

# Házi feladat

Programozás alapjai 3.  
Dokumentáció

**Kussa Richárd**  
**RONAOF**

2023. december 4.

---

## TARTALOM

1. A program bemutatása .....	2
2. Felhasználói útmutató.....	3
3. A program osztálydiagramja (fejlesztőknek) .....	6

## 1. A program bemutatása

Az elkészült projekt minden olyan alapvető feladatot képes elvégezni, amit a specifikációban szükségesnek ítélt meg, és még egy picit többet is. Mivel egy átlagos felhasználó nem biztos, sőt, egyáltalán nem valószínű, hogy olvasta a specifikációt, ezért illik néhány általános információt megosztanom és egy megfelelő leírást adnom a programomról.

### Mire jó a program?

2D Turing-gépek futtatására és megjelenítésére. Ezeket a gépeket sokféleképpen lehet hívni, legelterjedtebb rá a „turmesz” szó, az angol „turmite” tükörfordításából, de az egyszerűség kedvéért ebben a dokumentumban én **hangyának** fogom nevezni.

### Mit csinál a hangya?

A hangya egy rácson lépked és azon rajzolgat. Azt, hogy ezeket milyen szabályok szerint végzi, mi határozzuk meg. Ezt egy állapotgéppel csináljuk, amit egy egyszerű **txt** formátumban írunk meg és olvassuk be, majd a hangya ezt eltárolja. Természetesen nem kell minden felhasználónak ily módon hangyát „programozni”, mert van pár állapotgép előre bekészítve a projekt **input** mappájában.

### Hogyan van ez megjelenítve?

A projekt két négyzet alapú ablakot hoz létre. Bal oldalt a menü ablaka, tőle jobbra pedig a hangya pályájának megjelenítő ablaka. A pályán a háttér színe **fekete**, a hangya erre tud felvinni **fehér**, **kék**, **sárga** és **zöld** színeket vagy éppen letörölni azokat (a piros azért nincs a lehetőségek között, mert az a hangya színe).

### El lehet menteni a hangya munkáját?

A program tud képet is készíteni a pályáról, amit aztán **png** formátumban eltárol majd az **output** mappában.

## 2. Felhasználói útmutató

Most, hogy már van fogalmunk róla mivel van dolgunk tanuljuk meg használni is. A hangyánkat irányítani kell, pályáját be kell tudni állítani. Ebben segít nekünk az indítás után megjelenő menü.

### Parancsok adása

A menü kinézete egészen letisztult lett, innen irányítunk mindent gombok segítségével. A most következőkben az olvasható, hogy a különféle gombok mit csinálnak és milyen alapvető tudnivalók vannak még azokon kívül.

**Start/Stop/Folytat :** Ez a gomb tulajdonképpen három dolgot is csinál, attól függően, hogy éppen melyik felirat olvasható rajta. A „Start” esetben, elindítja a hangyát, „Stop”-nál megállítja azt (elaltatja), végül pedig „Folytat” esetén ismét munkára bírja a hangyát (felébreszti).

**Beolvas :** A gomb mellé tartozik egy szövegmező is (JTextField). Ide kell írni a beolvasni kívánt állapotgép fájlnevét. Nagyon fontos, hogy az állapotgép a projekt **input** mappájában legyen, csak akkor lesz beolvasva. Ezenkívül kiterjesztést nem kell írni a végére, a rendszer pontosan tudja, hogy **txt** formátumú fájlt kell keresnie. A beolvasás sikerességéről az ablak alján minden esetben kapunk rendszerüzenetet. Ha nem olvasunk be semmit, akkor a hangya a default állapotgépet veszi fel és Langton hangyája lesz.

**Beállít x2 :** Két ilyen feliratú gomb is van.  
Az egyik mellett egy spinner található, ahol egész számokat lehet beállítani 1 és 5 között. Ha megnyomjuk az emelletti „Beállít” gombot, azzal átállítjuk a hangya sebességét. Minél magasabb a szám annál gyorsabb a hangya, de az ötös érték már olyan sebességgel bír, hogy a hangya akár egy pillanat alatt kifut a pályáról (bizonyos állapotgépek azonban ezt megakadályozzák pl.: BinaryCounter).  
A második „Beállít” gomb a pálya méretét állítja. A mellette lévő legördülő listából kiválasztva a kívánt méretet, majd a gombot megnyomva egy új rácsot kapunk. Ez a beállítás a megjelentő ablak méretét nem változtatja.

**Pillanatkép :** Ha ezt a gombot megnyomjuk, akkor a rendszer készít egy **png** kiterjesztésű képet a hangya eddigi munkájáról és azt elmenti az **output** mappába.

