

Projekt: Zbiór punktów przecięcia odcinków

Skład zespołu nr 3,

Bartosz Fira

146951,

Lista wykonanych zadań w projekcie

- *Zrozumienie problemu: Przeprowadzenie badania i analizy problemu zbiór punktów przecięcia odcinków. Zapoznanie się z istniejącymi algorytmami i podejściami do tego problemu.*
- *Projektowanie algorytmu: Opracowanie algorytmu rozwiązującego problem. Przemyślenie różnych koncepcji i technik, aby znaleźć efektywne rozwiązanie.*
- *Dokumentacja i raportowanie: Dokumentowanie projektu, opis algorytmu, wykorzystanych technik i implementacji.*

Mikołaj Glistak

146724,

Lista wykonanych zadań w projekcie

- *Zrozumienie problemu: Przeprowadzenie badania i analizy problemu zbiór punktów przecięcia odcinków. Zapoznanie się z istniejącymi algorytmami i podejściami do tego problemu.*
- *Projektowanie algorytmu: Opracowanie algorytmu rozwiązującego problem. Przemyślenie różnych koncepcji i technik, aby znaleźć efektywne rozwiązanie.*
- *Testowanie i debugowanie: Przeprowadzenie różnorodnych testów na zaimplementowanym programie. Sprawdzenie poprawności wyników dla różnych scenariuszy i zestawów danych testowych. Debugowanie i naprawianie ewentualnych błędów.*

Piotr Swirkaitis

142410,

Lista wykonanych zadań w projekcie

- *Zrozumienie problemu: Przeprowadzenie badania i analizy problemu zbiór punktów przecięcia odcinków. Zapoznanie się z istniejącymi algorytmami i podejściami do tego problemu.*
- *Projektowanie algorytmu: Opracowanie algorytmu rozwiązującego problem. Przemyślenie różnych koncepcji i technik, aby znaleźć efektywne rozwiązanie.*
- *Implementacja algorytmu: Napisanie kodu programu, który implementuje opracowany algorytm. Używanie odpowiednich struktur danych i operacji geometrycznych do wykrywania i obliczania punktów przecięcia odcinków.*

Manual

Zadanie, które program ma realizować

Program wyznacza przecięcie dwóch odcinków na płaszczyźnie:

- Informuje, czy odcinki przecinają się (TAK/NIE).
- Jeżeli odcinki przecinają się, to program określa zbiór, na którym następuje przecięcie:
 - punkt i jego współrzędne, albo
 - odcinek i współrzędne jego końców.

Dane wejściowe: współrzędne końców dwóch odcinków.

