



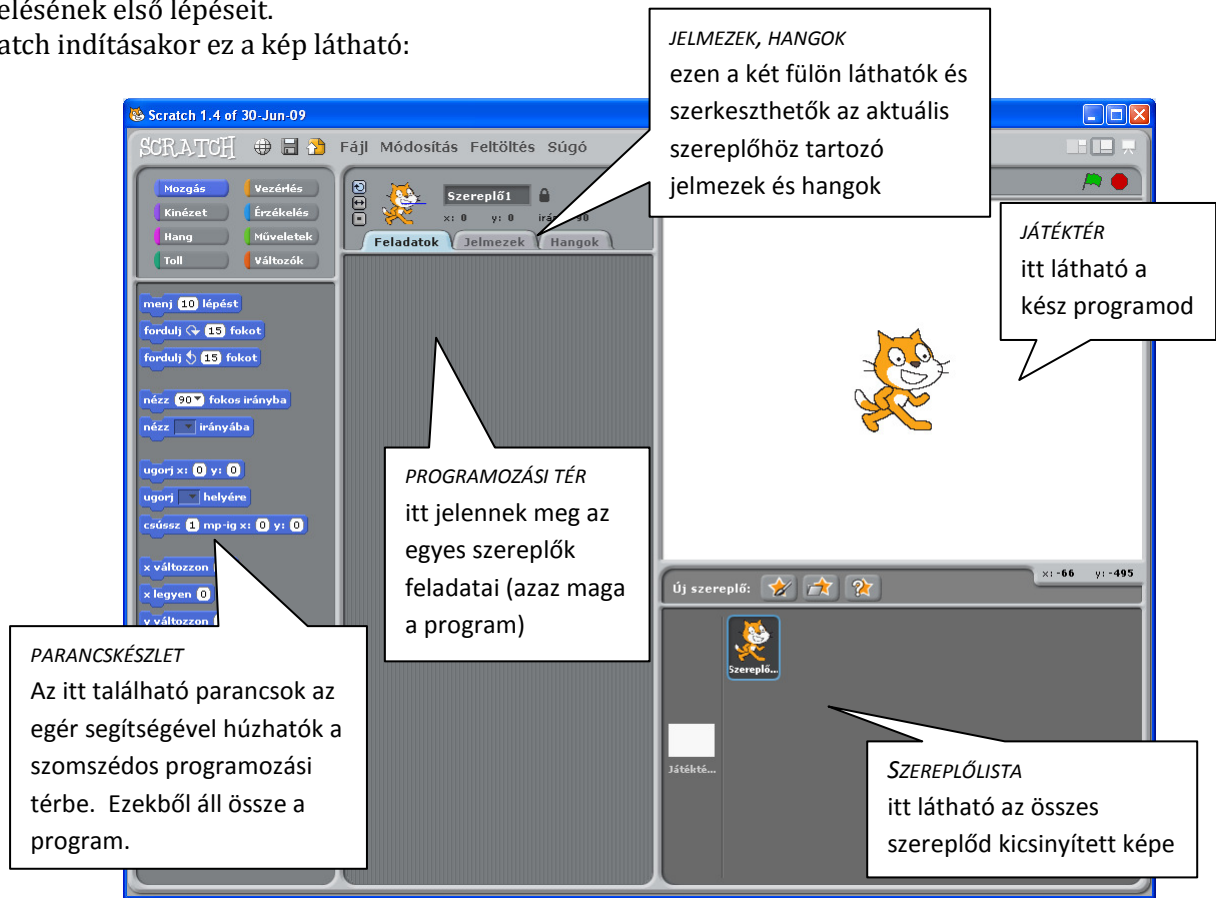
Scratch Magyarország Portál

Takács Valéria
2009.

1. Ismerkedés a Scratch környezettel

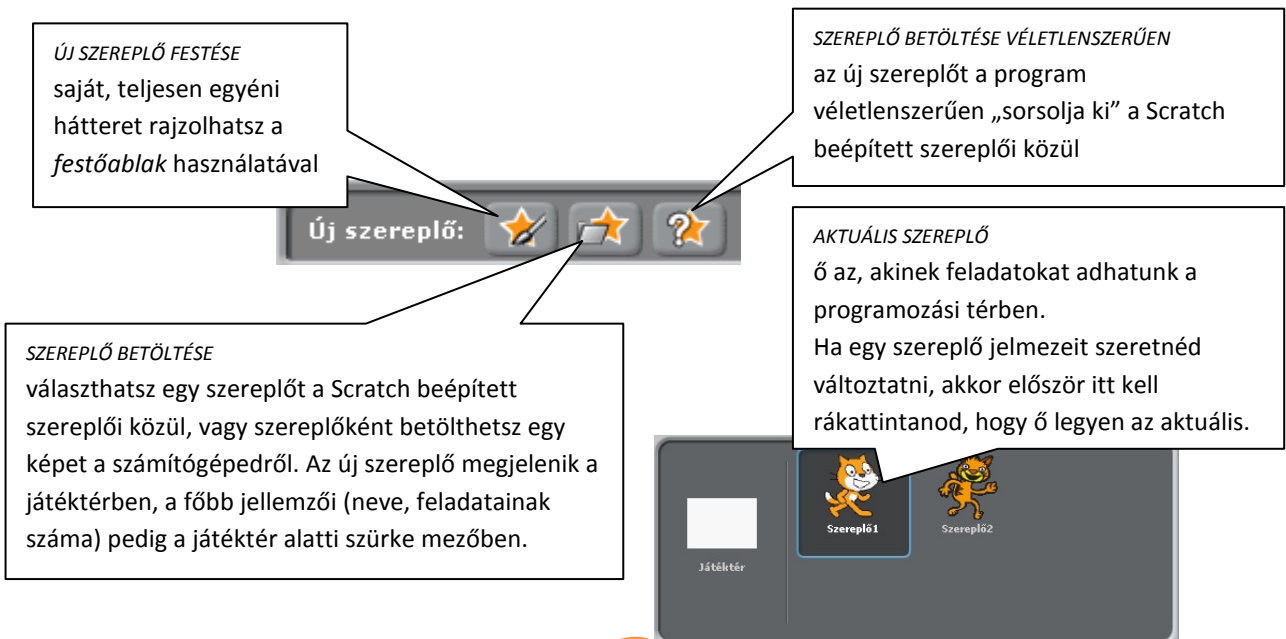
A Scratch mesék és játékok készítésére alkalmas programozási környezet. A következőkben megtanulhatod a kezelésének első lépéseit.

A Scratch indításakor ez a kép látható:



A Szereplők

A Scratchben a játékok, animációk szinte elképzelhetetlenek szereplők nélkül, ezért fontos megismerni a kezelésüket. Kezdetben egyetlen szereplő van a játéktérben, a képen látható macska. Ezen könnyen változtathatsz: néhány kattintással új szereplőket tölthetsz be vagy módosíthatod a már meglévőket. Ehhez a játéktér alatti, illetve a programozási tér jelmezek fülén lévő gombokat használhatod.



JELMEZEK

Egy szereplő jelmezei hasonlóak, mint egy film képkockái. Általában a szereplő „pillanatfelvételei” egy bizonyos tevékenység elvégzése közben (pl. a mozgásának fázisai). A jelmezek nagyon fontosak, gyakran tőlük válik „élővé” a szereplő.

FESTÉS

saját, teljesen egyéni jelmezt rajzolhatsz a szereplőnek a *festőablak* használatával

AKTUÁLIS JELMEZ

A jelmezek közül a késsel keretezett. Ilyen alakban jelenik meg a szereplő a játéktérben

MÓDOSÍTÁS
a *festőablak* használatával módosíthatod a jelmezt

SZEREPLŐ NEVE

az új szereplők automatikusan a *szereplőx* elnevezést kapják. Ez több szereplő esetén kavarodást okozhat, ezért a szereplőket **mindig nevezd el!**

BETÖLTÉS

betölthetsz egy jelmezt a Scratch beépített jelmezei közül, vagy egy képet a számítógépedről. Az új jelmez az utolsó lesz a jelmezek sorában.

TÖRLÉS

a jelmezt törölheted ezzel a gombbal, de csak ha van még rajta kívül másik a sorban

MÁSOLÁS
valójában a jelmez kettőzését jelenti, megjelenik belőle még egy példány a jelmezek sorában. Akkor érdemes alkalmazni, ha két jelmez csak kevésben különbözik egymástól, így az új jelmezt a másolatból könnyen elkészítheted

Új háttér betöltése

Ahhoz, hogy egy animáció vagy egy játék szép és élvezetes legyen, fontos a megfelelő háttér. Ez általában nem fehér, szóval változtasd meg! Mivel a háttér valójában a Scratch egy speciális szereplője, ezért a kinézetének módosítása hasonlít a szereplők változtatásához, tehát nem lesz nehéz dolgod:

- Kattints a játéktér alatti mezőben a fehér téglalapra (ez most a beállított háttér)
- a programozási környezetben válaszd ki a hátterek fület
- itt megjelennek a szereplőknél megismert lehetőségek:

FESTÉS
saját, teljesen egyéni hátteret rajzolhatsz a *festőablak* használatával

BETÖLTÉS
betölthetsz egy már létező képet a számítógépedről (vagy a Scratch hátterei közül). Az új kép az utolsó lesz a hátterek sorában

TÖRLÉS
a hátteret törölheted ezzel a gombbal, de csak ha van még rajta kívül másik a sorban