**Töröl :** Ennek a gombnak a segítségével törölhetjük le a pályát és állíthatjuk alaphelyzetbe a hangyánkat. A hangya tulajdonságai nem változnak, viszont, ha a parancs kiadásának pillanatában dolgozott, akkor a gomb megnyomása után már nem fog, tehát újból meg kell nyomnunk a „Start” gombot, ha működésre akarjuk bírni.

**Kilép :** Ezzel a gombbal tudunk kilépni, mindkét ablakot bezárja. Emellett készít még egy utolsó képet a hangya munkájáról.

## Megjegyzések

- A hangya, ha kifut a pályáról (olyan helyre szeretne lépni a rácson, ami nem létezik), akkor a munkafolyamat merő egyszerűséggel megáll. Ekkor a „Stop/Folytat” gomb sem működik, hiszen a hangyát termináltuk, nem lehet folytatni az eddigi munkáját, mert az indexelési hibával járna. Ilyenkor rá vagyunk kényszerülve arra, hogy letöröljük a pályát a „Töröl” gombbal.
- Ha átállítjuk a pálya méretét, a dolgozó hangya nem fog megállni, csak alaphelyzetbe kerül. Vagyis az új rács közepére pozicionál és felfele néz, más tulajdonsága nem változik.
- Minden egyes kép, amit készítünk más nevet kap, így nem áll fent a veszélye, hogy egy korábban készített kép elveszik, mert felülírnánk azt.

### A jó állapotgép (haladóknak)

Fontos, hogy az állapotgépet **txt** formátumban adjuk meg és az **input** mappába rakjuk, mert onnan végzi a program a beolvasást. Ahhoz, hogy tudjuk milyen egy jó állapotgép, kicsit jobban kell értenünk a hangya működését is.

Egy hangya lépése a következő szakaszokra bontható:

1. Megnézi milyen státuszban van és mi a cella értéke, amin áll
2. Új irányba fordul (L-balra, R-jobbra, N-nem fordul, U-teljesen megfordul)
3. Az aktuális cella értékét megváltoztatja, azaz rajzol
4. Előre lép egyet
5. Végül megváltoztatja a státuszát

Így egy lehetséges bemenet például a következő: *[állapot-cellaérték-irány-újérték-újállapot]*

0-0-R-1-0

0-1-R-1-1

1-0-N-0-0

1-1-N-0-1

Ebben a példában csak két színnel dolgozik a hangya, fehér és fekete. Többel is lehetne, viszont ahány színt használunk, annyi különböző sor kell egy állapothoz.

A különböző színekhez tartozó értékek: 0 – fekete, 1 – fehér, 2 – kék, 3 – sárga, 4 – zöld

Ügyelni kell a jó formázásra. Az előbbi állapotgéppel például ekvivalens a most következő:

00R10

0-1-R-1-1

1-0-N-0-0

1-1-N-0-----1

A kis elválasztó vonalak mennyisége nem változtat semmin, mert egyszerűen ignorálva vannak beolvasáskor. Azonban a tényleges információt tartalmazó karakterek száma nem lehet sem több, sem kevesebb ötnél egy sorban. Ezt a bemeneti konstrukció megköveteli.

A fentiek alapján tehát rossz bemenet a következő:

0-----R-1---0

0--1---R-----1-1

1-0-N-0-0-0

1---1----N-----0---1

Itt az első és a harmadik sor is hibás, mert az egyikben csak négy, a másikban már hat adat van.

Az is megtörténhet, hogy a bemenetünk helyes, de mégsem adtunk meg egy jó állapotgépet. Például a hangyánk az első pár lépés után megáll, mert olyan státuszba került, amiből nem tud kilépni, vagy éppen ugyanazokat a köröket teszi folyamatosan és ezért nem fog bővülni a rajza. Ezek olyan problémák, amiket a rendszer külön nem érzékel, mert neki nem feladata az, hogy megmondja egy állapotgépről, hogy az helyes vagy sem. Ez a mi dolgunk. Az övé pedig az, hogy megpróbáljon futtatni egy hangyát egy megadott állapotgéppel.

Természetesen lehet találni az interneten is ilyen állapotgépeket. Az egyetlen probléma az, hogy általában más a szabálya a formázásnak, ezért sajnos kénytelenek vagyunk átírni ezeket a fájlokat, hogy megfeleljen a fentebb részletezett követelményeknek.

Nagyon kellemes hangya programozást!

### 3. A program osztálydiagramja (fejlesztőknek)

