc jej działanie odbywa się w obszarze jednego urządzenia. Aplikacja korzysta z kamery zamontowanej w komputerze. Diagram wdrożenia przedstawiono na rysunku 4.1.1



Rysunek 4.1.1 Diagram wdrożenia

4.2 Opis poszczególnych modułów (struktura, funkcje).

- Parameters - klasa z parametrami rysowania. Opis w punkcie 4.3
- main - Funkcja ustalająca parametry wstępne
- config - Funkcja z menu konfiguracji parametrów rysowania
- searchcam - Funkcja z pętlą do rysowania. Pobiera współrzędne piksela z poniższych funkcji, dodaje go do listy pikseli z kolorem. Następnie koloruje je i piksele wokół nich na wyznaczony kolor.
- findclosestpix_part - Funkcja znajdująca współrzędne piksela najbardziej zbliżonego do poszukiwanego koloru z części obrazka wokół poprzedniego piksela. Szybsza, ale zawodniejsza. Używana w standardowym poszukiwaniu piksela.
- findclosestpix_all - Funkcja znajdująca współrzędne piksela najbardziej zbliżonego do poszukiwanego koloru z całego obrazka. Wolniejsza, ale powoduje mniej błędów. Używana na starcie gry i w przypadku pojawienia się błędu

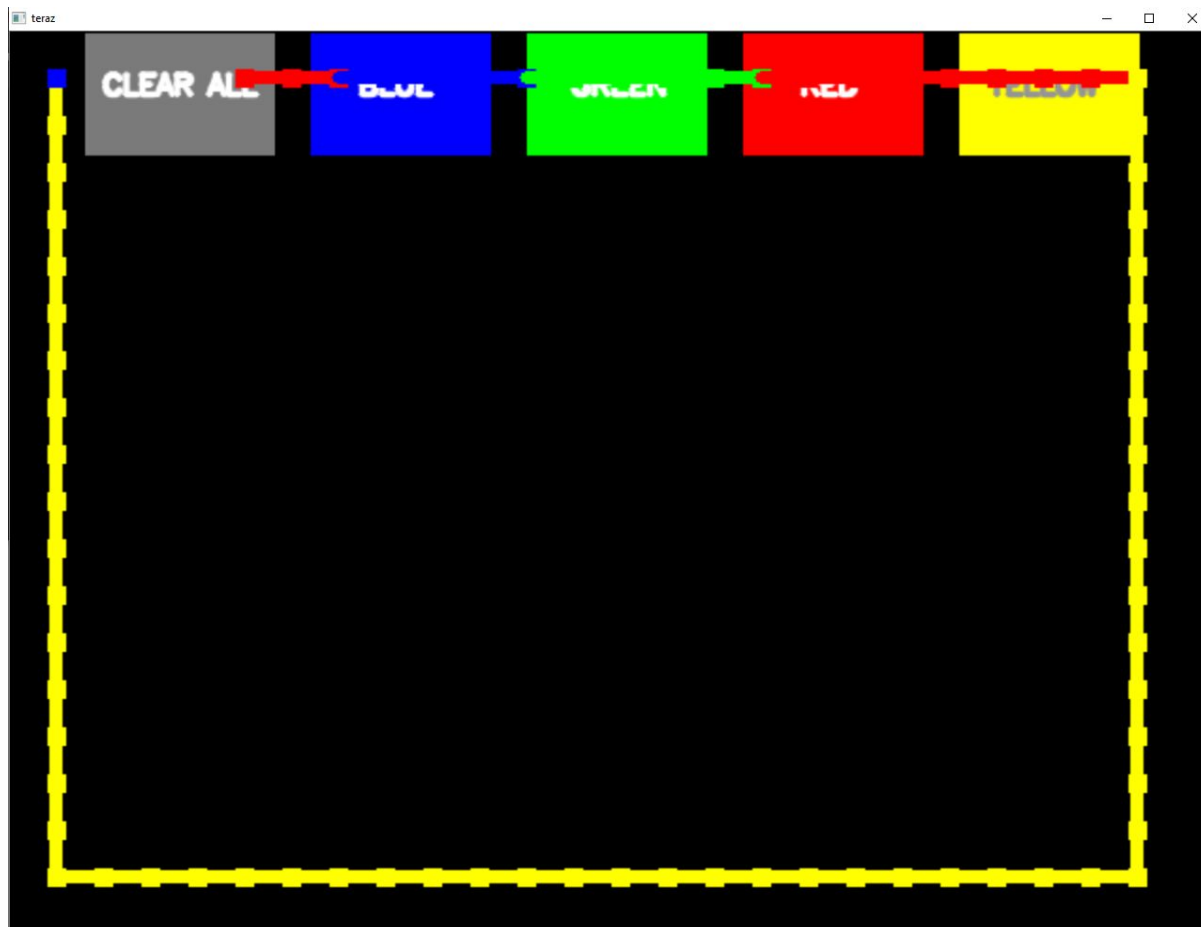
4.3 Parametry

program posiada klasę Parameters, której wartości oddziałują na jego działanie. Wartości można zmienić przy uruchomieniu programu, bądź w trybie pauzy

- value - szukany kolor
- thickness - grubość linii
- search_area - rozmiar pola poszukiwań nowego punktu wokół poprzedniego punktu
- draw_color - kolor rysowania
- max_table_size - długość, ile pikseli naraz będzie pokolorowanych
- tolerance - poziom tolerancji koloru. Im wyższy, tym większa szansa, że znajdzie pasujący piksel do koloru. Oznacza to też zmniejszona stabilność
- flipped - tryb lustrzanego obrazu. przy podaniu enter przyjmuje wartość False, w przeciwnym wypadku True.

4.4 Testy

4.4.1 Wychodzenie kursorem poza ekran



Rysunek 4.4.1.1 ścieżka narysowania po granicach palety.

Program nie dopuszcza wyjście poza granice obrazu. kursor jest zatrzymany przy dokonaniu tej próby.

4.4.2 Podanie złych parametrów

```
czy chcesz poznać instrukcje? 1=tak asd  
pozostale = domyslne parametry, 1 = podaj kolor i grubosc, 2 = podaj dokladne parametry asda  
wymiarzy obrazu: 480 640
```

Rysunek 4.4.2.1 przykład podania błędnych wartości przy wyborze opcji podawania parametrów

W przypadku podaniu złych parametrów (w tym przypadku liter, gd oczekiwane są cyfry) program kontynuuje działanie z domyślnymi parametrami


```

USTAWIENIA
1 = grubosc linii, 2 = porzadzany kolor, 3 = kolor rysowania. 4 = poleposzukiwan
5 = maksymalna ilosc kolorowych pikseli, 6= poziom tolerancji szukania, 7= tryb flipped2
podaj wartosci RGB porzadanego koloru (def=[255, 255, 255])
B dsf
wrong input

```

Rysunek 4.4.2.2 przykład podania błędnych wartości przy podawaniu parametrów

Przy podawaniu wartości parametrów, w przypadku błędu, program ostrzega o błędnym wejściu i kontynuuje działanie z poprawnymi parametrami

```

czy chcesz poznać instrukcje? 1=tak
pozostale = domyslne parametry, 1 = podaj kolor i grubosc, 2 = podaj dokladne parametry1
podaj wartosci RGB porzadanego koloru
B sada
wrong input
pozostale = domyslne parametry, 1 = podaj kolor i grubosc, 2 = podaj dokladne parametry

```

Rysunek 4.4.2.3 przykład podania błędnych wartości przy podawaniu parametrów po uruchomieniu programu

Jeśli jednak niepoprawne parametry zostaną podane przy uruchomieniu programu, program będzie próbować wymusić od użytkownika poprawną odpowiedź.