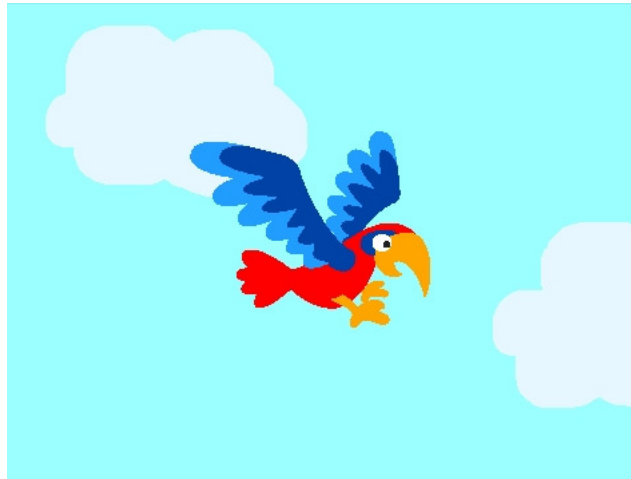
MÓDOSÍTÁS
a *festőablak* használatával módosíthatod az aktuális hátteret

MÁSOLÁS
valójában a háttér kettőzését jelenti, megjelenik belőle még egy példány a hátterek sorában

2. Megmozdulnak a szereplők

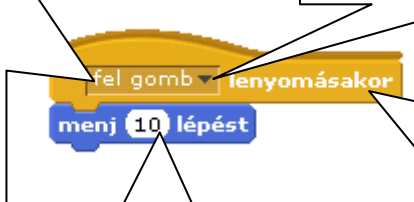
Egy játék akkor szórakoztató, ha irányíthatjuk a szereplőit (vagy legalább a főszereplőt). Ebből a leckéből megtudhatod, hogyan lehet ezt megtenni.

Egy egyszerű játékot fogunk elkészíteni, amelyben egy papagájt lehet röptetni az égen:





A játék elkészítése

- Először töröld ki a macskát a játéktérből és töltsd be az Animals könyvtárból a parrot1-a nevű papagájt. A nevét változtasd Papagájra. A háttér legyen kék ég felhőkkel (ezt letöltheted a honlapról vagy megrajzolhatod a festőablakban.)
- Most a papagájnak adunk feladatokat. Ehhez húzd a parancsokat az egérrel a parancskészletből a programozási térbe. A parancsokból akkor lesz feladat, ha megfelelő sorrendben összeillesztjük őket egy sapka alatt.
- Először oldjuk meg azt, hogy a fel nyíl lenyomásakor a papagáj haladjon egy kicsit előre. Ehhez a következő parancsokra lesz szükség:



A programot azonnal kipróbálhatod. A fel billentyű lenyomásakor a papagáj a játéktérben 10 lépést (10 képpontnyit) halad előre. Azonban nem áll meg a szélénél, kirepül a képből. Ennek elkerülésére illeszd a **ha szélén vagy, pattanj vissza** parancsot az eddigi kettő közé. Hatására a fel gomb lenyomásakor ha a papagáj éppen a játéktér szélén van, akkor megváltoztatja az irányát és csak azután halad tovább.

A sapka alapbeállításban a szóköz billentyű lenyomását figyeli:

A fekete nyílra kattintva legördülő listából választható másik billentyű.

A parancsok színe megegyezik annak a parancscsoportnak a színével, ahol megtalálható. Tehát ez a parancs a narancssárga **vezérlés** csoportba tartozik.


Ha szeretnéd, hogy a madarad gyorsabban repüljön, írd át a „sebességét” 20 lépésre!
Ha ide negatív számot írsz, akkor a szereplő hátrafelé fog haladni.

- Ahhoz, hogy a papagáj haladási irányát is változtatni tudjuk, még két egyszerű feladatot kell adnunk az eddigihez:



- Most már csak az a baj, hogy a papagáj nem mozgatja a szárnyait repülés közben. Ezen könnyen lehet segíteni az előző leckében említett jelmezek ügyes váltogatásával. Töltsd be a Papagáj jelmezei közé az Animals könyvtárból a parrot1-b nevűt. Ha egymás után kattintasz a jelmezekre, a játéktérben úgy látszik, mintha a papagáj csapkodna a szárnyaival. A jelmezeket nem csak így lehet váltogatni.

Ha minden egyes feladathoz hozzáilleszted a **válts jelmezt** parancsot, akkor a szereplő a gombnyomások hatására elfordul vagy repül egy kicsit és még a sorban következő jelmezt is magára ölti. (Mivel most csak két jelmez van, ezért e kettő fog váltakozni.)

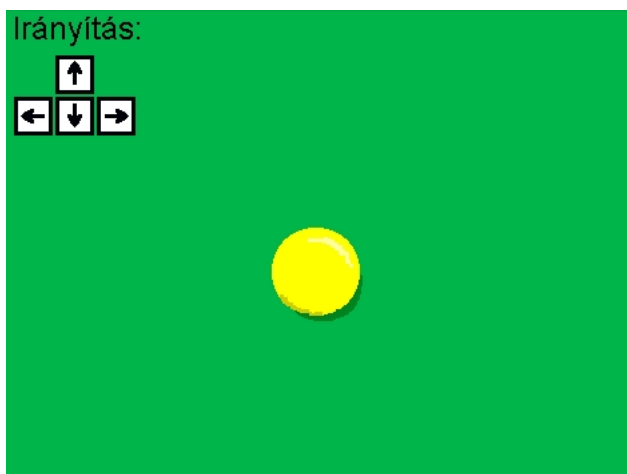
A papagáj feladatai tehát:




Mentsd el a projektedet papagaj néven!

Variáció

Hogyan készülhetett ez a program? Próbáld meg te is elkészíteni!



Ha elakadtál, töltsd le a programot a honlapról vagy olvasd el a hozzá fűzött megjegyzéseket itt:



The image shows a sequence of Scratch code blocks for a ball movement program. The blocks are as follows:

- fel gomb lenyomásakor** (when the up button is pressed):
 - fordulj 90 fokot (turn 90 degrees)
 - menj 10 lépést (move 10 steps)
 - fordulj 90 fokot (turn 90 degrees)
- le gomb lenyomásakor** (when the down button is pressed):
 - fordulj 90 fokot (turn 90 degrees)
 - menj 10 lépést (move 10 steps)
 - fordulj 90 fokot (turn 90 degrees)
- jobbra gomb lenyomásakor** (when the right button is pressed):
 - menj 10 lépést (move 10 steps)
- balra gomb lenyomásakor** (when the left button is pressed):
 - menj -10 lépést (move -10 steps)

Two yellow callout boxes provide additional information:

- The first callout box, next to the first block, explains: "Azért fordítjuk el a golyót először jobbra, majd vissza balra, hogy az eredeti iránya ne változzon a feladat elvégzése után. A golyó eredeti irányát a fenti kék vonalka jelöli." (We turn the ball right first, then left back, so its original direction doesn't change after the task is completed. The ball's original direction is indicated by the blue line above).
- The second callout box, next to the last block, explains: "-10 lépést is lehet menni. Ez azt jelenti, hogy a golyó nem előre, hanem átra gurul." (You can also move -10 steps. This means the ball doesn't move forward, but backward).

3. Ismétlődések

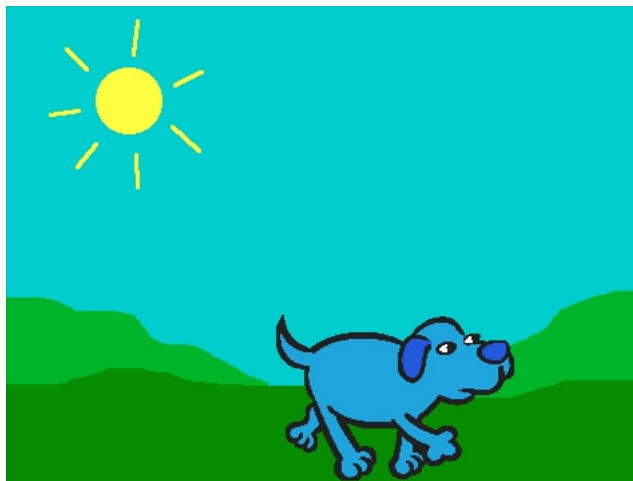
Az előző leckében a szereplő akkor tett meg valamit, ha leütöttünk egy billentyűt. Most megmutatjuk, hogy szereplőink beavatkozás nélkül is tudnak feladatokat elvégezni.

