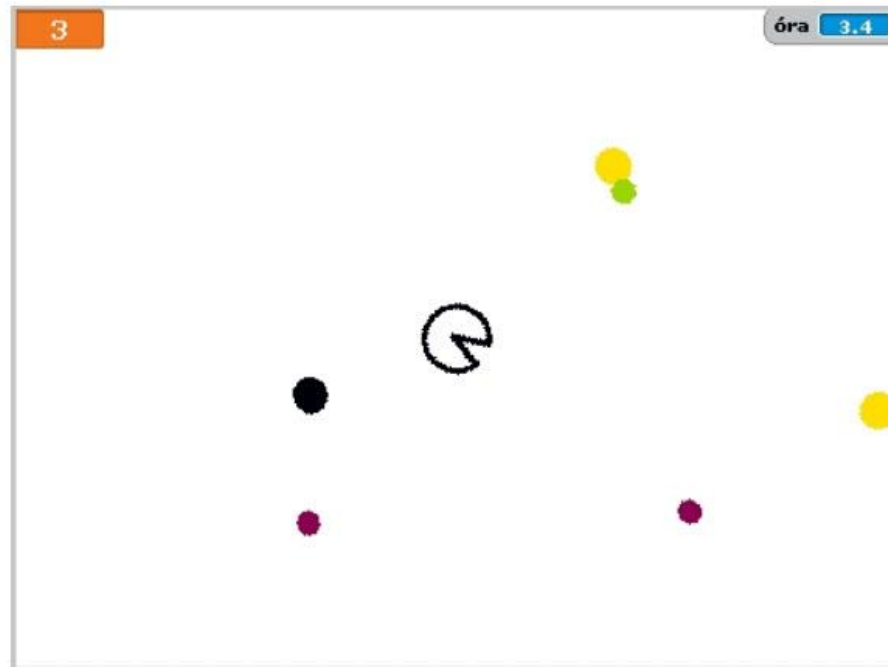


# 28. óra Scratch\_4

## 7. Változók

Az eddig megismert módszerekkel nem lehet igazán izgalmas, „tétre menő” játékokat készíteni. Ebben a leckében végre megtudhatod, hogyan lehet például pontokat szerezni és időt mérni.

A következő játékot készítjük el:



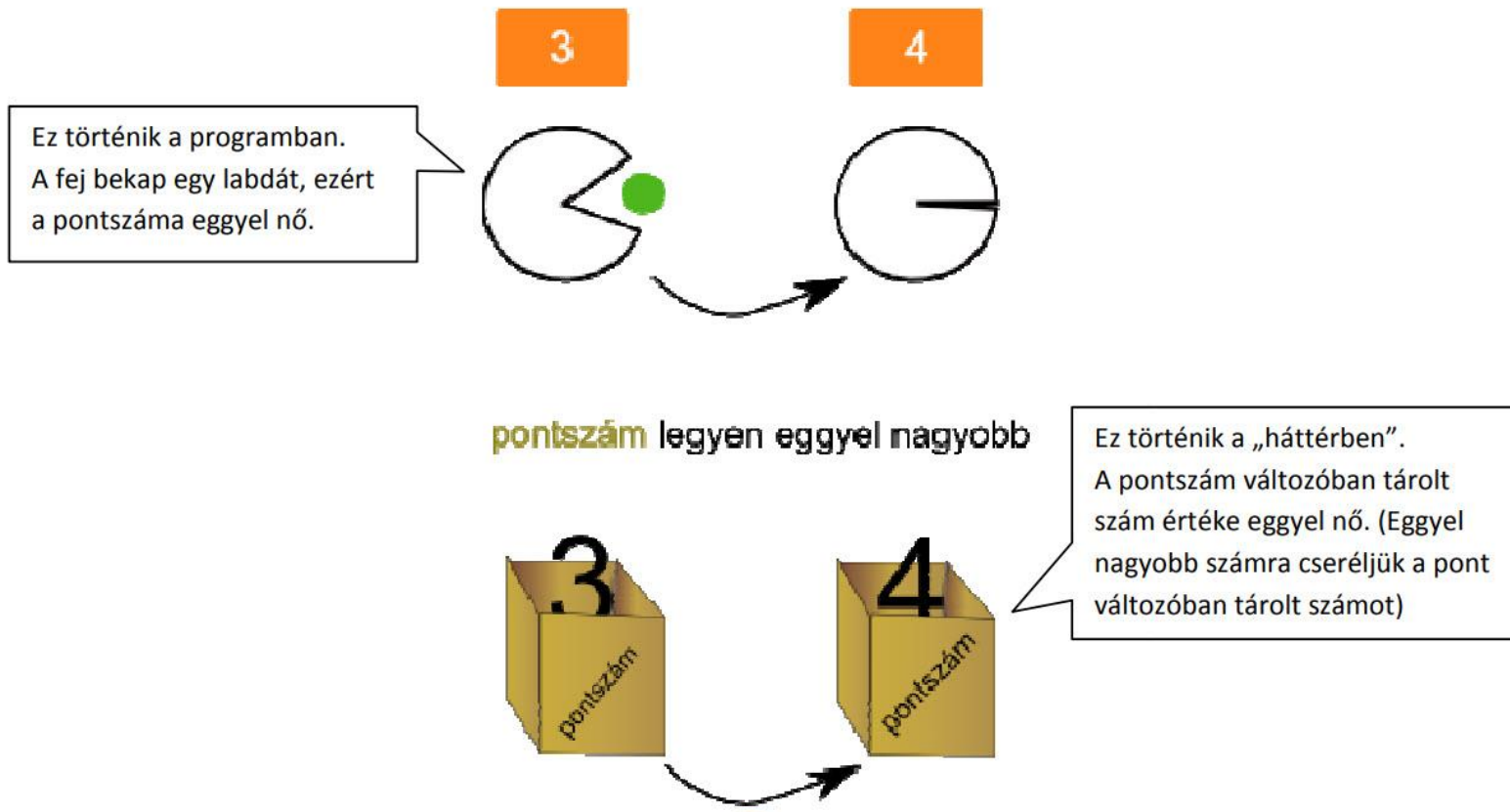
Takács Valéria szakdolgozata alapján

<http://grafit.netpositive.hu/download/scratch/szakdolgozat.pdf>

## Változó

A változó (az informatikában) legegyszerűbben megfogalmazva olyan hely a számítógépen, ahol valamilyen mennyiséget tárolunk. A változónak van neve és valamilyen értéke – ez az érték a program futása során változhat.

A változót úgy lehet elképzelni, mint egy dobozt, amelyben tárolhatunk valamilyen értéket. Amikor szükségünk van rá, akkor azt kivesszük és felhasználjuk, majd esetleg egy másik értéket teszünk vissza helyette.



## A játék elkészítése

- Szereplők: tátogó fej, színes labdák (könnyen megrajzolhatod, de le is töltheted őket)
- Változó: pontszám
- A szereplők mozgása, irányítása kissé eltér az eddigiektől. Erről a hajtás után, a véletlenszámok kapcsán olvashatsz.
- A játék lényege, hogy a fejet irányítva 30 másodperc alatt (ennek megvalósítását **Az óra** című leckében találod) minél több pontot kell szerezni a véletlenszerűen mozgó színes labdák elkapásával. A fekete labda érintése pontlevonással jár.

## Pontszámolás

- A pontszám a játék indulásakor 0. Eggyel nő, ha a fej elkap egy színes labdát és eggyel csökken, ha hozzáér a feketéhez. Tehát a pontszám egy változó. Létrehozása: **Változó létrehozása**

A változó minden szereplőhöz tartozik, azaz bárki fel tudja használni, vagy meg tudja változtatni.

Változó neve?

pontszám

Mindenkié  A kiválasztott szereplőé

OK Mégse

Ha a változó csak egy szereplőhöz tartozik, akkor csak ő tud vele dolgozni, a többiek „nem látják”, nem tudják használni.

- Ekkor megjelenik a pontszám változó és a vele kapcsolatos parancsok a parancskészlet változók csoportjában, a játéktérben pedig a változó „kijelzője”. Ez háromféle lehet, a jobb egérgombbal kattintva előjövő helyi menüből választható ki, hogy melyik jelenjen meg a játéktérben. (ha nem szeretnéd, hogy a játéktérben megjelenjen a változó, tüntesd el a pipát a változó neve előtt a parancskészletben):



- Első lépésként állítsuk be, hogy a pontszám kezdetben 0 legyen. Ezt a feladatot adjuk pl. a játéktérnek (de mindegy, hogy melyik szereplő kapja ezt a feladatot).