5. Użytkowanie (Instrukcja)

5.1 Rady

- Zaleca się wybranie kontrolera o konkretnym kolorze, np. białe światło z komórki przy szukaniu białego koloru
- Przy ustawianiu kolorów pamiętaj, że przyjmują one wartości 0-255 w skalach Blue Green Red. Należy podać wartość wszystkich pól, jako że kolor wypadkowy jest efektem złożenia tych wartości. [0,0,0] - kolor czarny, [255,255,255] - kolor biały
- Aby uruchomić program, należy posiadać środowisko python i uruchomić Tracer.py, bądź uruchomić Tracer.exe w folderze dist
- skrócona wersja instrukcji dostępna jest przy uruchomieniu programu

5.2 Sterowanie

- wybrany kontroler o charakterystycznym kolorze do rysowania, aby program mógł śledzić jego ruch

- klawiatura do podawania parametrów w konsoli w trybie pauzy, bądź przy uruchomieniu programu.
- SPACE = Nacisnąć w oknie obrazu. Pauza, ustawianie parametrów na konsoli.
- WSAD = Nacisnąć w oknie obrazu. Alternatywne poruszanie się po palecie

5.3 Dobieranie parametrów przy uruchomieniu programu

```
czy chcesz poznać instrukcje? 1=tak
pozostale = domyslne parametry, 1 = podaj kolor i grubosc, 2 = podaj dokladne parametry1
podaj wartosci RGB porzadanego koloru
B 255
G 255
R 255
podaj grubosc linii (def=10)20
```

Rysunek 5.3.1 przykład podawania najważniejszych parametrów przy uruchomieniu programu

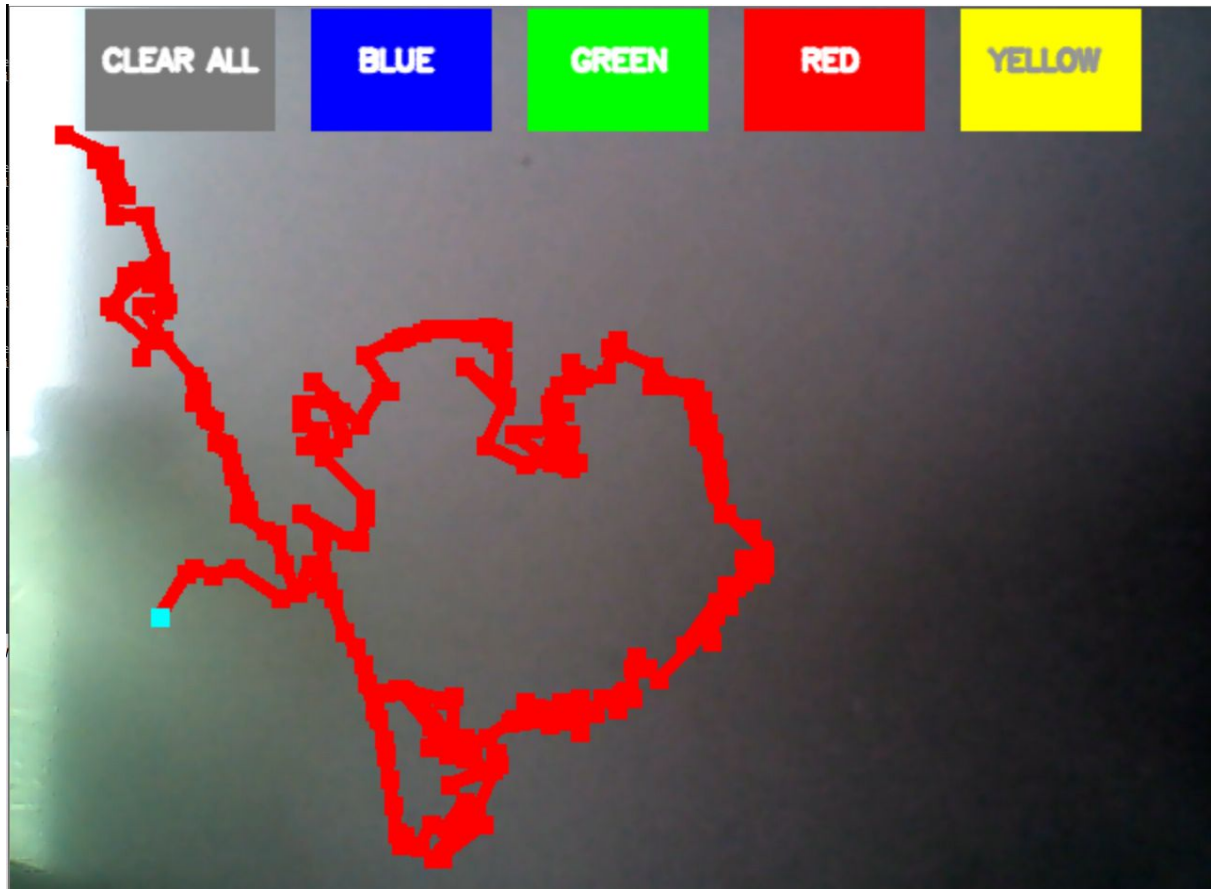
```
C:\studia\sem6\PT\Tracer\dist\Tracer.exe
czy chcesz poznać instrukcje? 1=tak
pozostale = domyslne parametry, 1 = podaj kolor i grubosc, 2 = podaj dokladne parametry2
podaj grubosc linii (def=10)10
podaj ilosc pixeli do rysowania (def=500)1000
podaj poziom tolerancji koloru (def=6000)6000
tryb flipped ENTER=False (def=True)
podaj wartosci RGB porzadanego koloru (def=[255, 255, 255])
B 255
G 255
R 255
podaj wartosci RGB koloru rysowania (def=[0, 0, 255])
B 0
G 255
R 100
podaj rozmiar pola przeszukiwan (def=[50, 50])
X 50
Y 50
```