Ciklusok


A ciklusok ismétlődő tevékenységek megvalósítására szolgálnak. Ebben a leckében két ciklusfajtaival ismerkedhetsz meg:

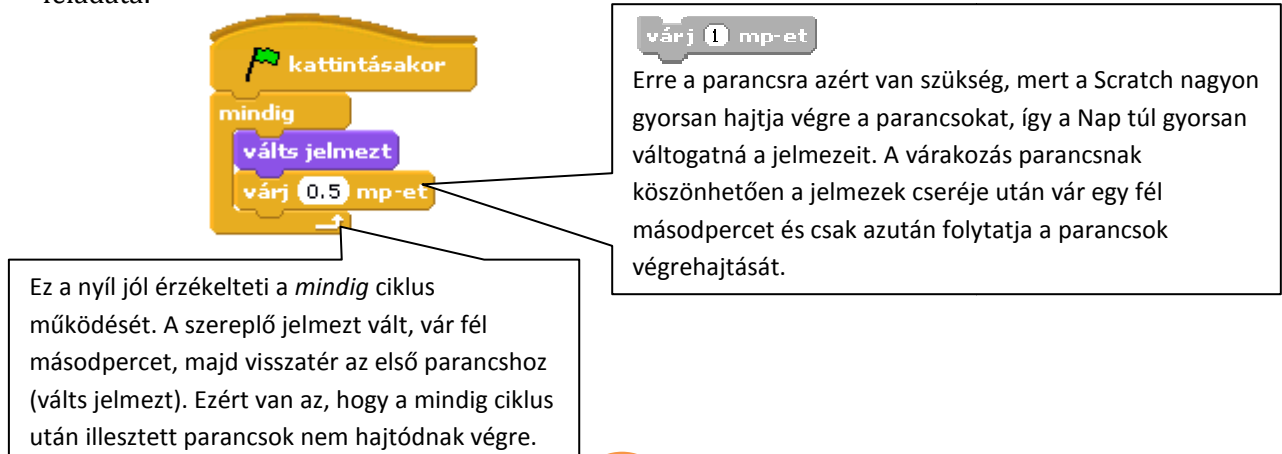


Most lássunk egy példát is a parancsok használatára. Egy olyan programot készítünk, amelyben egy napsütéses délutánon egy kutya végigsétál a képernyőn:



A program elkészítése

- Szereplők: Nap (a jelmezeit letöltheted a honlapról), Kutya (jelmezei: Animals/dog2-b, Animals/dog2-c)
- A parancsok végrehajtása a zöld zászlóra kattintáskor indul, tehát a  sapkához kapcsoljuk őket.
- A Nap a program indításától kezdve folyamatosan – mindig – változtatja a jelmezeit, tehát az ő feladata:



- A Kutya nem ismétli végtelen sokszor a feladatát: elindul, megy egy darabig – *valahányszor* A Kutya nem ismétli végtelen sokszor a feladatát: elindul, megy egy darabig – *valahányszor* megismétli a **menj 10 lépést** parancsot – majd megáll. Az ő feladata:

A szereplő most 20-szor megismétli a cikluson belüli parancsokat, majd megáll (mivel más feladatot nem adtunk neki).

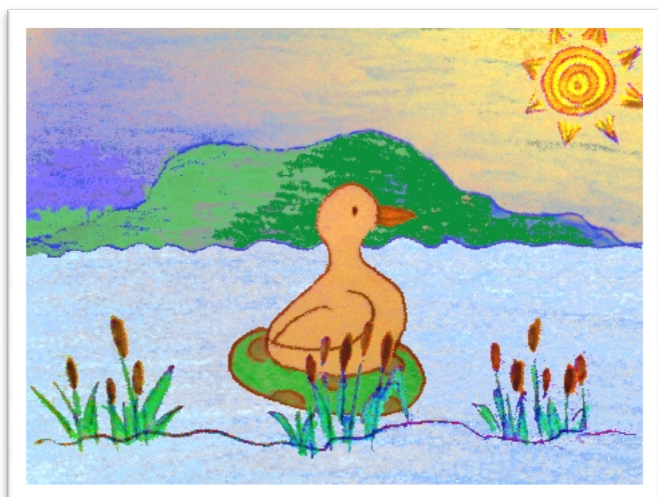


Erre a parancsra azért van szükség, mert a zöld zászló lenyomásakor a kutya mindig onnan indul, ahol az előző futtatáskor befejezte a sétát, tehát előfordulhat, hogy „kimegy a képből”. Ahhoz, hogy a kutya visszapattanáskor ne álljon fejre, a forgási stílusát át kell állítani erre:

Mentsd el a projektedet kutya néven!

Feladat

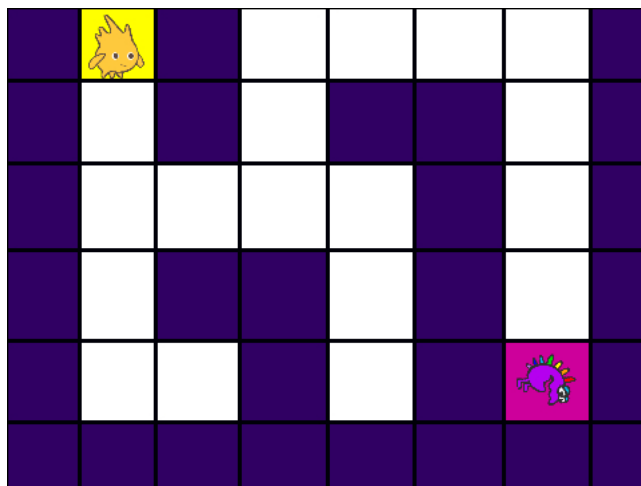
Készíts animált képeslapot az alábbi példák alapján:



4. Feltételek

A játékok jelentős részében a szereplők nem ugyanúgy viselkednek minden helyzetben. Ennek megvalósítására feltételekhez kötjük az egyes parancsok végrehajtását.

Ezt az egyszerű labirintusos játékot fogjuk elkészíteni, melynek célja, hogy a Gobo nevű szereplőt a sárga négyzetről a magenta színűre juttassuk anélkül, hogy a falhoz érjünk. A Lila szereplő ekkor örömmel kiáltson fel, hogy barátja megtalálta:



A játék elkészítése

- A háttér letölthető a honlapról. Szereplők: Fantasy/gobo1 (mérete az eredeti méret 40%-a), Fantasy/fantasi11 (mérete az eredeti 35%-a).
- Gobo mozgatása tetszőleges (pl. a második leckében megismert is lehet). Ezúttal egy koordináta-rendszerre támaszkodót használunk, mivel a szereplő csak jobbra-balra, illetve fel-le mozog és az ilyen típusú mozgás az x és y koordináták változtatásával jól megvalósítható:



- Fontos, hogy a játék kezdetén Gobo a start (sárga) négyzeten álljon. Ehhez a következő feladatot kell adnunk neki:

UGORJ
ez a parancs a játéktér megadott koordinátájú pontjába helyezi át a szereplőt

A kívánt koordináták könnyen „megszerezhetőek”: helyezd a szereplőt a sárga mezőre. Ha ezután kiválasztod a mozgás csoportot, akkor az *ugorj* parancs koordinátái a szereplő aktuális helyét mutatják (azaz a most szükséges „startpozíciót”).
Egyszerűbb megoldás, ha leolvasod az egérmutató koordinátáit a játéktér alatti sávból akkor, amikor az a sárga négyzet fölött áll.

- A játék lényege, hogy a szereplőnek úgy kell végigmennie a labirintuson, hogy nem érhet hozzá a falhoz. Azonban Gobo ezt most gond nélkül megteszi. Ennek megakadályozásához arra lenne szükség, hogy ha falhoz ér, akkor „kapjon büntetést”, pl. kezdje újra a játékot.

Ezt a  parancssal valósíthatod meg:

HA
ha teljesül a megadott feltétel, akkor végrehajtódik a benne megadott parancs (csak akkor)

```
fel gomb lenyomásakor
y változzon 5
ha érintesz színt?
ugorj x: -159 y: 152
```

érintesz színt?
Az érzékelés csoportba tartozó parancs. Igaz értéket ad, ha a szereplő érinti a megadott színt a játéktérben (szín megadásához kattints a színmintán, majd válassz egy színt a palettáról vagy a játéktérből).

Ebben az esetben tehát a szereplő mindenképpen lép egyet és csak azután ellenőrzi, hogy érinti-e a falat.

A fel gomb lenyomásakor a szereplő 5 lépést megy felfelé. Ez után a program megvizsgálja, hogy érint-e sötétlila színt (ez most a labirintus falát jelenti). Ha érint, akkor visszaugrik a sárga mezőre (amelynek megadtuk a helyét az ugorj parancsban).

- A szereplő feladatai tehát:

```
kattintásakor
ugorj x: -159 y: 152
```

```
fel gomb lenyomásakor
y változzon 5
ha érintesz színt?
ugorj x: -159 y: 152
```

```
bábra gomb lenyomásakor
x változzon -5
ha érintesz színt?
ugorj x: -159 y: 152
```

```
le gomb lenyomásakor
y változzon -5
ha érintesz színt?
ugorj x: -159 y: 152
```

```
jobbra gomb lenyomásakor
x változzon 5
ha érintesz színt?
ugorj x: -159 y: 152
```

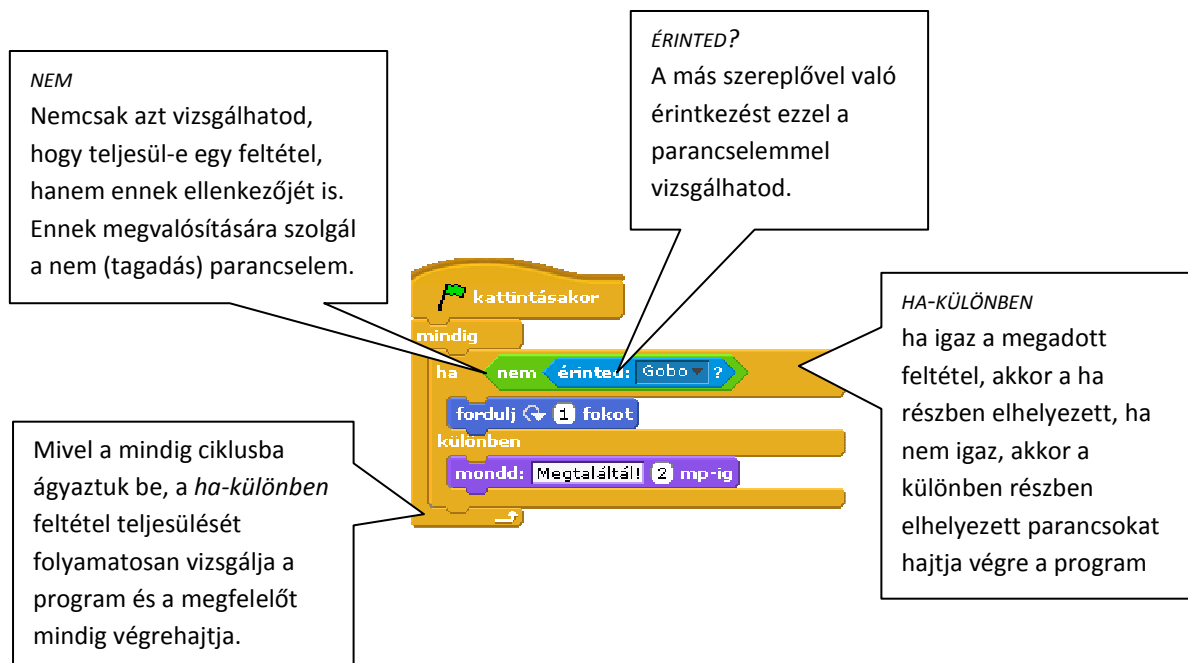
A következő oldalon megismerheted a labirintus végén várakozó másik szereplő feladatait.