Lista opcji do wyboru z krótkim opisem każdej z nich

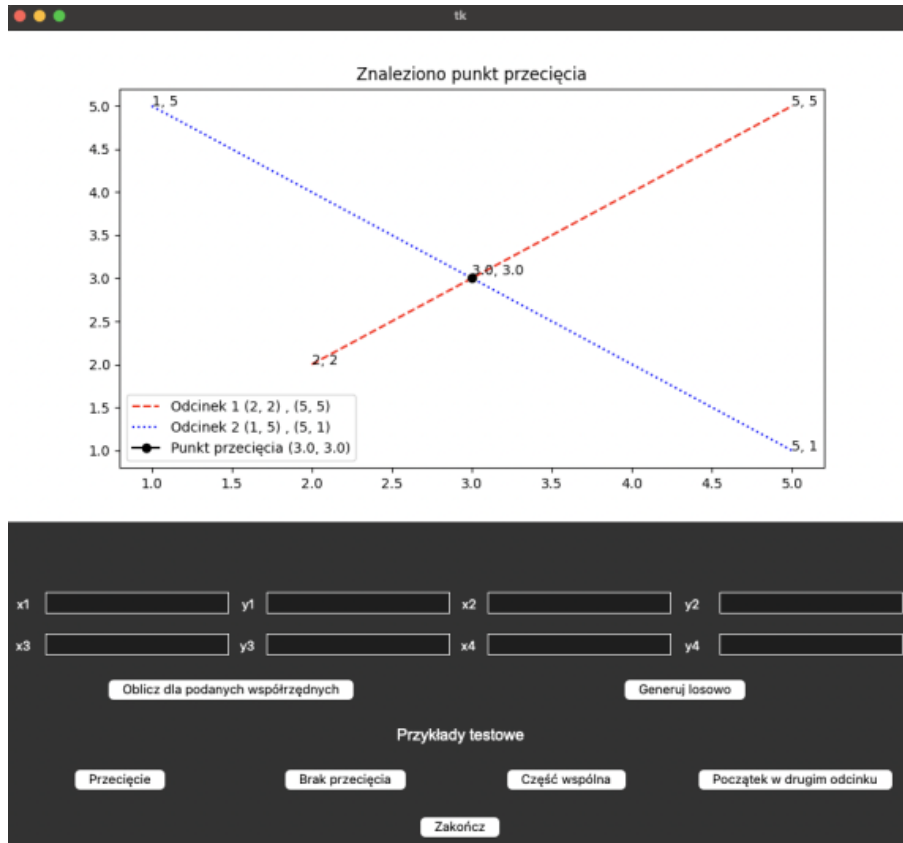
Program nie narzuca żadnych ograniczeń dla wprowadzanych liczb, co oznacza, że użytkownik może wprowadzać liczby o dowolnej wartości. Brak tych ograniczeń daje użytkownikowi pełną swobodę w wyborze wartości liczbowych, które chce użyć w programie. Program nie dopuszcza do wprowadzenia innego typu danych niż liczby.

Program, podczas generowania liczb losowych, używa definiowanego zakresu od -20 do 20. Oznacza to, że generowane liczby losowe będą znajdować się w przedziale od -20 do 20, włącznie z tymi wartościami.

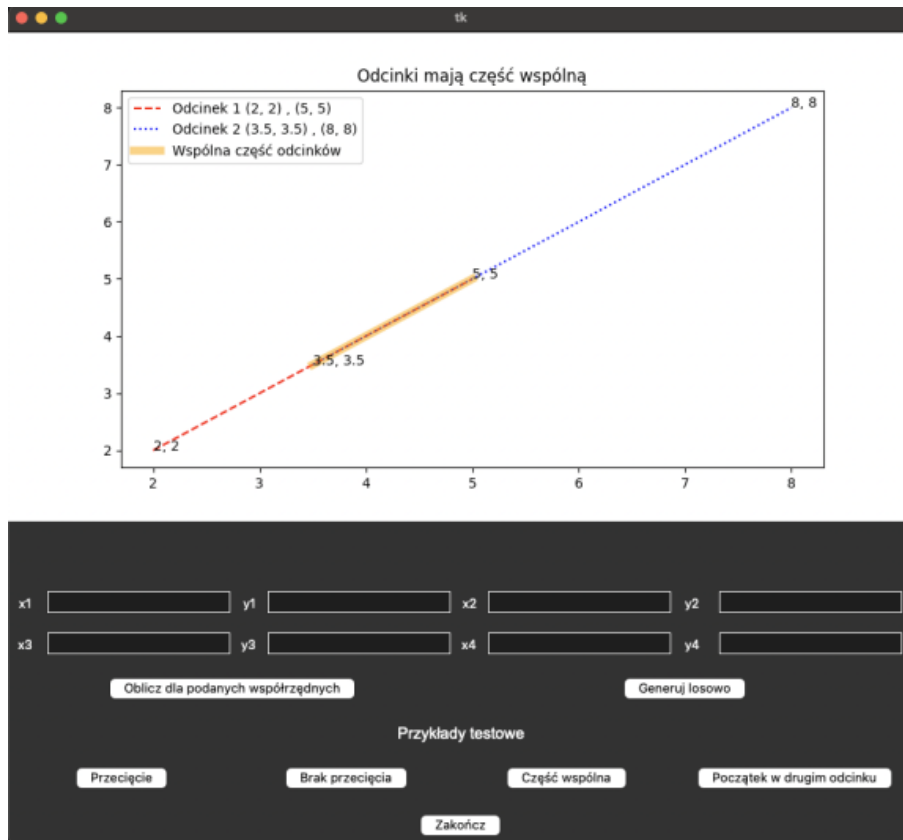
Opisy przycisków w programie, który przyjmuje punkty dwóch odcinków i oblicza punkt przecięcia:

1. Przycisk "Oblicz dla podanych współrzędnych": Po kliknięciu tego przycisku, program oblicza punkt przecięcia dla dwóch odcinków na podstawie wprowadzonych współrzędnych punktów.
2. Przycisk "Generuj losowe": Przycisk generuje losowe wartości współrzędnych dla punktów odcinków w określonym zakresie. Następnie program oblicza punkt przecięcia dla wygenerowanych odcinków.
3. Przycisk "Przecięcie": Klikając ten przycisk, program oblicza punkt przecięcia dla dwóch wcześniej zdefiniowanych odcinków, które zostały narzucone z góry. Przycisk ten jest przeznaczony do prezentacji przypadku, w którym odcinki faktycznie się przecinają.
4. Przycisk "Brak przecięcia": Ten przycisk służy do prezentacji przykładu, gdzie odcinki nie przecinają się. Po kliknięciu program informuje użytkownika, że odcinki nie mają punktu przecięcia.
5. Przycisk "Część wspólna": Przycisk ten jest przeznaczony do prezentacji przypadku, w którym istnieje odcinek, który jest wspólny dla obu odcinków.
6. Przycisk "Początek w drugim odcinku": Po kliknięciu tego przycisku, program prezentuje przypadek, gdzie początek odcinka znajduje się w drugim odcinku.

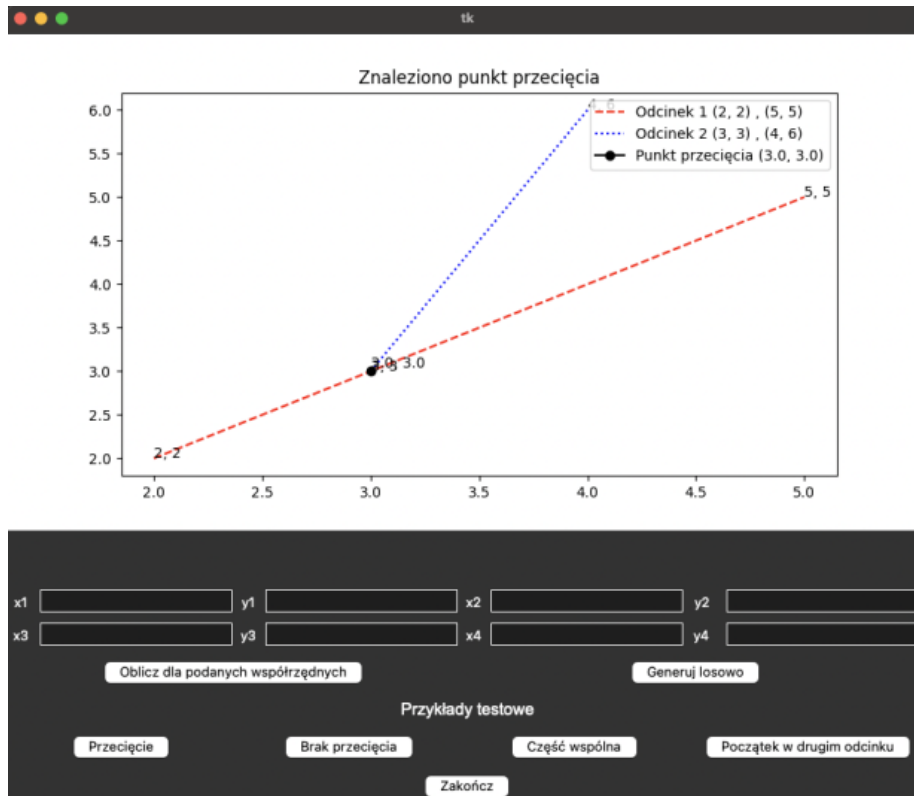
Zrzuty z aplikacji:



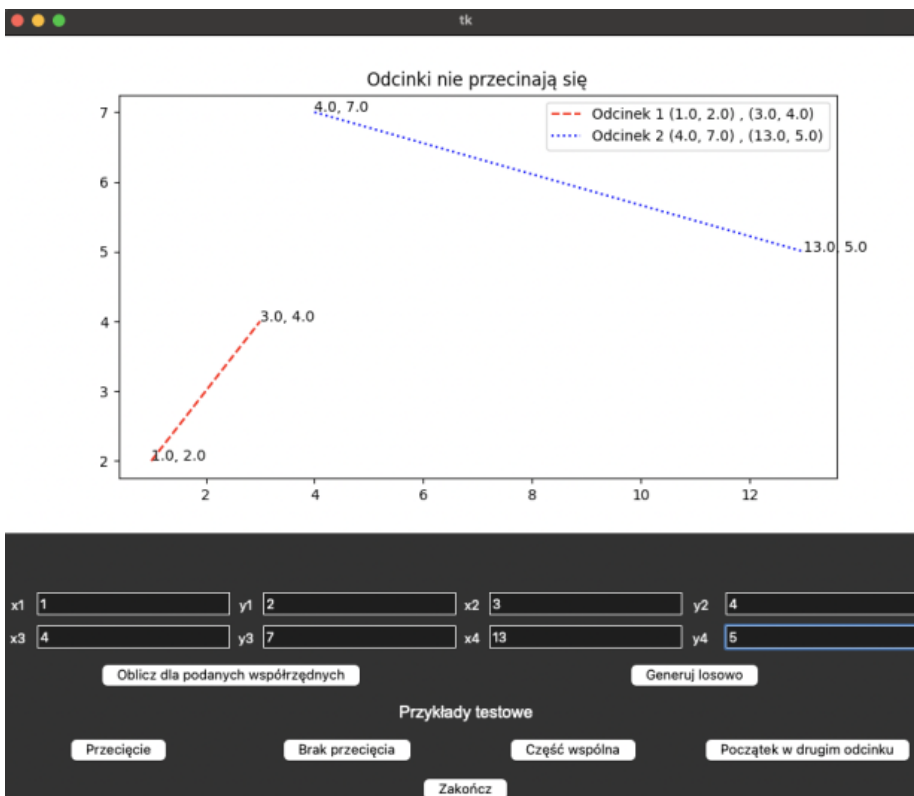
Rys. 1. Podane odcinki: Odcinek 1 (2,2).(5,5), Odcinek 2(1,5).(5,1)



Rys. 2. Podane odcinki: Odcinek 1 (2,2).(5,5), Odcinek 2(3,5;3,5).(0,0)



Rys. 3. Podane odcinki: Odcinek 1 (2,2).(5,5), Odcinek 2(3,3).(4,6)



Rys. 4. Podane odcinki: Odcinek 1 (1,2).(3,4), Odcinek 2(4,7).(13,5)

Opis kodu

Lista plików z kodem źródłowym wchodzących w skład programu

- `segment_intersection_tkinter.py`

Schemat algorytmu albo pseudokod (z odniesieniami do kodu programu)

Funkcja **check_intersection** służy do sprawdzania czy dwa odcinki przecinają się, a także określania punktu przecięcia, jeśli istnieje. Oto opis krok po kroku działania funkcji:

1. Konstruujemy składowe wektorów z odcinków 1 i 2, które są potrzebne do dalszych obliczeń. Składowe te są obliczane na podstawie współrzędnych początku i końca każdego odcinka.
2. Tworzymy wektor 3, który jest wektorem od początku odcinka 2 do początku odcinka 1. Obliczamy jego składowe na podstawie współrzędnych punktów.
3. Obliczamy iloczyn wektorowy wektorów 1 i 2, używając odpowiednich wzorów. Ten iloczyn pozwala nam sprawdzić czy odcinki są równoległe.
4. Obliczamy iloczyny wektorowe odcinka 1 i wektora 3 oraz odcinka 2 i wektora 3. Te iloczyny są potrzebne do dalszych obliczeń.
5. Jeżeli iloczyn wektorowy wektorów 1 i 2 jest równy 0, oznacza to, że odcinki są równoległe. W takim przypadku sprawdzamy czy wektor 3 jest równoległy do obu odcinków. Jeśli tak, oznacza to, że odcinki są współliniowe. Następnie sprawdzamy czy odcinki mają część wspólną za pomocą funkcji **find_overlapping_segment(seg_1, seg_2)**. Jeśli wektor 3 nie jest równoległy do obu odcinków, odcinki są równoległe, ale nie są współliniowe, więc zwracamy wartość **None**.
6. Jeżeli iloczyn wektorowy wektorów 1 i 2 nie jest równy 0, oznacza to, że odcinki nie są równoległe. Każdy punkt na odcinku 1 możemy przedstawić jako kombinację liniową początku odcinka i składowych wektora odcinka. Analogicznie, każdy punkt na odcinku 2 możemy przedstawić w postaci kombinacji liniowej. Sprawdzamy, czy istnieją takie wartości parametrów **t1** i **t2**, dla których równania te są spełnione. Jeżeli tak, to odcinki się przecinają.
7. Obliczamy wartości parametrów **t1** i **t2** za pomocą odpowiednich wzorów, które wynikają z redukcji układu równań.
8. Liczymy współrzędne punktu przecięcia, używając wartości **t2** i składowych wektora odcinka 1.
9. Sprawdzamy czy punkt przecięcia należy do obu odcinków. Jeśli wartości **t1** i **t2** są w przedziale od 0 do 1 (włącznie), to punkt przecięcia leży wewnątrz obu odcinków. Zwracamy ten punkt jako obiekt **Point**.
10. Jeżeli punkt przecięcia nie należy do obu odcinków, oznacza to, że odcinki nie są równoległe i nie przecinają się. Zwracamy wartość **None**.

Funkcja `find_overlapping_segment` służy do znajdowania odcinka wspólnego dla dwóch odcinków, jeśli istnieje. Oto opis krok po kroku działania funkcji:

1. Sprawdzamy warunek, czy odcinki są rozłączne. Jeśli którykolwiek z końców jednego odcinka znajduje się poza zakresem drugiego odcinka (zarówno wzdłuż osi x , jak i y), oznacza to, że odcinki są rozłączne. W takim przypadku zwracamy wartość **None**.
2. Analizując współrzędne początku i końca odcinków, znajdujemy współrzędne początku i końca odcinka wspólnego. Odcinek wspólny powstaje poprzez wybór maksymalnej wartości początku i minimalnej wartości końca wzdłuż osi x i y .
3. Obliczamy współrzędne początku odcinka wspólnego, biorąc pod uwagę maksimum z minimum współrzędnych początku odcinka 1 i początku odcinka 2 wzdłuż osi x i y .
4. Obliczamy współrzędne końca odcinka wspólnego, biorąc pod uwagę minimum z maksimum współrzędnych końca odcinka 1 i końca odcinka 2 wzdłuż osi x i y .
5. Tworzymy obiekt **Segment** z wyznaczonymi współrzędnymi początku i końca odcinka wspólnego, tworząc nowe instancje **Point** na podstawie obliczonych współrzędnych.
6. Zwracamy ten odcinek wspólny jako wynik.