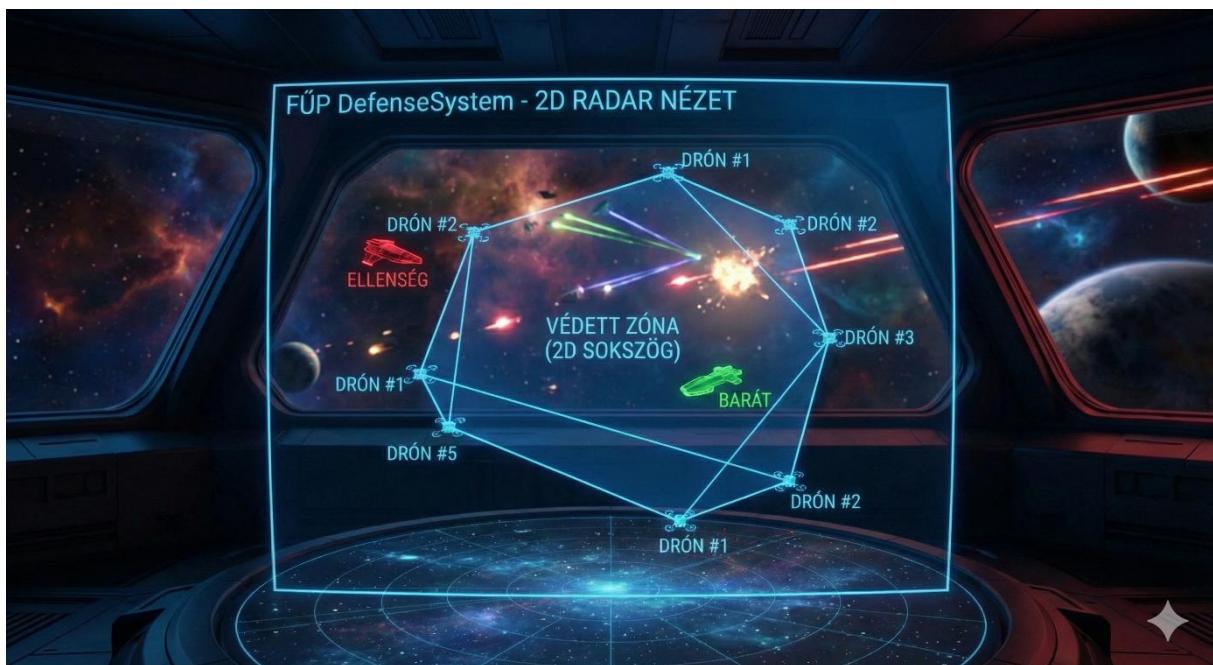


Algoritmus dömping

A Védelmi Pajzs Protokoll



Vörös kód a 7-es Szektorban

A Földi Őrvédelmi Parancsnokság (FÜP) egy új, dinamikus védelmi rendszert tesztel. A rendszer lényege, hogy nem fix falakat építünk, hanem **autonóm drónrajok** jelölik ki a védett zóna határait. A drónok folyamatosan mozognak, és véletlenszerű pozíciókat vesznek fel, de együtt egy zárt, biztonságos területet (sokszöget) alkotnak a parancsnokság körül.

A probléma az, hogy a radarunkon megjelenő ismeretlen jelekről (űrhajók, meteoritok) azonnal el kell döntenünk: **Barát vagy Ellenség?**

A szabály egyszerű: ami a drónok által kifeszített energiamezőn (sokszögön) **belül** van, az védett objektum. Ami **kívül** van, az potenciális fenyegetés.

A te feladatod a **DefenseSystem** szoftver magjának megírása. Két kritikus modult kell fejlesztened: az egyik a véletlenszerű drónokból egy összefüggő pajzsot generál, a másik pedig ezredmásodpercek alatt eldönti egy pontról, hogy átjutott-e a védelmen.

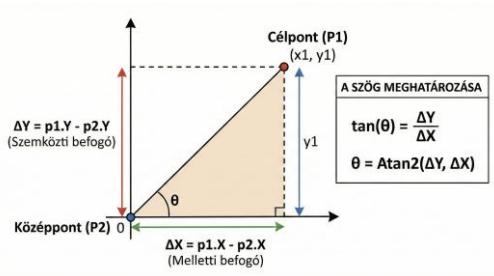
A Pajzs Generálása (Radiális söprés)

A drónok (azaz pontok generálása) véletlenszerűen szóródnak szét a térben. Ha csak úgy összekötnénk őket a generálás sorrendjében, a vonalak össze-vissza metszenék egymást, és nem kapnánk értelmezhető alakzatot.

Hogy egy egyszerű (önmetszés-mentes) sokszöget kapjunk, a **Radiális Söprés** módszerét használjuk.