Egymásba ágyazás

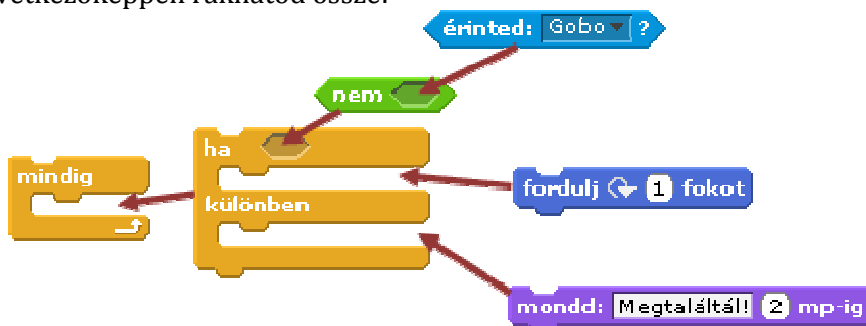
A lila szereplő ha nem érintkezik Goboval, akkor folyamatosan egy helyben forog. Különben (a Goboval való érintkezéskor) azt mondja, hogy "Megtaláltál!".

Ennek megvalósításához több parancs egymásba ágyazására van szükség. Ilyenl már találkoztál a ciklusoknál is, amikor a ciklus belsejébe illesztetted a parancsokat. Ennek a leckének az első felében pedig a *ha* parancs feltételét kellett megadnod beillesztés segítségével.

A szereplő feladatai:



Ezt a feladatot a következőképpen rakhatod össze:



A *ha...különben* parancs két ágába helyezett parancsok felcserélhetők, ha a feltételt tagadod. Így a két feladat hatására ugyanazt csinálja a szereplő:



Mentsd el a projektet labirintus néven!

5. Üzenetek

Többszereplős játékok esetén elengedhetetlen, hogy a szereplők valamilyen módon kommunikáljanak egymással. Erre valók a Scratchben az üzenetek.

Először megismerheted a használatukat egy nagyon egyszerű programon keresztül, majd megmutatjuk, hogy hogyan teheted a segítségükkel teljesebbé a már elkészült játékaidat.



Ismerkedés az üzenetekkel

- A játék szereplői: Anna (People/girl4-sitting), Bea (People/girl5) és a Fiú (boy4 laughing). A háttér letölthető a honlapról.
- A játék igen egyszerű: ha Annára kattintunk, akkor a Fiú Annára néz és közelebb lép hozzá, ha Beára kattintunk, akkor a Fiú felé néz és hozzá lép közelebb. Ezt üzenetek használatával valósíthatjuk meg, ugyanis szereplőink üzenetek küldésével érhetik el, hogy egy másik szereplő csináljon valamit.
- A lányoknak csak egy feladatuk van: ha rájuk kattintunk, küldjenek üzenetet a fiúnak, hogy nézzen rájuk. Ez Anna esetében:

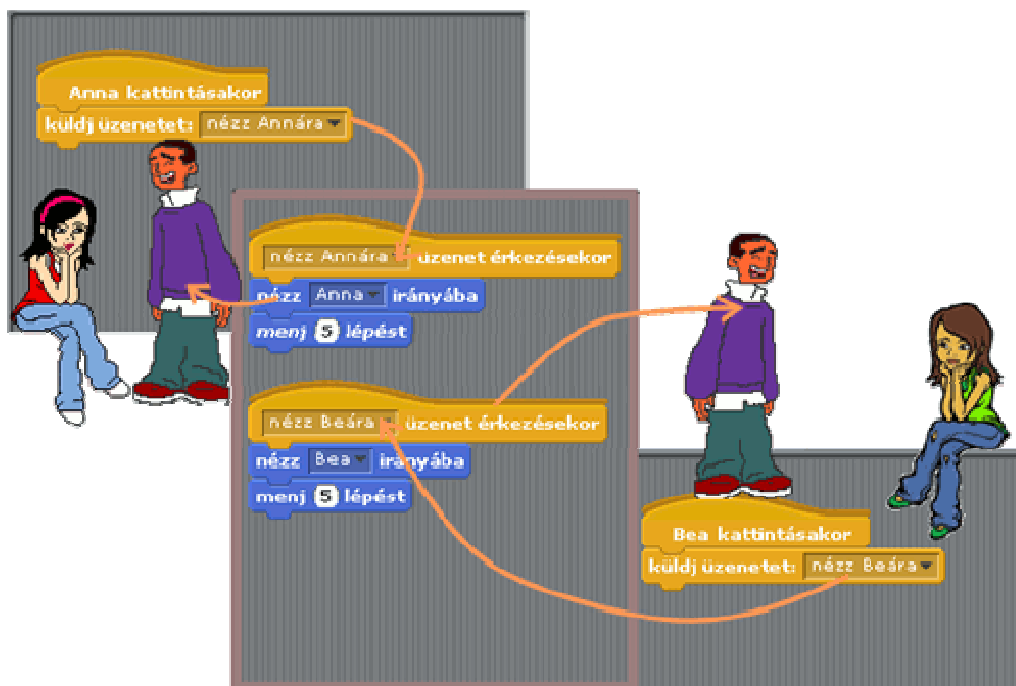


ÚJ ÜZENETET úgy hozhatsz létre, hogy a fekete háromszögre kattintva a legördülő menüből kiválasztod az Új... lehetőséget, majd beírod az üzenet nevét. (Itt választhatsz a már létező üzenetek közül is)

- A Fiú pedig a „nézz Annára” üzenet érkezésekor (ami pontosan egy időben történik az üzenet küldésével) Annára néz és közelebb lép hozzá. (A Fiú forgási stílusát persze meg kell változtatni ahhoz, hogy ne álljon fejre akkor, amikor egyik lányról a másikra néz.)



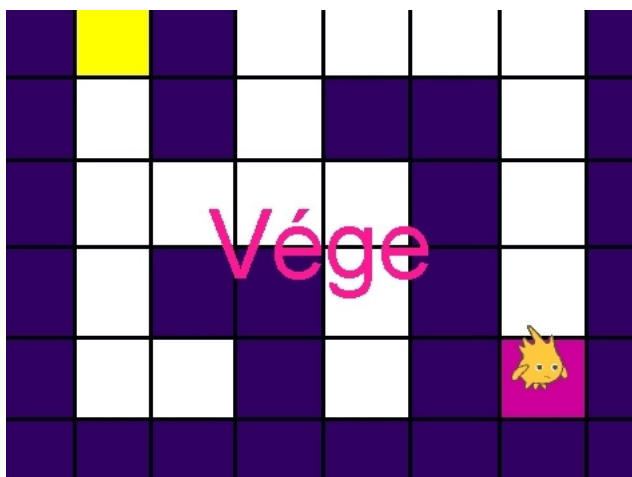
- Bea esetén ugyanezt kell tennünk. Tehát a lejátszó üzenetküldés a következőképpen zajlik:



Mentsd el a projektedet sulis néven!

Vége a játéknak

Üzenetek használatával „továbbfejlesztheted” eddigi programjaidat. Például a 4. lecke labirintusos játékát:



- A játék alapja nem változott. Az egyetlen újdonság, hogy amikor Gobo eléri a cél mezőt, akkor véget ér a játék: megjelenik a Vége felirat.
- Két új szereplő van: a cél mező és a Vége felirat (letölthetők a honlapról, de könnyen el is készíthetők).

- A cél mező feladatai:



VÁRJ EDDIG: [FELTÉTEL]

A megadott feltétel teljesüléséig vár, majd elvégzi az utána következő feladatokat. Ebben az esetben addig vár, amíg Gobo hozzá nem ér, majd vége üzenetet küld.

- A felirat feladatai:

TŰNJ EL
„láthatatlanná” teszi a szereplőt. Most azért használjuk, mert a játék kezdetén és a játék alatt nincs szükség a Vége felírra, akkor nem kell látszania.



KERÜLJ LEGELŐRE

a szereplők a létrehozásuk sorrendjében egymáson elhelyezkedő rétegekre kerülnek, ezért el tudják takarni egymást. Ez a parancs „legfelülre” helyezi a szereplőt, ő eltakarja a többiekét, de senki nem takarja őt.

