dokumentace a popis algoritmu

po spuštění programu je do konzole vypsán úvodní text

pokud je aplikace otevřena pomocí .txt souboru, je považován za vstup (přednostně)

dále pokud je ve složce s aplikací soubor s názvem vstup.txt,

pokud ani jedno, otevře se dialog pro vybrání souboru

soubor je nutné načítat ve formátu:

první řádek dvě čísla oddělená pouze pomlčkou,

druhý řádek trasa prvního řidiče, třetí řádek trasa druhého řidiče,

kroky trasy oddělené čárkou, na konci ani na začátku nepřebývají znaky (např. "50W,80S,99E")

krok se musí skládat z čísla a písmena W,S,E nebo N, číslo musí být první, bez mezery.

(soubor vstup.txt přiložený k programu splňuje tyto požadavky)

Algoritmus a myšlenkový postup

procházet trasu kterou řidič projel bod po bodu a srovnávat s trasou druhého řidiče by bylo výpočtově extrémně náročné, proto musím hledat bod po rovných úsecích. (takovéto podobě jsou i data načítána)

trasa je udávána pouze pomocí jednotlivých kroků (řekněme jako vektory ve dvou kolmých směrech), proto abych mohl úseky srovnávat musím převést trasu na absolutní souřadnice.

Na začátku ze souboru načtu interval a dva řetězce dráhy.

textový řetězec dráhy rozdělím podle čárek na string[], deklaruju počáteční bod (0.0)

potom procházím každý úsek a přepočítávám polohu, přičítám k ujeté vzdálenosti, úsek ukládám do proměnné, kterou jsem si definoval (nižší souřadnice, vyšší souřadnice a konstantní souřadnice) a přidávám do seznamu (horizontálních a vertikálních úseků).

Jelikož hledáme bod pouze v daném intervalu, není třeba uvažovat začátek a konec trasy

tedy ukládám až úseky po překročení začátku intervalu (+ úsek vstupu do intervalu a totéž při výstupu z intervalu) a končím výpočet polohy po vystoupení z intervalu, protože již není co hledat za koncem.

Tímto získávám dvě trasy které začínají (a končí) obecně na různých místech.

poté co takto převedu obě trasy porovnávám horizontální úseky z první trasy s vertikálními z druhé trasy a naopak. Ve chvíli, kdy najdu první bod ukončuju hledání.

Evidentně pokud by dva rovnoběžné úseky měly společný bod, najdu minimálně jeden další společný bod mezi jedním z nich a následujícím, nebo předchozím z druhého – tedy nemusím je porovnávat. To však má jedno výjimku, a to konce a začátky porovnávaných úseků (viz. obrázek 1, tečkovaně trasa mimo interval, křížky značí krajní body) ořezaných tras

Proto pokud předchozí způsobem žádný bod algoritmus nenajde, porovná ještě krajní úseky (obou seznamů, tedy i některé ne z kraje)

Diagram, schematic

Description automatically generated

Obrázek