Rysunek 5.3.2 przykład podawania dokładnych parametrów po uruchomieniu programu

Domyślne wartości parametrów są przedstawione poprzez (def=xxxx), gdzie xxxx jest obecną wartością.

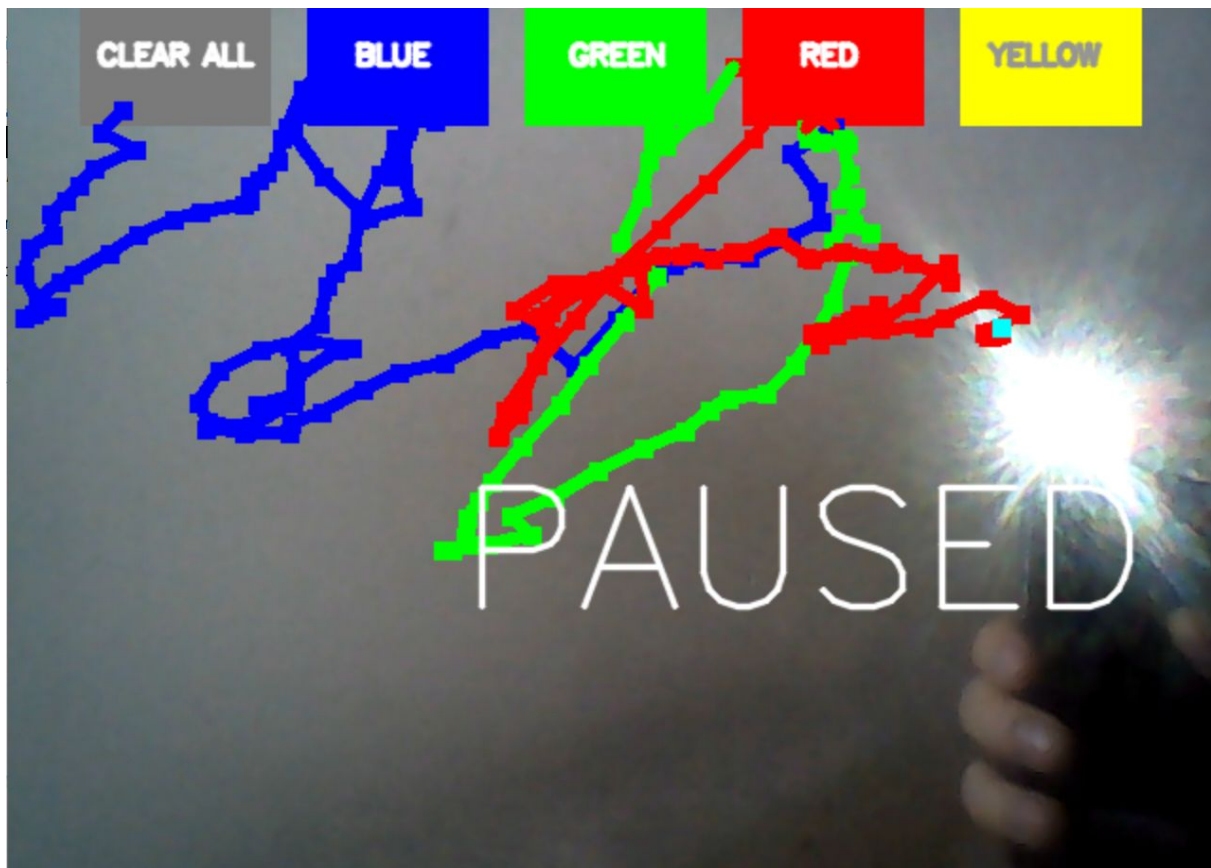
tryb flipped jako jedyny przyjmuje wszystkie wartości podane i odczytuje je jako True. Jedynie naciśnięcie enter bez podania niczego rezultuje w wartości False

5.4 Rysowanie



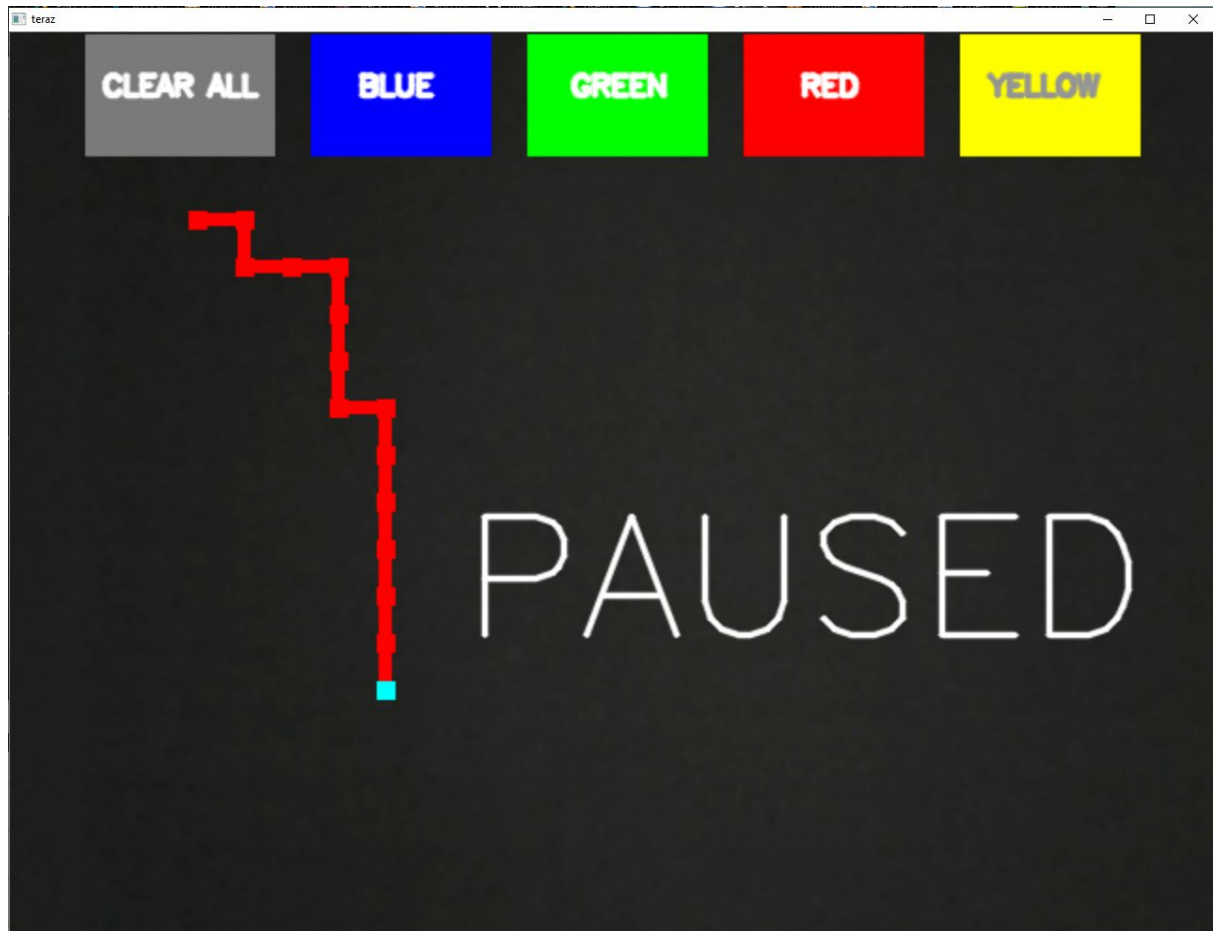
Rysunek 5.4.1 przykład rysowania prostego kształtu (serca) przy użyciu latarki komórki przy wykrywaniu białego światła

na zrzucie ekranu kursor już się nie porusza, gdyż nie znalazł kontrolera o kolorze wystarczająco zbliżonym do szukanego (białego) w swojej najbliższej okolicy (parametr search_area)



Rysunek 5.4.2 przykład rysowania białym kontrolerem (latarką komórki) ze zmianą koloru poprzez kolorowe skrzynki

5.5 Tryb pauzy



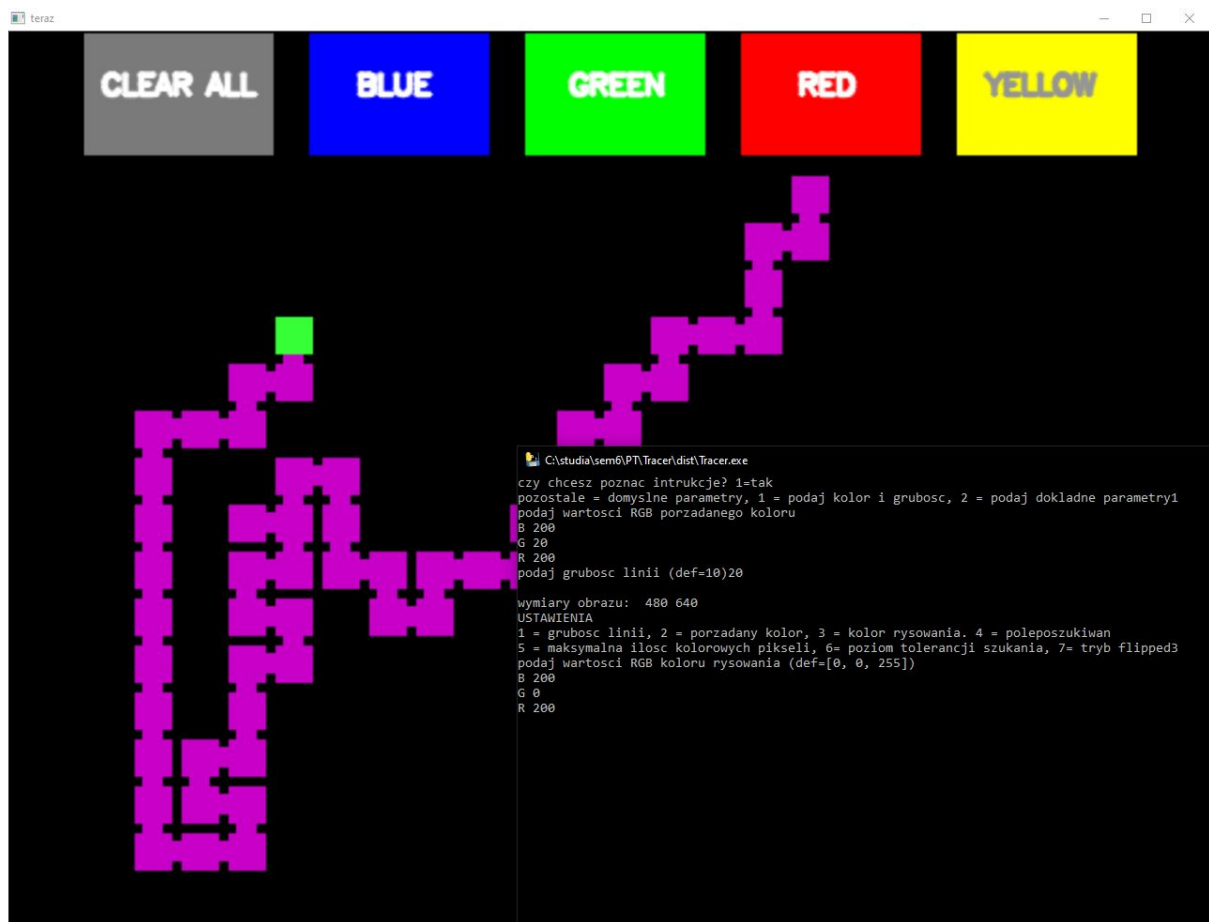
Rysunek 5.5.1 przykład wejścia w tryb pauzy przez naciśnięcie SPACE

na ekranie pokazuje się napis informujący o pauzie gry. czytanie obrazu z kamery oraz rysowanie zostało zatrzymane, aż do czasu ustalenia parametrów.

```
USTAWIENIA
1 = grubosc linii, 2 = porzadzany kolor, 3 = kolor rysowania. 4 = poleposzukiwan
5 = maksymalna ilosc kolorowych pikseli, 6= poziom tolerancji szukania, 7= tryb flipped
```

Rysunek 5.5.2 przykład dostępnych parametrów do zmiany w ustawieniach na konsoli w trybie pauzy

Z pauzy wyjść można, zmieniając parametry, bądź podając wartość nieodpowiadającą żadnego parametrowi. Samo naciśnięcie ENTER jest również skuteczne



Rysunek 5.5.3 przykład zmiany parametru obecnego koloru na [200,0,200]

6. Podsumowanie

6.1 Krótki opis celów zrealizowanych i niezrealizowanych

Zrealizowane: Rysowanie ruchem kontrolera po obrazie kamery z ustalonymi parametrami przy uruchomieniu programu, bądź edytowanymi w trybie pauzy. Dana jest też dodatkowa możliwość zmiany kolorów przez najechanie kursorem na skrzynki zmiany kolorów. Użytkownik może także usunąć swoją pracę, najeżdżając kursorem na skrzynkę “clear all”

Niezrealizowane: Aspekty pominięte to głównie zaprogramowanie gumki, usuwającej narysowane punkty. Ponadto, planowaliśmy dodać opcję edycji skrzynek do zmiany kolorów, jednak nie byliśmy pewni, co do słuszności tej koncepcji.

6.2 Wskazanie możliwych kierunków rozbudowy systemu

Można rozbudowywać system, dodając funkcjonalności niezrealizowane, zwłaszcza gumka. Pojawiło się też kilka niezrealizowanych koncepcji, przez odrzucenie ich praktyczności, jednak możliwe jest ich ponowne rozpatrzenie, jak wyłączanie skrzynek do zmiany kolorów w ustawieniach, zmienianie ich rozmiarów, ilości i wartości kolorów. możliwe jest także zmienienie obszarów palety i zapis obrazu do pliku. Inną możliwością jest dostosowywanie rozmiaru obrazu.

6.3 Wnioski i obserwacje dot. pracy zespołowej.

Przez pracę zdalną była utrudniona komunikacja, ale udało się nam z niej dobrze wywiązać. Udało nam się zachować podstawowy kontakt, jednak zdarzały się problemy typu: brak dostępu do najnowszych danych, czy problem spotkania się w czasie odpowiadającym obu stronom. Jednak kiedy ktoś napotkał problem w implementacji swojej funkcjonalności, nieraz bardziej doświadczony członek zespołu spotykał się z nim, by mu pomóc.

7. Literatura

[1] <https://docs.opencv.org/> - Dokumentacja do biblioteki OpenCV

[3] <https://www.pyinstaller.org/documentation.html> - Dokumentacja do biblioteki pyinstaller