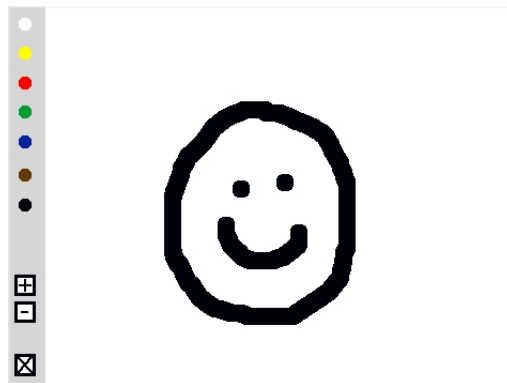
A felirat tehát a játék kezdetén láthatatlanná válik és csak akkor jelenik meg újra, ha a cél mező elküldi neki a vége üzenetet.

JELENJ MEG

a tűnj el parancs fordítottja. Láthatóvá válik az addig láthatatlan szereplő.

6. Rajzoljunk

A Scratch nem csak játékok és animációk készítésére alkalmas. Ebben a leckében egy egyszerű rajzóprogramot állítunk össze.



A program elkészítése

- A program lehetőségei: szín változtatása, tollméret változtatása, rajzok törlése és természetesen rajzok készítése.
- Szereplők: színes pontok (színváltáshoz), + és – gombok (méret változtatásához), x gomb (rajzok törléséhez), tollhegy (rajzoláshoz). A tollhegy itt egy kicsi fekete pont, de természetesen más jelmeze is lehetne (pl. egy ceruza).
- A program elkészítéséhez a **parancskészlet** Toll csoportját fogjuk használni.
- Lássuk először a bal oldali sáv szereplőinek feladatait. Ők többnyire csak üzenetet küldenek a tollhegynek, hogy min változtasson (a színekből csak egyet mutatunk be példaképp):



TÖRÖLD A RAJZOKAT
Az összes (szereplők által rajzolt) tollrajzot és lenyomatot törli a játékterről.

- A tollhegy a megadott színnel és tollvastagsággal rajzol, ha lenyomva tartjuk az egér bal gombját, és nem rajzol, ha a gomb nincs lenyomva:

A program indításától kezdve folyamatosan ellenőrizzük, hogy le van-e nyomva az egérgomb. Ha igen, akkor a tollhegy „hozzátapad” az egérkurzorhoz és a toll vonalat húz. Ha nem, akkor felemeli a tollat és nem rajzol.

Próbáld ki, mi történik, ha ezt a két parancsot fordított sorrendben illeszted ide! Melyik a jobb megoldás?

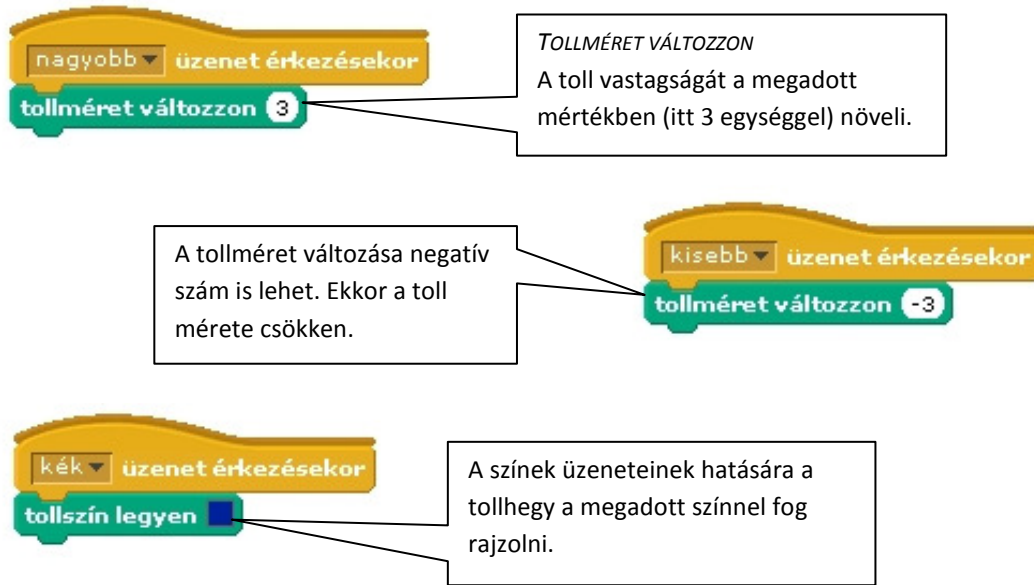


Itt vizsgáljuk, hogy az egér gombja le van-e nyomva. Csak akkor kell rajzolnia, ha igen, tehát csak akkor teszi le a tollat a szereplő.

TOLLAT LE
A szereplő rajolni fog, miközben mozog.

TOLLAT EMELD FEL
A szereplő felemeli a tollát, tehát ezután nem rajzol mozgás közben.

- Reagálás a többi szereplőtől kapott üzenetre:

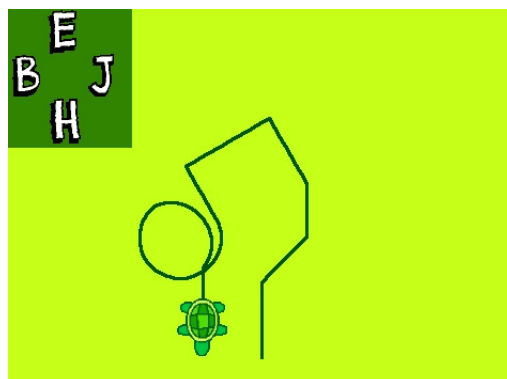


- Már csak az lehet probléma, hogy a szürke sávba is tudunk rajzolni. Ennek két megoldása van: az egyik, hogy a szürke sávot nem a háttérre rajzoljuk, hanem szereplőként hozzuk létre. Mivel a toll a háttérre rajzol, ez a szereplő azt el fogja takarni.
- A másik megoldás egy feltétel beillesztése: a tollhegy csak akkor rajzoljon, ha az x-koordinátája nagyobb, mint a szürke sáv jobb szélének ezen adata:



Mentsd el a projektedet rajztabla néven!

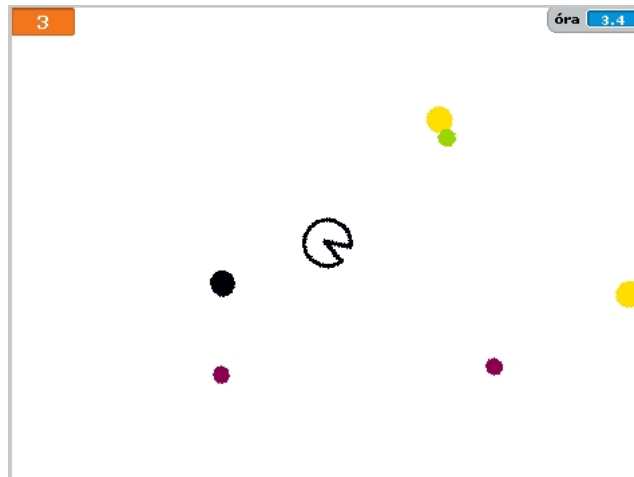
Készíts más típusú rajzolóprogramokat! Ismered az Imagine-t? Próbáld hozzá hasonló programot készíteni!



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.

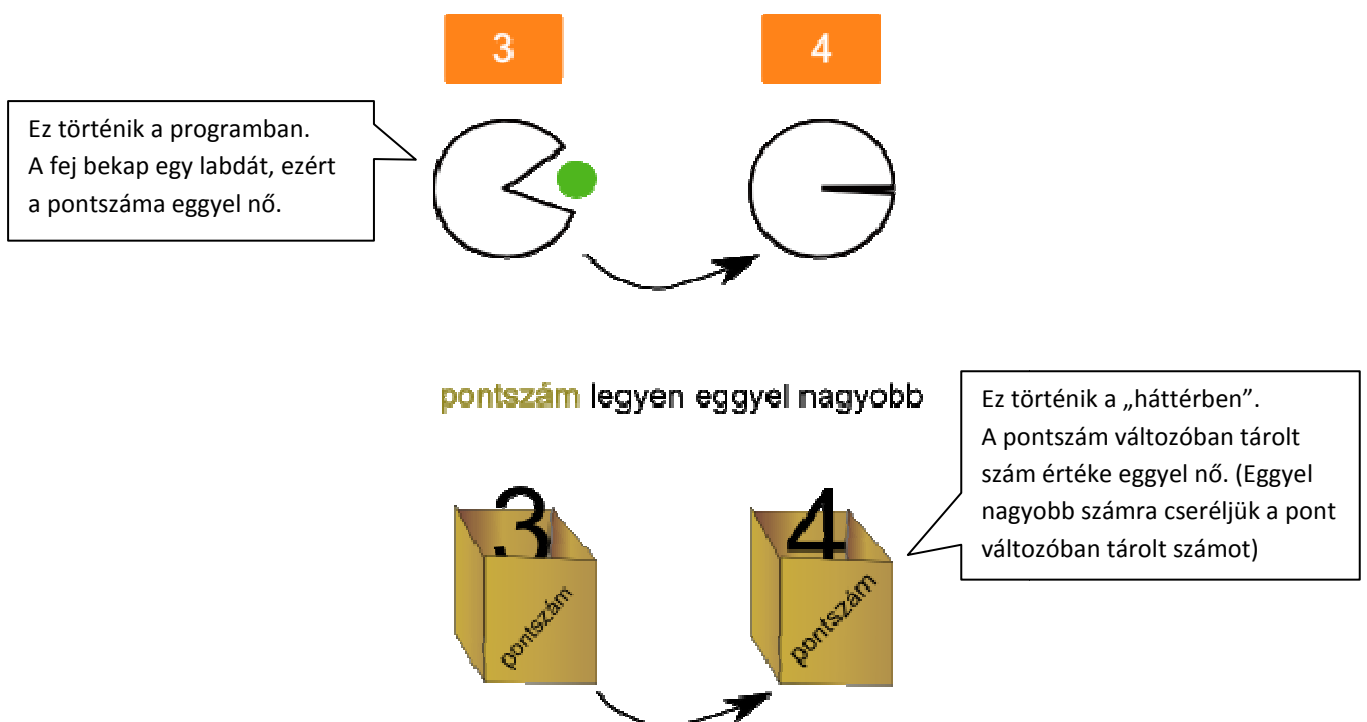
A következő játékot készítjük el:



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: tátozó fej, színes labdák (könnyen megrajzolhatod, de le is töltheted őket)
- Változó: pontszám
- A szereplők mozgása, irányítása kissé eltér az eddiektől. 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 feketehez. Tehát a pontszám egy változó. Létrehozása: **Változó létrehozása**



- 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).



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



- 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 megadhatasz. 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

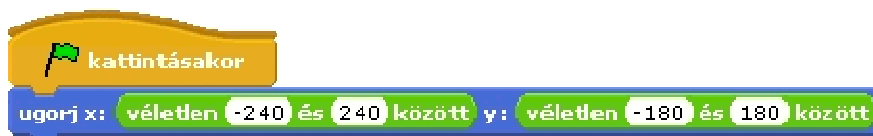
- lássuk először a golyókat:

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



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:



- 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:



Mentsd el a projektedet hamihami néven!

8. Listák

A változók után ebben a leckében a listákkal ismerkedhetsz meg. A következő játékot készítjük el, melyben az irányítótornyos által meghatározott város fölé kell repülni (ha odaértél, nyomd meg a szököz billentyűt):



A listákban számokat vagy szövegeket tárolhatunk sorban egymás után. Új elem hozzáadásakor az elem a lista végére kerül, tehát ez a szerkezet hasonlít egy egyszerű bevásárló listához: ha eszünkbe jut valami, amit venni kell, a lista végére írjuk.

A listák létrehozása

Listákat is a változók csoportban hozhatsz létre a **Lista létrehozása** gombbal. A létrehozás ugyanúgy történik, mint változók esetén. A lista létrejötte után megjelennek a listakezelő parancsok:



A megadott szöveget hozzáadja a kiválasztott (ebben az esetben ez a Város) listához. Az új elem a lista utolsó helyére kerül.



Listához számot vagy akár egy változó aktuális értékét is hozzáadhatsz.



A törlés parancsban az első, az utolsó vagy az összes elem törlését választhatod az adott listából.



A választhatókon kívül törölheted pl. az i változó aktuális értékének sorszámán lévő elemet is.

szúrd be Város 1 helyére: valami

Ennek a parancsnak a segítségével új elemet illeszthetsz a lista első, utolsó, vagy egyik, azaz véletlenszerű helyére.

szúrd be Város i helyére: j

Ide is illeszthetsz változókat. Ez a parancs a Város lista i-edik helyére szúrja be az j változó értékét. Pl. ha $i=3$ és $j=5$, akkor a parancs végrehajtása után a Város lista 3. eleme 5 lesz.

cseréld le Város 1 elemét: valami

Megadja a kiválasztott lista kiválasztott (első, utolsó, egyik vagy egy konkrétan megadott sorszámú) elemét.

A beszúrás parancshoz hasonlóan le is cserélheted a lista elemeit.

Megadja a kiválasztott lista hosszát, azaz elemeinek számát.

Város 1 eleme

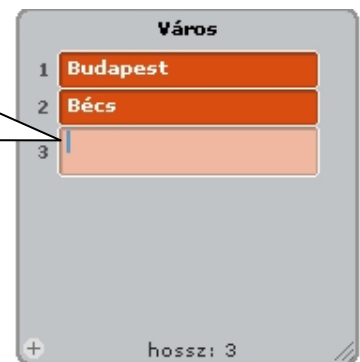
Város hossza

A listák elemeit felveheted, törölheted és módosíthatod a program futása közben a fenti parancsok segítségével, vagy a játéktérben megjelenő listakezelőben is:

A listához a + gomb segítségével adhatsz új elemet.



Az elemek egyszerűen begépelheted és később bármikor módosíthatod.



Azt, hogy melyik listakezelési módot érdemes használni, mindig az adott probléma határozza meg.

A játék elkészítése

- szereplők: repülő, irányítótorony (tőle kapja a repülő a következő célállomást), város (de ő csak hiba esetén jelzi célváros helyét). A háttér és a szereplőket letöltheted a honlapról.
- változó: cél (a célállomás listabeli sorszámát tartalmazza)
- listák: város (az európai fővárosok nevét tartalmazza), X_koordináta (a városok x koordinátáját tartalmazza), Y_koordináta (a városok y koordinátáját tartalmazza)
- A játék: repülj a repülővel az irányítótorony által megadott város fölé, majd nyomd meg a szóköz billentyűt. Ha jó város fölé szálltál, akkor új feladványt kapsz. Ha rossz a találatod, akkor egy pillanatra felvillan a város helye, megmutatja, hová kell repülnöd.
- Először hozd létre a listákat. A városok nevét és a megadott háttérhez tartozó koordinátáikat megtalálod a honlapon. Figyelj arra, hogy a város neve, x és y koordinátái az egyes listák ugyanolyan sorszámú helyére kerüljenek. Ha túl hosszúnak találsz a listát, elég csak 5-10 város adatait megadnod belőle, a játék működését ez nem befolyásolja.

Az irányítótorony feladatai

A kezdeti beállítások után *új* cél üzenetet küld, mivel szükség van úticélra.

```

start üzenet érkezésekor
jelenj meg
kerülj legelőre
ugorj x: -211 y: -120
küldj üzenetet: újcel
    
```

A cél változó értéke egy 1 és **Város** **hossza** közötti véletlenszám. Ha a listában 37 város van, akkor ez a parancs egy 1 és 37 közötti számot ad, ha a lista 5 elemű, akkor 1 és 5 közötti számot.

```

újcel üzenet érkezésekor
cél legyen véletlen 1 és Város hossza között
mondj: Város cél eleme
    
```

Helyes üzenetet a repülőgép küld, ha eltaláltad a várost. Ilyenkor következhet egy új cél.

```

helyes üzenet érkezésekor
mondj: Helyes! (1) 1 mp-ig
küldj üzenetet: újcel
    
```

A torony megadja az úticélt: a **Város** lista cél változóban tárolt számú („cél-edik”) elemét.

```

téves üzenet érkezésekor
mondj: Téves! (2) 2 mp-ig
mondj: Város cél eleme
    
```

Ha a *téves* üzenetet küldi a repülő, akkor a cél változó értéke nem változik, az előző várost kell megtalálni.

A város feladatai

```

téves üzenet érkezésekor
x legyen X_koord cél eleme
y legyen Y_koord cél eleme
jelenj meg
várj 1 mp-et
tűnj el
    
```

A város csak *téves* találat esetén jelenik meg egy másodpercre a célállomás helyén. Ezért x-koordinátája az X_koord lista „cél-edik” eleme, y-koordinátája pedig az Y_koord lista „cél-edik” eleme lesz.

A repülőgép feladatai

A repülőgép irányítása tetszőlegesen megvalósítható. Egy feltételre kell figyelni: amikor a játékos megnyomja a szököz billentyűt, akkor előfordulhat, hogy bár a repülő a megfelelő város fölött van, mégis *téves* üzenetet küld. Ez azért van, mert nagyon nehéz pontosan eltalálni a játéktér egy bizonyos koordinátájú pontját. A játék akkor is élvezetes marad, ha 10 pixelnyi "szabadságot" adunk a cél eltalálásakor:

```

(X_koord cél eleme + 10 > x hely és x hely > X_koord cél eleme - 10)
(Y_koord cél eleme + 10 > y hely és y hely > Y_koord cél eleme - 10)
ha és
küldj üzenetet: helyes
különben
küldj üzenetet: téves
    
```

Mentsd el a projektedet repcsi néven!

Festőablak

A festőablakban saját szereplőket vagy háttereket rajzolhatsz, illetve módosíthatod a már meglévőket. Az ablakot több gombbal is előhívhatod:



A programozási tér hátterek (játéktér esetén) vagy jelmezek (szereplő esetén) fülén.



A Festőablak funkciói

BETÖLTÉS
betölthetsz egy már létező képet.

VISSZAVONÁS
az utolsó 9 lépést lehet visszavonni

PALETTA
innen választható ki a rajzoló- és a háttérszín

ÁTLÁTSZÓ „SZÍN”
Egy speciális szín. Főleg szereplők rajzolásakor alkalmazzuk. Amit átlátszóra színezzük, nem takarja el a mögötte lévőket.

FORGÁSKÖZÉPPONT
az a pont, amely körül elforgatjuk a szereplőt és az a pont, amely a tollhegy lesz a tollat le parancs esetén.

TÜKRÖZÉS
vízszintes vagy függőleges tengelyre tükrözi a az egészképet vagy a kijelölt részt.

TÖRLÉS
törli a képet (visszavonható)

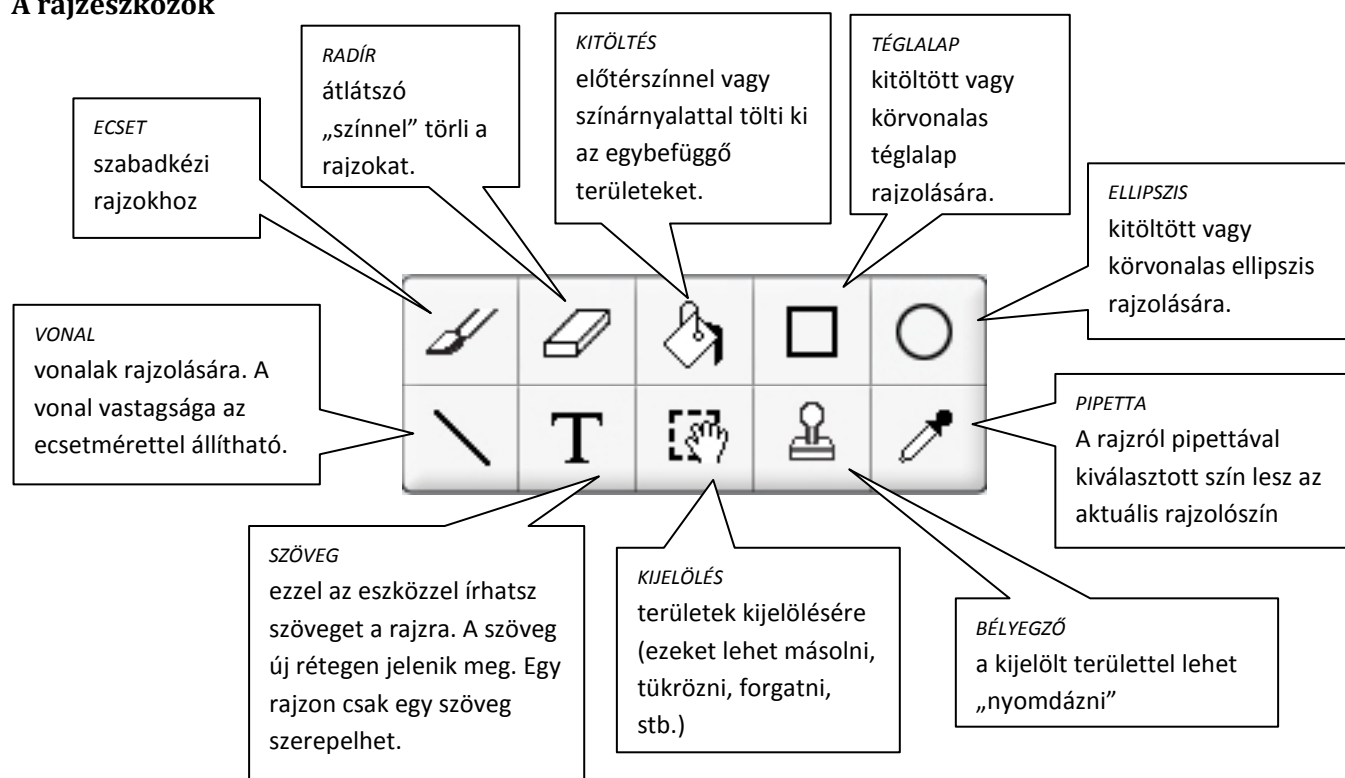
ISMÉT
megismétli az utoljára visszavont lépéseket.

RAJZOLÓSZÍN, HÁTTÉRSZÍN
az aktuális rajzoló- és háttérszín. Jobb gombbal kattintva lehet váltani közöttük.

SKÁLA
a nézet változtatására használható: közelebről vagy messzebről nézhetjük a rajtot. Tehát a tényleges mérete nem változik, csak könnyebb rajzolni az apró részleteket.

Középpont beállítása

A rajzeszközök



Forgási stílusok

A szereplők a **ha szélen vagy, pattanj vissza** parancs teljesítése után néha „fejre állnak”, máskor viszont egyszerűen megfordulnak. Ez a forgási stílusuktól függ, amelyet a programozási tér fölött állíthatsz be.

Egy kísértet mozgásán keresztül láthatsz példát a stílusokra:



A kísértet feladata:



Viselkedése a forgási stílusától függően:



Minden irányba elfordul

A szereplő ilyenkor mindig a haladási irányába néz. (A haladási irányát a kék pálcika jelzi).



Csak balra és jobbra néz

A szereplő ekkor mindig csak jobbra vagy balra néz. Ha jobbra száll (és közben akár emelkedhet vagy süllyedhet is), akkor jobbra néz, ha pedig balra repül, akkor balra fordul el.



Sosem fordul el

A szereplő mindig az eredeti irányába néz, és nem is fordul el.

Kinézet

A leckékben már megismerkedhettél a **parancskészlet** Kinézet csoportjának egy részével. Ez a kiegészítés összefoglalja a velük kapcsolatos tudnivalókat.

jelmez legyen válts jelmezt

Ezekkel a parancsokkal lehet a szereplő kinézetét meghatározó jelmezeket váltani. A válts jelmezt a jelmezek sorában következő jelmezre vált.

mond: 2 mp-ig 

Megjeleníti egy szövegbuborékban a szereplőnek adott szöveget (a „Szia” helyére változót vagy listaelemet is illeszthetünk, akkor azok értékét mondja a szereplő). A szövegbuborék az első esetben a megadott ideig látszik, a második esetben újabb Mondd parancs kiadásáig. (Akkor tűnik el, ha a parancsot „üresen” adod ki.)

gondold: 2 mp-ig 

A Mondd parancshoz hasonlóan működik, csak itt szövegbuborék helyett gondolatfelhő jelenik meg a szereplő fölött.

méret változzon: méret legyen % 

Ezekkel a parancsokkal a szereplő méretét változtathatod. Az első esetben a megadott mértékkel (csökkentheted is, ha ez a szám negatív). A második esetben azt adhatod meg, hogy az eredeti méretnek (ez a 100%) hány százaléka legyen az új.

jelenj meg

A szereplők lehetnek láthatatlanok is (tűnj el). Ilyenkor a többi szereplő nem érzékeli őket. A megfelelő pillanatban láthatóvá teheted őket a jelenj meg paranccsal.

kerülj legelőre szinttel hátrébb

A szereplők külön rétegeken helyezkednek el. Aki előrébb van, az eltakarhatja a mögötte lévőket. Ha azt szeretnéd, hogy egy szereplőt ne takarhasson el senki, akkor használd a kerülj legelőre parancsot, vagy az őt takarókat küldd néhány szinttel hátrébb.

szín hatás változzon szín hatás legyen

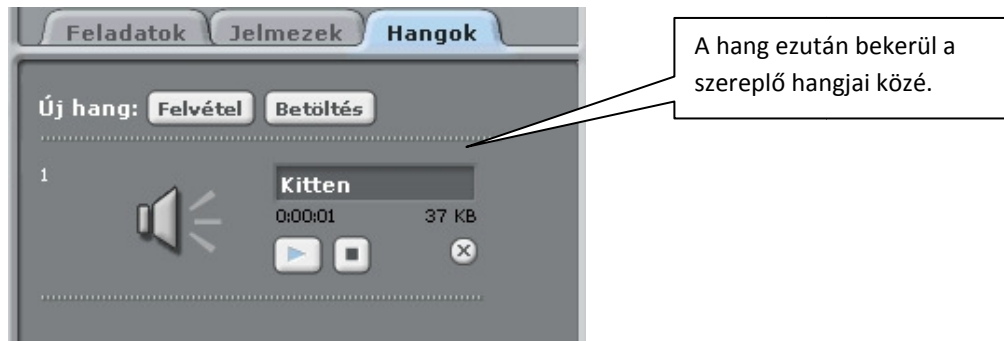
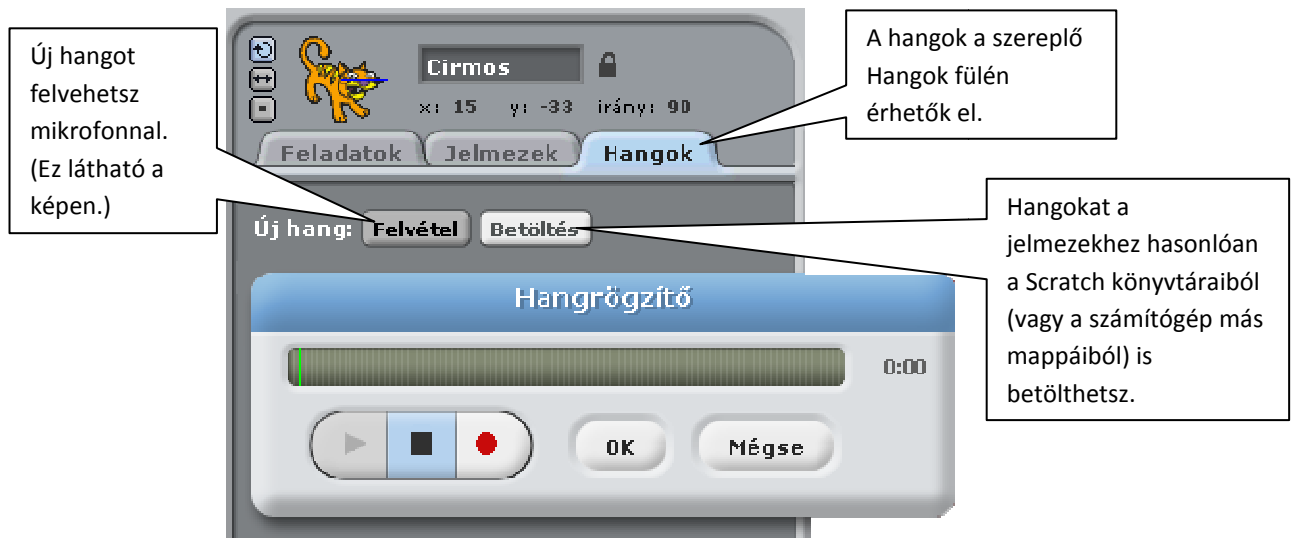
Ezekkel a parancsokkal különböző grafikus hatásokat adhatsz a szereplőkhöz, vagy törölheted ezeket a hatásokat. (Próbáld ki mindet!)

Hangok

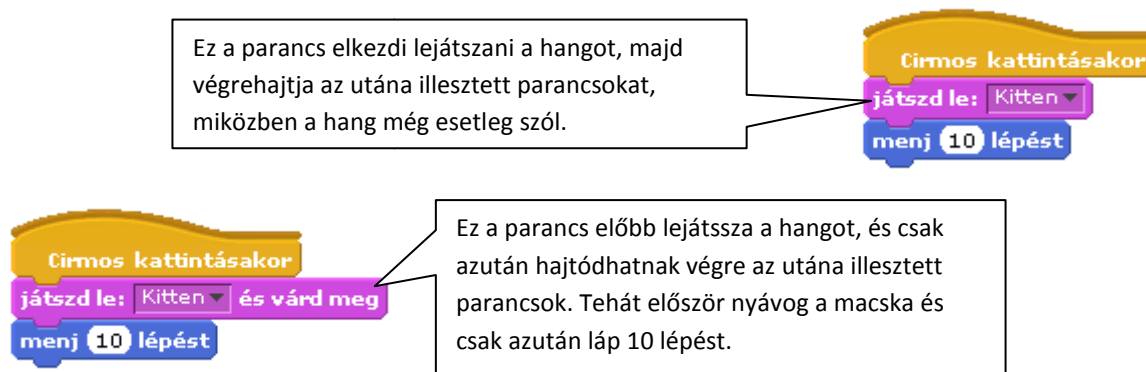
Felvett hangok

A szereplők nemcsak szövegbuborékkal jelzett „hangokat”, hanem igaziakat is tudnak produkálni. Ezt ráadásul kétféleképpen is megtehetik.

A hangok kezelése hasonló a jelmezekéhez. A szereplőhöz tartozó hangok a Hangok fülön találhatóak. Ez a szereplő létrehozásakor általában üres, tehát a szereplő még nem tud hangot adni.

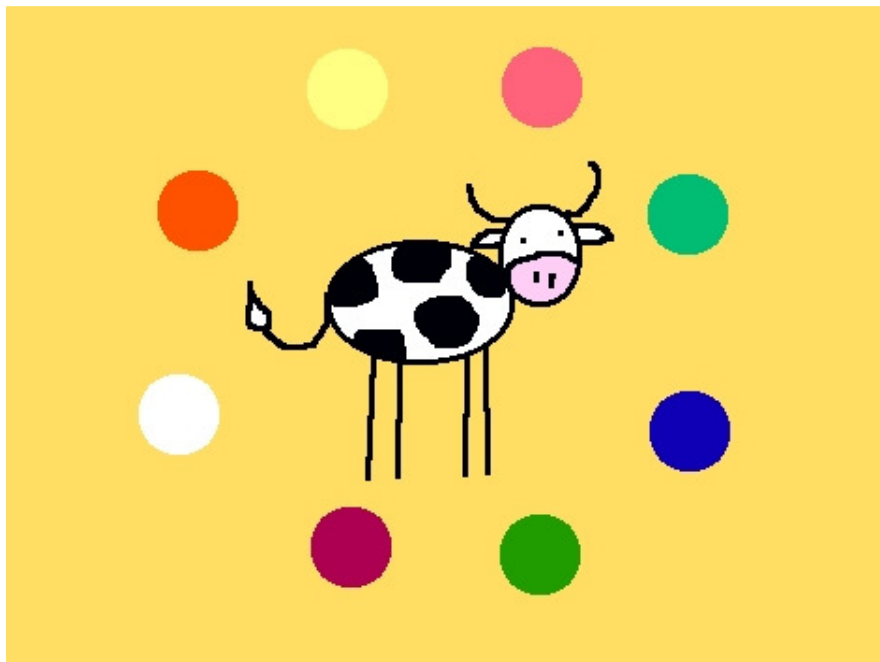


A hangokat a projektben a Hang parancscsoport parancsaival lehet kezelni. A Cirmos macskához tartozó Kitten hangot például így lehet előhívni:




Beépített hangszerek

A szereplők nemcsak beépített hangokat tudnak lejátszani, hanem különböző hangszerekkel játszott dallamokat is. A következő projekt például így készült:



A színes gömbökre kattintva különböző hangszereken csendül fel a mindenki által jól ismert Boci, boci tarka... első néhány hangja. A labdák feladata :

Itt kell megadni, hogy melyik hang hány ütemig szóljon. A hangok a legördülő menüből egy zongora billentyűiről választhatók ki.



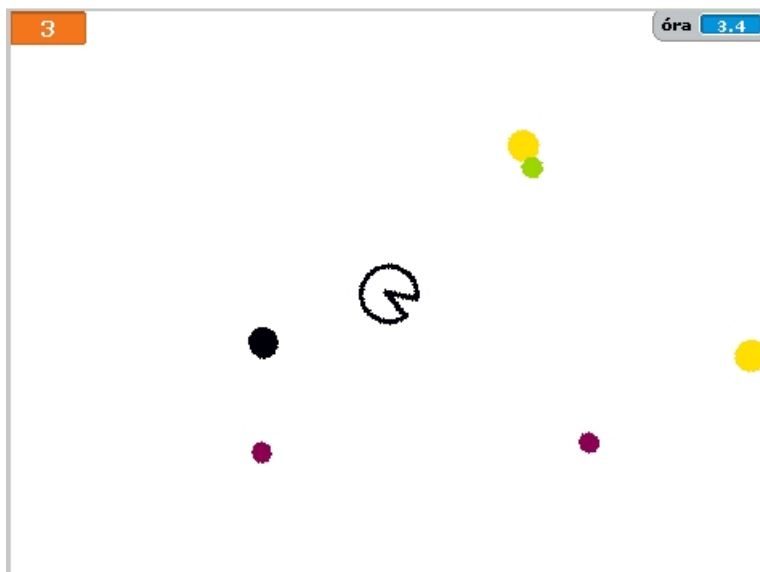
6 kattintásakor
hangszer legyen 25
szóljon 48 0.5 ütemig
szóljon 52 0.5 ütemig
szóljon 48 0.5 ütemig
szóljon 52 0.5 ütemig
szóljon 55 1 ütemig
szóljon 55 1 ütemig

Először be kell állítani, hogy melyik legyen a hangszer, amely majd lejátsza a hangokat. Ebben az esetben ez a 25-ös számú klasszikus gitár. A hangszerek a legördülő menüből választhatók.

A hangerő és a tempó parancsokkal változtatható, de általában megfelelőek az előre beállított értékek.

Az óra

A 7. leckéhez készített játék pontosan 30 másodpercig tart. Ehhez a Scratch beépített óráját kell használni. Lássuk, hogyan! Emlékeztetőül a játék:



A Scratch órája

Az óra a Scratch indításától kezdve folyamatosan jár, a megnyitás pillanatától eltelt időt mutatja másodpercekben. A parancskészlet érzékelés csoportjában a neve (óra) melletti szürke négyzet kattintásával jeleníthető meg a játéktérben, vagy tüntethető el onnan.

A megjelenítés a változókhöz hasonlóan kétféle lehet:  valamint 

Az órát egyetlen paranccsal lehet befolyásolni: . Ez értelem szerűen nullára állítja az óra értékét, azaz újratekinti a számolást. Az időmérést használó játékok elején ezt érdemes megtenni:

INICIALIZÁLÁS

a játék kezdetén érdemes egy külön sapkán beállítani a kezdeti értékeket. Például hogy az óra 0-ról induljon és persze a pontszám is 0 legyen.



Az óra pillanatnyi értékének nemcsak a megjelenítése, hanem a kezelése is hasonló a változókéhoz: a játék akkor ér véget, ha az óra értéke nagyobb, mint 30.



Azért nem azt vizsgáljuk, hogy az óra=30 feltétel teljesül-e, mert a programnak egyszerre sok feladatot kell elvégeznie, és lehet, hogy abban a pillanatban, amikor az óra értéke pontosan 30 másodperc, a program éppen nem azzal a vizsgálattal foglalkozik, amelyben az óra=30 feltételt ellenőrizzük, így nem állít le mindent.

1.	Ismerkedés a Scratch környezettel	1
	A Szereplők	1
	Új háttér betöltése	2
2.	Megmozdulnak a szereplők	3
	A játék elkészítése	3
	Variáció	5
3.	Ismétlődések	6
	Ciklusok	6
	A program elkészítése	6
	Feladat	7
4.	Feltételek	8
	A játék elkészítése	8
	Egymásba ágyazás	10
5.	Üzenetek	11
	Ismerkedés az üzenetekkel	11
	Vége a játéknak	13
6.	Rajzoljunk	14
	A program elkészítése	14
7.	Változók	16
	Változó	16
	A játék elkészítése	17
	Pontszámolás	17
	Véletlenszámok	18
	A mozgások megvalósítása	19
8.	Listák	20
	A listák létrehozása	20
	A játék elkészítése	21
	Az irányítótorony feladatai	22
	A város feladatai	22
	A repülőgép feladatai	22
	Festőablak	23
	Az ablak felépítése	23
	A rajzeszközök	24
	Forgási stílusok	25
	Minden irányba elfordul	25
	Csak balra és jobbra néz	25
	Sosem fordul el	25
	Kinézet	26
	Hangok	27
	Felvett hangok	27
	Beépített hangszerek	28
	Az óra	29
	A Scratch órája	29