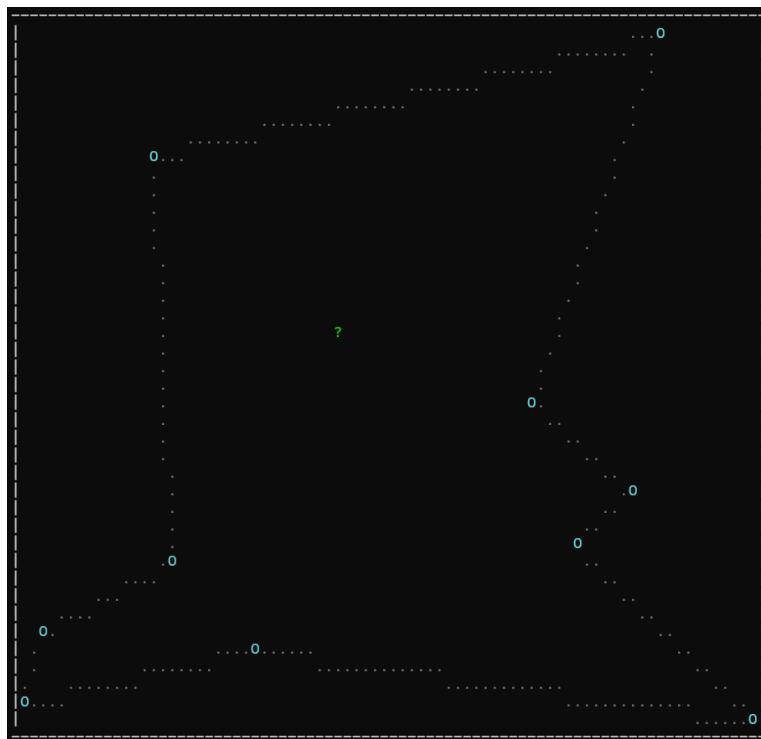


- Írj egy függvényt, ami generál N darab pontot x és y koordinátákkal.**
- Írj egy függvényt, ami meghatározza a súlypont:** Kiszámoljuk a pontfelhő "tömegközéppontját". Ez az összes X és az összes Y koordináta átlaga. Ez lesz a "tengely", ami köré a pajzs épül.
- Írj egy függvényt, ami meghatározza, hogy milyen szögben látszik a súlypontból:**
Ehhez az `Math.Atan2(dy, dx)` függvényt használjuk, ami radiánban adja vissza az értéket ($-\pi, \pi$ között).



4. **Írj egy függvényt, ami rendezи minimum kiválasztásos rendezés alapján a pontokat a szögek alapján!** A szögeket úgy kapjuk meg, hogy a súlypontnak és hozzá véve az összes pontot meghatározzuk a szögét az x tengellyel bezárva.
5. **Írj egy eljárást, ami kiírja a sokszög csúcsait!** Ha a rendezett listán sorban haladunk (mintha egy óramutató pásztázná végig a terepet), garantált, hogy az élek nem fogják keresztezni egymást.

Bizonyítás: Mivel a pontokat a középpont körüli szögük szerint rendezзük növekvő sorrendbe, a középpontból húzott sugarak minden pontot csak egyszer érintenek, szigorúan monoton sorrendben. Így lehetetlen, hogy két él keresztezze egymást, hiszen az "előrébb" lévő él pontjai nagyobb szögen vannak, mint a "hátrébb" lévő él pontjai.



A Behatoló Azonosítása (Ray Casting)

Adott egy tetszőleges, akár konkáv (behorpadt) alakzatunk, és egy tesztpontunk. Szemre könnyű megmondani, hogy belül van-e, de matematikailag nehéz, mert a "távolság a középponttól" módszer konkáv alakzatoknál hibás eredményt ad.

A Sugárkövetés (**Ray Casting**) elvét alkalmazzuk, konkrétan a vízszintes sugár módszerét.

1. **A Sugár:** Képzeletben indítunk egy végtelenbe tartó vízszintes vonalat a tesztpontból jobbra (a pozitív X irányba).
2. **Metszéspontok számlálása:** Megszámoljuk, hányszor metszi át ez a sugár a sokszög oldalait.

3. A Szabály:

- Ha **PÁRATLAN** számú falon megyünk át → **BELÜL** vagyunk. (Beleléptünk a házba).
- Ha **PÁROS** számú falon megyünk át → **KÍVÜL** vagyunk. (Beleléptünk, majd a túloldalon kiléptünk).

A következő függvényeket valósítsd meg!

1. **Írj egy függvényt, ami eldönti, hogy az adott pont magasságban (y koordináta) közrefogják a pontot!** Ha a pont a szakasz alatt vagy felett van, a vízszintes sugár biztosan nem találja el. Kizáró VAGY logika: az egyik pont legyen feljebb, a másik lejjebb: (pont_1.y > test.y) != (pont_2.y > test.y).
2. **Írj egy függvényt, ami meghatározza, hogy hol metszi a szakasz a pontunk vízszintes vonalát.**

Ez egyenes egyenletéből levezethető:

$$x_{metszés} = \frac{(x_2 - x_1)(y_{test} - y_1)}{(y_2 - y_1)} + x_1$$

3. **Írj egy függvényt, ami ellenőrzi az irányt,** mert nekünk csak azok a metszések számítanak, amik a ponttól **jobbra** vannak (a sugár iránya). Tehát megvizsgáljuk, hogy a test.X kisebb-e, mint a kiszámolt metszéspont.
4. **Írj egy függvényt, ami végig megy az összes élen** és minden alkalommal, amikor találunk egy érvényes metszést (a feltételek igazak), átbillent egy logikai változót, ami kezdetben hamis. A végén az értéke megadja a választ (Páratlan = True = Belül).

A főprogram:

1. Kérd be a pajzs csúcsainak számát
2. Generáld le a pajzsot.
3. Írd ki a generált csúcsokat.
4. Kérd be a gyanús objektum (Tesztpont) koordinátáit.
5. Írd ki eredményét!