A legördülő menüből választható ki, hogy melyik változóra vonatkozik a parancs.



Itt adható meg, hogy mennyi legyen a változó értéke. Kezdetben 0.



- Ha valamelyik színes golyó hozzáér a fejhez, akkor a pontszám nőjön eggyel:



Ha hozzáér a fejhez, akkor eltűnik (mert a fej megeszi) és a pontszám +1-gyel változik (vagyis nő).

- A fekete golyó nem tűnik el érintkezéskor, így máshogyan kell megoldani a pontlevonást, különben több pontot is levonna, amíg áll. Ezért is van szükség a várakozásra, hogy addig a fej el tudjon menni a közeléből. Az érintkezés akár 1 másodpercig is eltarthat, ezalatt az idő alatt a program fut tovább – mindig érintkezést érzékel, így többször is levon egy pontot (pedig látszólag csak egyszer találkozik a két szereplő).


A színes golyóknál ez azért nem probléma, mert az érintkezéskor azonnal eltűnnek és a láthatatlan szereplőkre nem működik az érintkezésvizsgálat. Tehát a fekete golyó feladatai:



## Véletlenszámok

A játékban a labdák véletlenszerűen mozognak, a fej pedig egy kicsit „gyorsabb”, mint az eddigi főszereplőink. Most megtudhatod, hogyan lehet ezt megoldani. A véletlenszámok nagy segítséget jelenthetnek a játékok elkészítésében. Segítségükkel megvalósítható például az „ellenségek” kiszámíthatatlan mozgása, amitől érdekesebbé válik a játék. A Scratch-ben két szám közötti véletlenszámot adhatunk meg. **véletlen 1 és 10 között**

Ez a parancs ebben az esetben egy 1 és 10 közötti számot ad és mindegyiket egyforma eséllyel, ugyanúgy, mintha egy 10 oldalú dobókockával dobnánk. Persze az 1 és 10 helyett más számokat (vagy akár változókat) is megadhatsz. Ha mindkét szám (a tartomány végpontjai, amelyből véletlenül választ a program számot) egész, akkor eredményül is egész számot kapsz. Ha valamelyik végpont nem egész szám, akkor a kapott szám sem lesz az.



## A mozgások megvalósítása

- lássuk először a golyókat:

Az elfordulás is véletlenszerű. Ennek köszönhető, hogy nem mindig ugyanazt a pályát járja be a szereplő.

```

kattintásakor
mindig
  menj véletlen 10 és 15 között lépést
  ha szélen vagy, pattanj vissza
  fordulj véletlen 1 és 10 között fokot

```

A játék kezdetétől a golyó folyamatosan mozog, de nem azonos sebességgel. Véletlenszerű a lépéshossza. Ha növelnénk a tartományt, akkor még kiszámíthatatlanabb lenne a mozgás, de ekkora játéktérben az már zavaró lehet.

- Ha ezek után lemásolod a szereplőt (hogy több golyó legyen a pályán), akkor minden golyó hasonló irányban fog mozogni, mivel ugyanarról a helyről indulnak és nem túl nagy tartományban mozog az elfordulásuk és a lépéshosszuk. Ez könnyen kiküszöbölhető, ha a golyók kezdőpozíciója is véletlen:

```

kattintásakor
ugorj x: véletlen -240 és 240 között y: véletlen -180 és 180 között

```



- A fej irányításához – az eddigiekkel ellentétben – billentyűérzékelést használunk. Nézzük, miben különbözik ez a „sapkás” megoldástól.



**BILLENTYŰ-LENYOMÁS:** Ekkor ha folytonosan lenyomva tartod a szóköz billentyűt, akkor nem fog folyamatosan végrehajtódni a hozzá rendelt esemény. Mindig vár egy picit, mielőtt megismétli (kivárja a billentyűismétlési holtidőt). Ráadásul több billentyű együttes lenyomása csak a legutolsót teszi érvényessé.



**BILLENTYŰÉRZÉKELÉS:** Itt nincs várakozás, folyamatosan lenyomott gomb mellett szünet nélkül zajlik az esemény ismétlése. Az így megvalósított irányítás kiválóan alkalmazható például autóversenyes játékokban.

- Tehát a fej irányítása:

