



Scratch-ról általánosságban

A Scratch egy ingyenes programozási környezet, amellyel játékokat, animációkat vagy meséket is készíthetsz.

A Scratch-ben való programozás **kirakó játékozásra** hasonlít: parancsokat és változókat felhasználva úgy lehet algoritmusokat összeépíteni, mint egy kirakót a darabjaiból. Ezek az elemek **csak helyes módon** illeszkednek egymáshoz.

Vezérlésutasítások

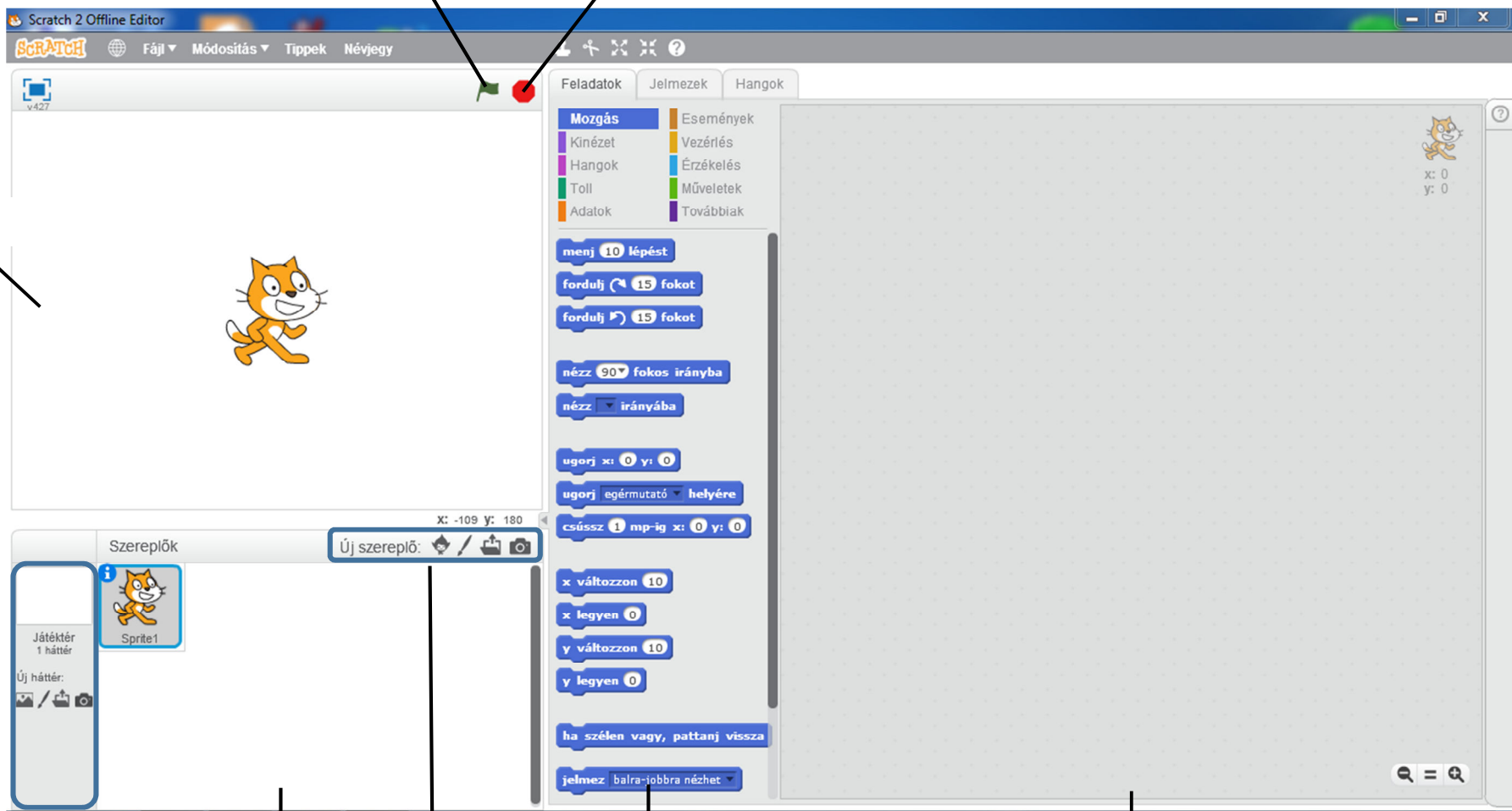
szekvencia	A kirakóelemek legtöbbje mindkét irányból enged csatlakozást, de vannak feladatkezdő (eseményjelző sapkák), és feladatlezáró elemek is.	
ismétlés	Lehetőség van adott számszor, egy feltétel teljesüléséig, vagy állandóan ismételni egy utasítássort.	
elágazás	A <i>ha, ha ... különben ...</i> szokásos szerkezeteken túl különleges feltételvizsgáló parancsokat is lehet használni, amelyek a játékkészítést egyszerűsítik (ilyen például a <i>mindig ha ...</i> parancs).	

Scratch felülete

Feladatok elindítása

Feladatok megállítása

Játéktér



Szereplőlista

Parancskészlet

Programozási tér

Új háttér létrehozása

Új szereplő létrehozása

1. Új szereplő választása

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.

Aktuális szereplő:
Neki adhatunk
feladatokat a
programozási térben



Új szereplő festése:
saját magad is tervezhetsz, festhetsz
szereplőt

Szereplő kiválasztása könyvtárból:
választhatsz szereplőt a Scratch
beépített szereplői közül

2. Szereplők jelmezei:

Egy szereplőnek több jelmeze is lehet, amelyek valamilyen esemény (például billentyű lenyomás hatására) megváltozhatnak. A jelmezek folyamatos változtatásával szereplőinket animálttá tehetjük. A hozzáadott szereplő jelmezek sorrendje tetszőlegesen módosítható a jelmezek mozgatásával.



3. Új háttér létrehozása

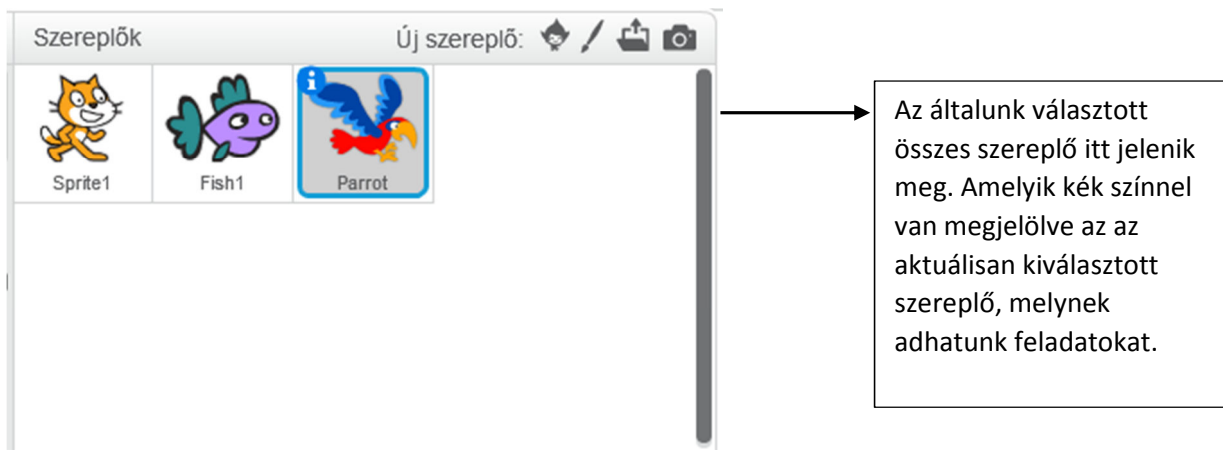
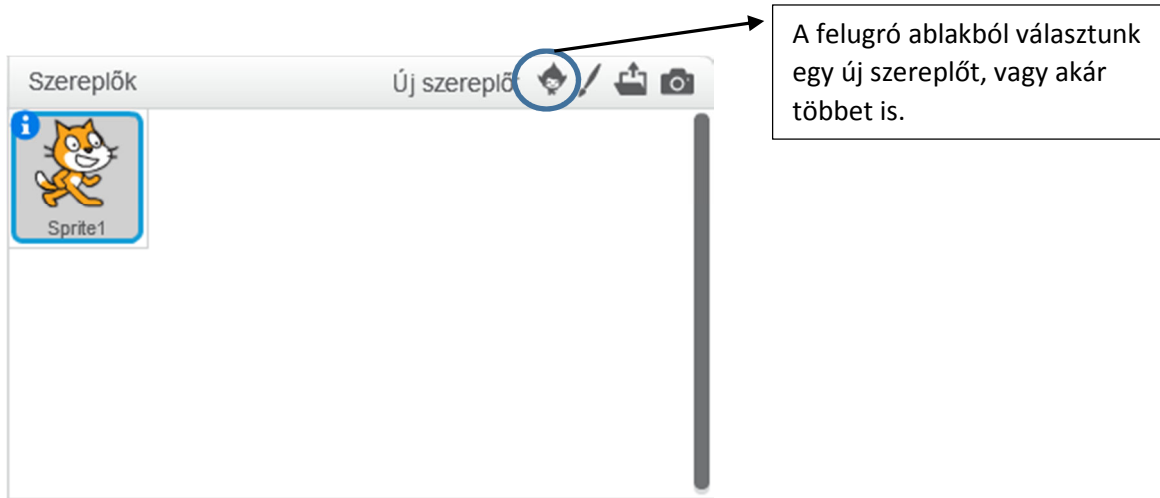
A háttér ugyanolyan fontos része a programunknak vagy az animációnknak. Fontos hogy szép és élvezetes legyen. A háttér alapbeállítása a fehér, módosítása hasonló módon történik, mint a szereplő változtatásánál. Vagyis betölthetünk már kész háttereket, melyek a Scratch beépített hátterei vagy mi magunk is rajzolhatunk.

The screenshot shows the Scratch 'Szereplők' (Sprites) panel. The 'Játéktér' (Stage) area is highlighted with a blue box and labeled 'Játéktér 1 háttér'. A parrot sprite is visible in the 'Szereplők' area. The 'Új háttér:' (New Background) section is highlighted with a blue box and labeled 'Új háttér festése'. This section contains icons for selecting a background from the library, drawing a new one, and taking a photo. A box on the left labeled 'Új háttér kiválasztása könyvtárból:' (Selecting a new background from the library) points to the library icon. Another box labeled 'Kiválasztott háttér: alapbeállításként, ez fehér' (Selected background: default, this is white) points to the white background area. A third box labeled 'Új háttér festése' (Drawing a new background) points to the drawing icon.

Szereplő változtatása

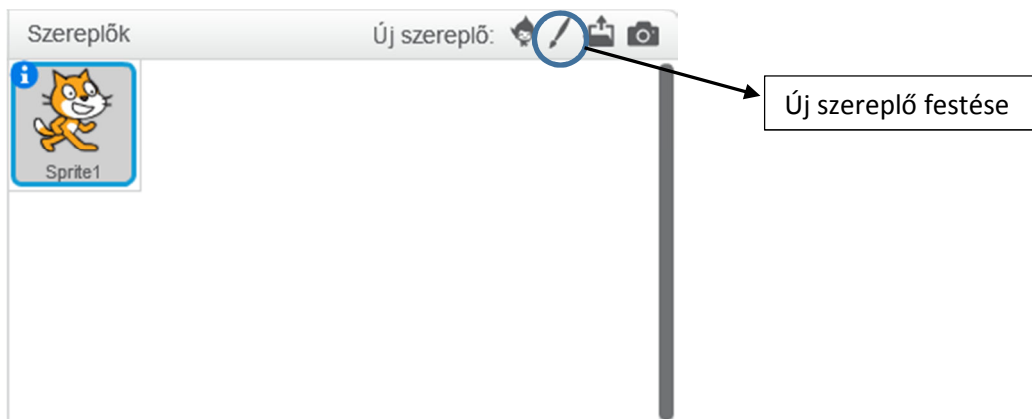
1. Szereplő kiválasztása könyvtárból

Először nézzük, hogyan lehet a Scratch beépített szereplői közül kiválasztani új szereplőt.

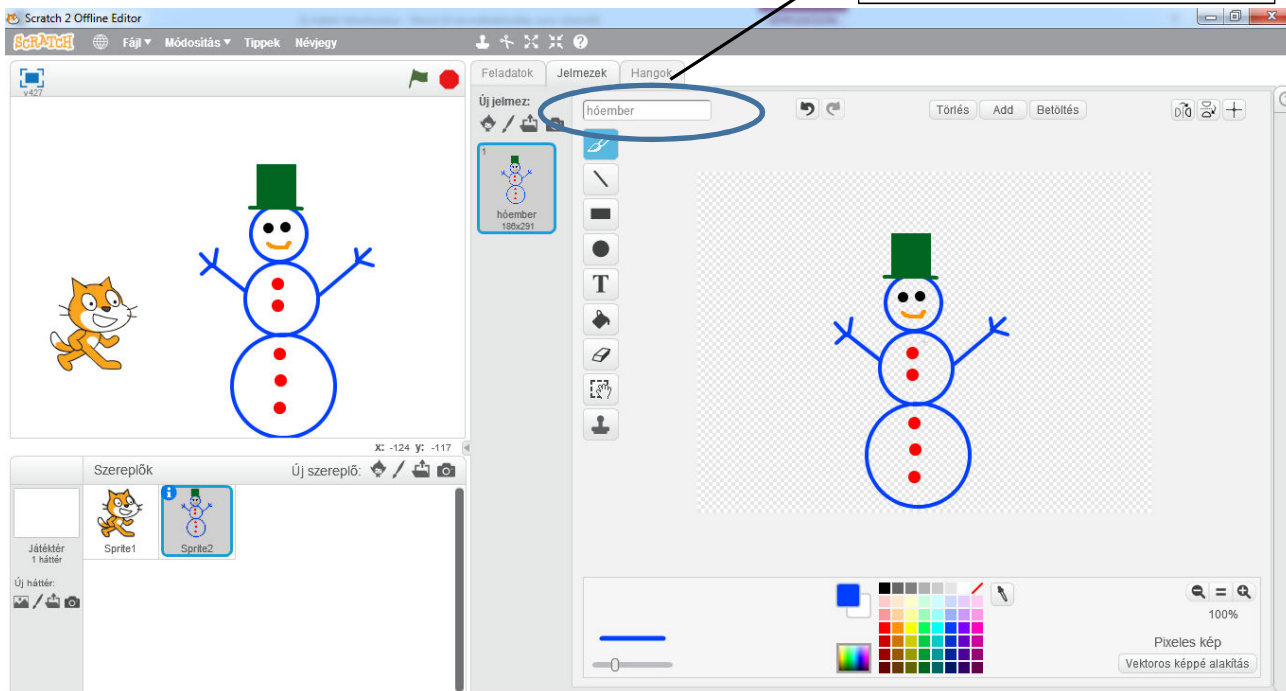


2. Új szereplő festése

Ha szeretnénk, saját magunk is tervezhetünk szereplőt. A program ezen része nagyon hasonlít a Paint rajzoló programhoz.



Itt adhatunk nevet az elkészült rajzunknak, szereplőnknek.

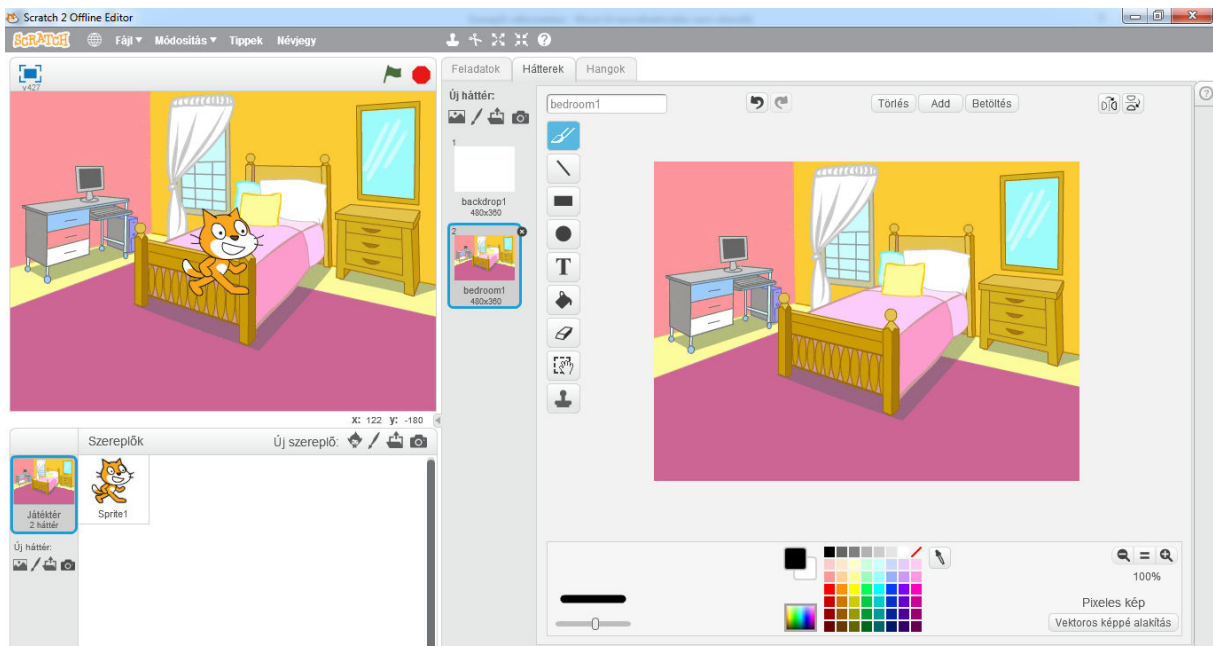


Új háttér létrehozása

1. Háttér kiválasztása könyvtárból

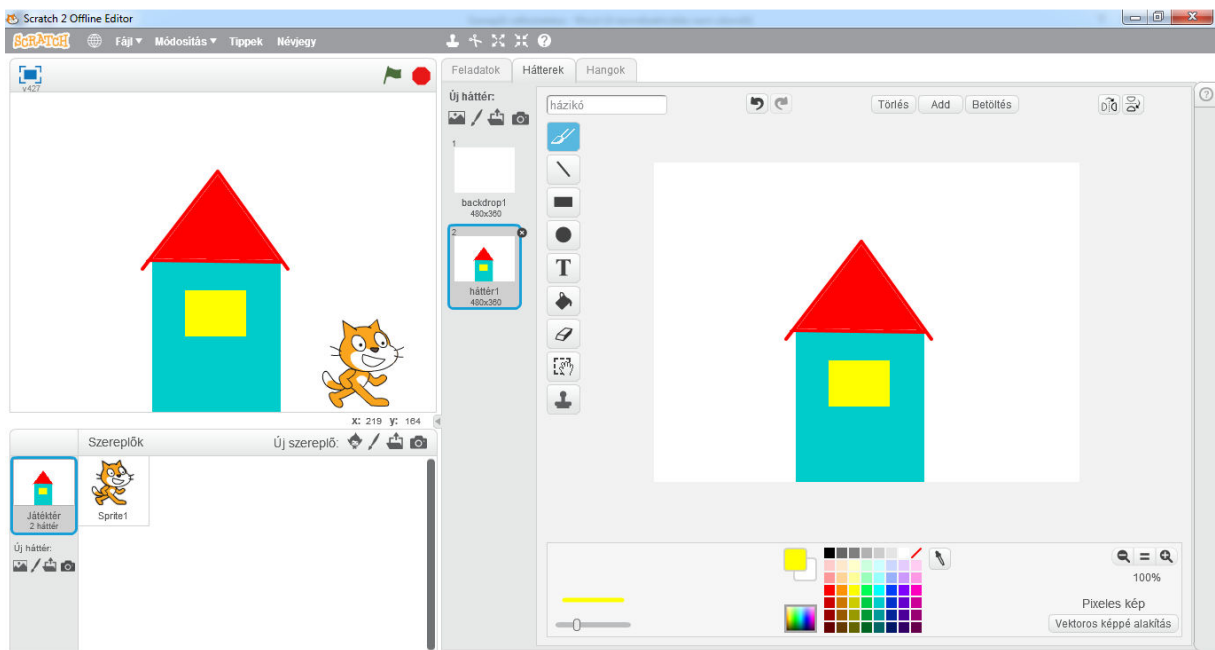
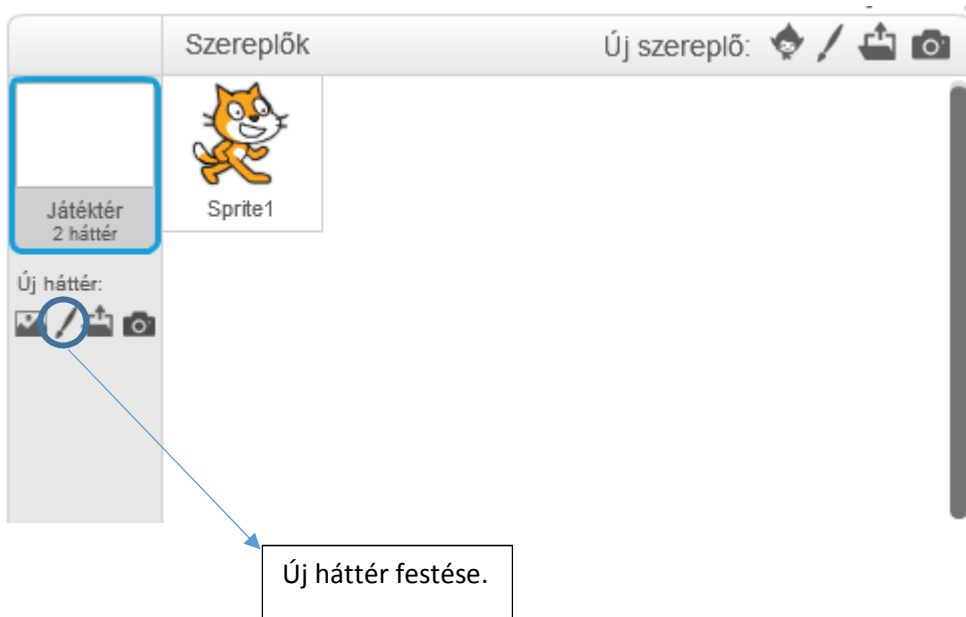


A felugró ablakból
választunk háttérret.



2. Új háttér festése

Ha szeretnénk, saját magunk is tervezhetünk háttérret. A program ezen része nagyon hasonlít a Paint rajzoló programhoz.



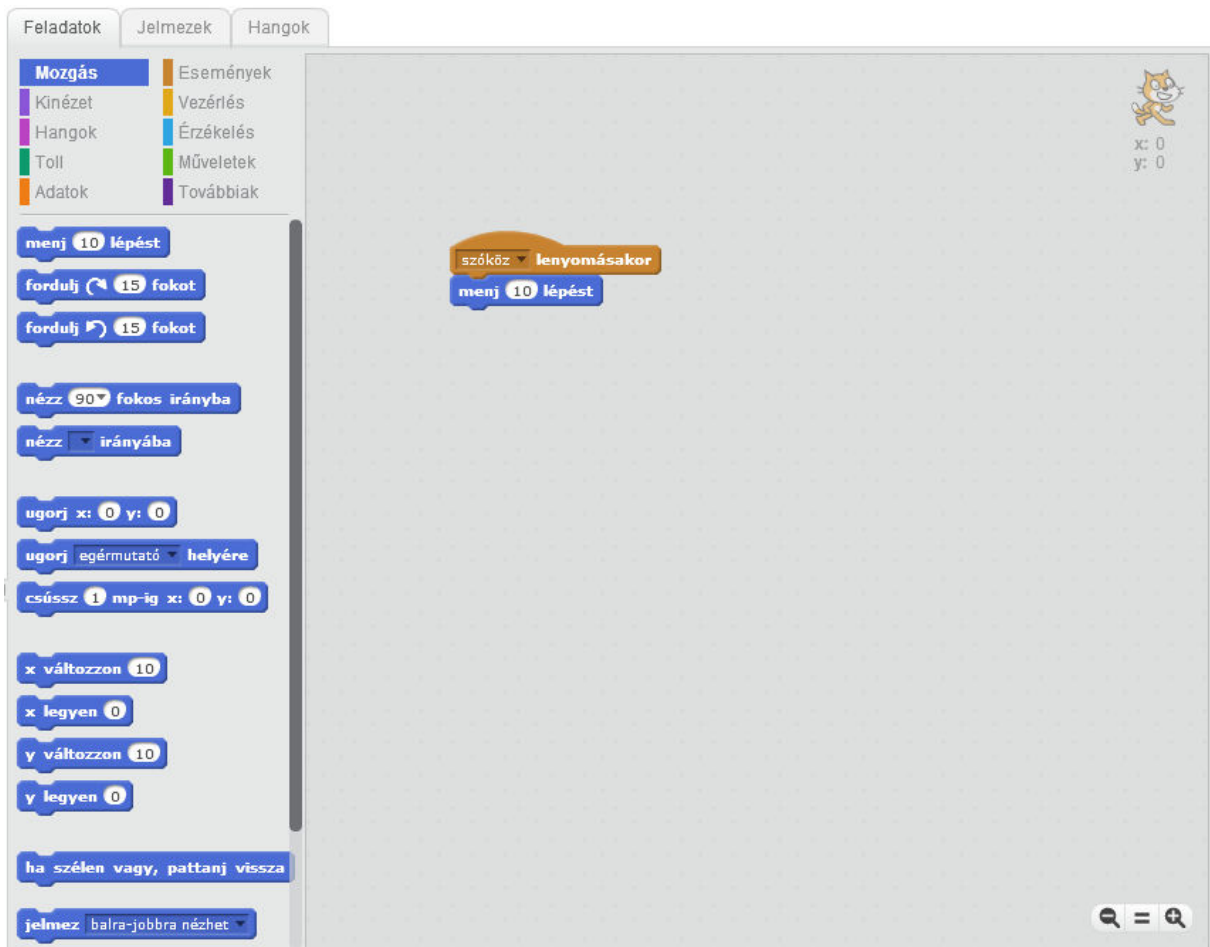
1. lecke: Mozgassuk meg a cicát!

Egy játék akkor szórakoztató, ha irányíthatjuk a szereplőit.

Most a cicát fogjuk megmozgatni: menjen előre, majd forduljon vissza

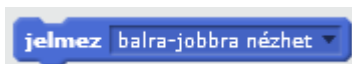
Lépéseink:

- adjunk a macskának feladatokat, ehhez húzzuk a parancsokat az egérrel a parancskészletből a programozási térbe. A parancsokból akkor lesz feladat, ha megfelelő sorrendben összeillesztjük őket egy sapka alatt.
- először adjuk meg, hogy a macskánk a szóköz billentyű lenyomásakor induljon meg előre



- A programot azonnal kipróbálhatod. A zöld zászló lenyomásakor a macska a játéktérben 10 lépést (10 képpontnyit) halad előre. Azonban nem áll meg a szélénél, kimegy a képből.
- Ezt úgy tudjuk elkerülni, hogy **ha szélén vagy, pattanj vissza** parancsot betesszük az előző két utasítás közé
Ezzel elértük, hogy a szereplőnk már nem fog kimenni a játéktérből

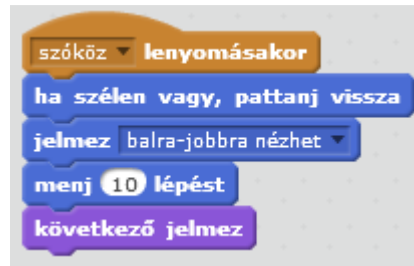
Még egy probléma merül fel a cica mozgatásával, ha a széléről visszapattan, akkor fejfel lefelé fog tovább haladni. Ezt a következő paranccsal oldhatjuk meg:



- Most már csak azt szeretnénk elérni, hogy tényleg úgy mozogjon, mintha menne a mi macskánk, ehhez pedig a következő feladatot kell megadni:

következő jelmez

Tehát a cicánk feladatai:



Kész program: [cica](#)

Házi feladat:

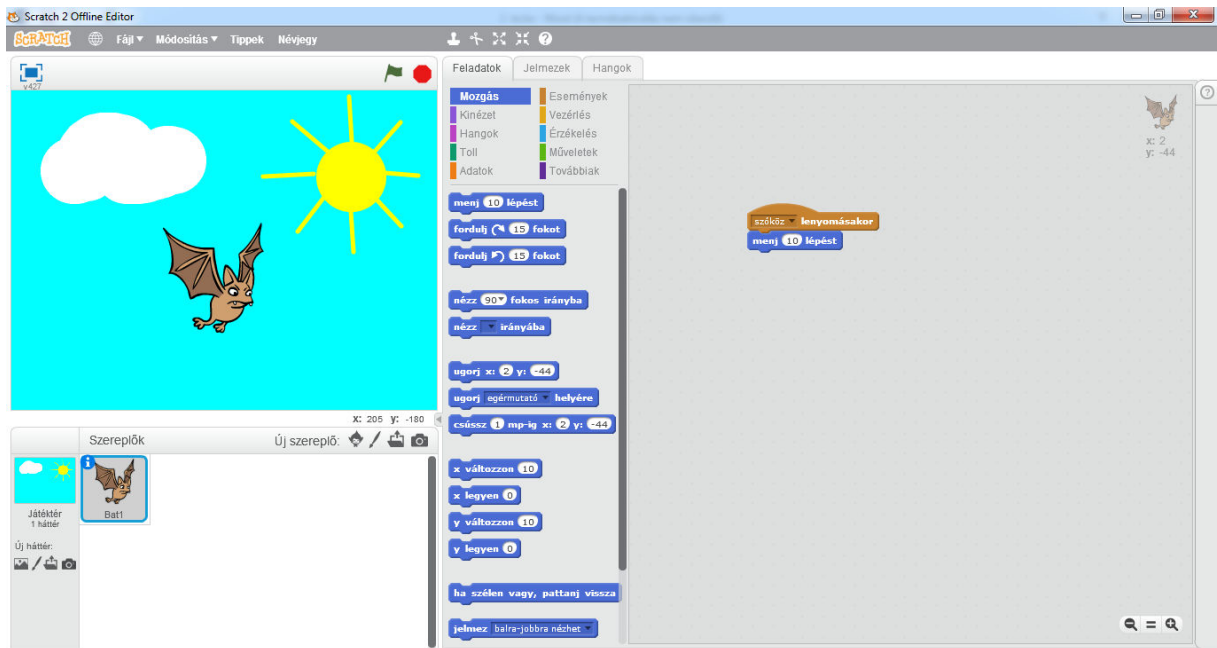
Készítsd el a cicának az előbb leírtak alapján:

- a.) Bal gomb lenyomásakor forduljon balra 15 fokot
- b.) Jobb gomb lenyomásakor forduljon jobbra 15 fokot

2. lecke: Háttér

Az előzőektől egy kissé bonyolultabb feladatunk lesz. Most a hátteret is meg fogjuk változtatni.

- Töröld ki a macskát a játéktérből és töltsd be új szereplőnket bat-1 (denevér).
- Rajzoljuk meg a hátteret festőablakban: kék ég felhőkkel és napocskával.
- Most a denevérnek adunk különböző feladatokat, mint ahogy már megismertük az előző leckében, vagyis ehhez húzzuk a parancsokat az egérrel a parancskészletből a programozási térbe.
- először adjuk meg, hogy a denevér a szóköz billentyű lenyomásakor induljon meg előre



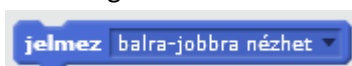
- Ahhoz, hogy a denevér haladási irányát is változtatni tudjuk, még két egyszerű feladatot kell adnunk az eddigihez:



- A programot azonnal kipróbálhatod. A szóköz lenyomásakor a denevér a játéktérben 10 lépést (10 képpontnyit) halad előre. Azonban nem áll meg a szélénél, kimegy a képből.

- Ezt úgy tudjuk elkerülni,  hogy parancsot betesszük az előző két utasítás közé
Ezzel elértük, hogy a szereplőnk már nem fog kimenni a játéktérből

Még egy probléma merül fel a denevér mozgásával, ha a széléről visszapattan, akkor fejfel lefelé fog tovább haladni. Ezt a következő parancssal oldhatjuk meg:



- Most már csak azt szeretnénk elérni, hogy tényleg mozgassa a denevér a szárnyait, ehhez pedig a következő feladatot kell megadni:

következő jelmez

- 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.)

Tehát a denevér feladatai:



Kész program: [denever](#)

Házi feladat:

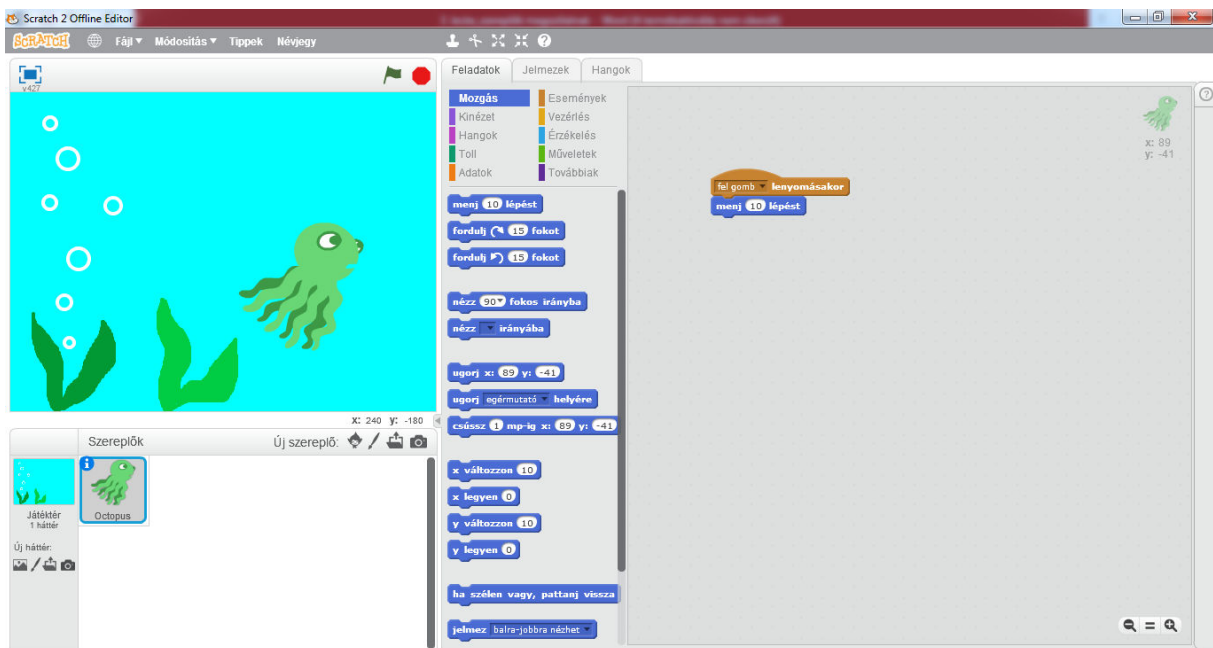
Készíts egy hasonló animációt. A főszereplő az elefánt és a háttér pedig egy fa legyen, felhővel.


Kép a házi feladathoz: [elefant](#)

3. lecke - Megszólalnak a szereplők

Ebben a részben a szereplő mozgatása mellett, meg is fogjuk szólaltatni.

- Töröld ki a macskát a játéktérből és töltsd be az új szereplőnket octopus (polip) néven.
- Rajzoljuk meg a hátteret festőablakban: tenger alatti világot. (lásd lent)
- Most a polipnak adunk különböző feladatokat, mint ahogy már megismertük az előző leckében, vagyis ehhez húzzuk a parancsokat az egérrel a parancskészletből a programozási térbe.
- Először adjuk meg, hogy a polip a fel billentyű lenyomásakor induljon meg előre



- A programot azonnal kipróbálhatod. A szóköz lenyomásakor a polip a játéktérben 10 lépést (10 képpontnyit) halad előre. Azonban nem áll meg a szélénél, kimegy a képből.
- Ezt úgy tudjuk elkerülni,  hogy parancsot betesszük az előző két utasítás közé
Ezzel elértük, hogy a szereplőnk már nem fog kimenni a játéktérből

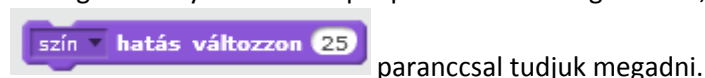
Még egy probléma merül fel a polip mozgatásával, ha a széléről visszapattan, akkor fejfel lefelé fog tovább haladni. Ezt a következő parancssal oldhatjuk meg:



- Most már csak azt szeretnénk elérni, hogy tényleg mozogjon a polip, ehhez pedig a következő feladatot kell megadni:



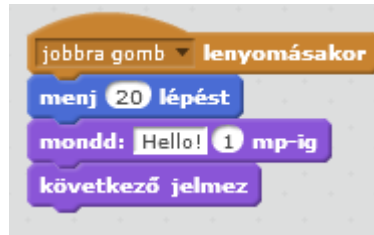
- A következőkben pedig a jobb és bal gombhoz rendelünk még különböző feladatokat.
- A bal gomb lenyomásakor a polip változtassa meg a színét, miközben halad előre. Ezt a



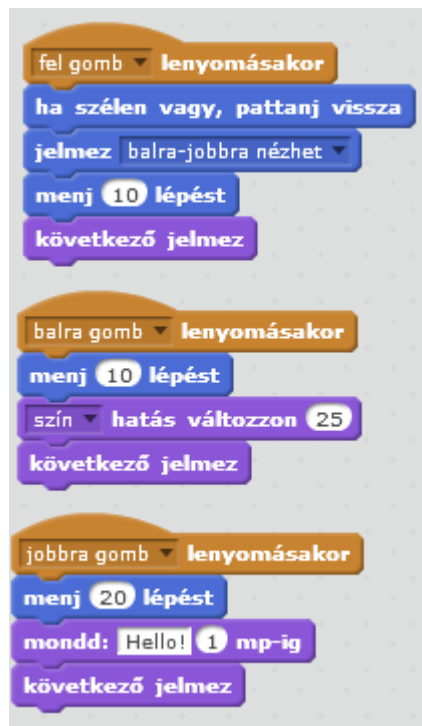
Ez a következőképpen fog kinézni:



- A jobb gomb megnyomásakor pedig menjen előre és köszönjön.
Ez a következőképp fog kinézni:



Tehát a polip feladatai:



Kész program: [polip](#)

Házi feladat:

- a.) A le gomb megnyomásakor a polip ugorjon oda ahová az egér/kurzor mutat.
- b.) A szóköz gomb lenyomásakor polip változtassa meg a méretét, miközben halad előre.

4. lecke: Ismétlődések, ciklusok

Az előzőekben a szereplő csak akkor csinált valamit, ha leütöttünk egy billentyűt a billentyűzeten.

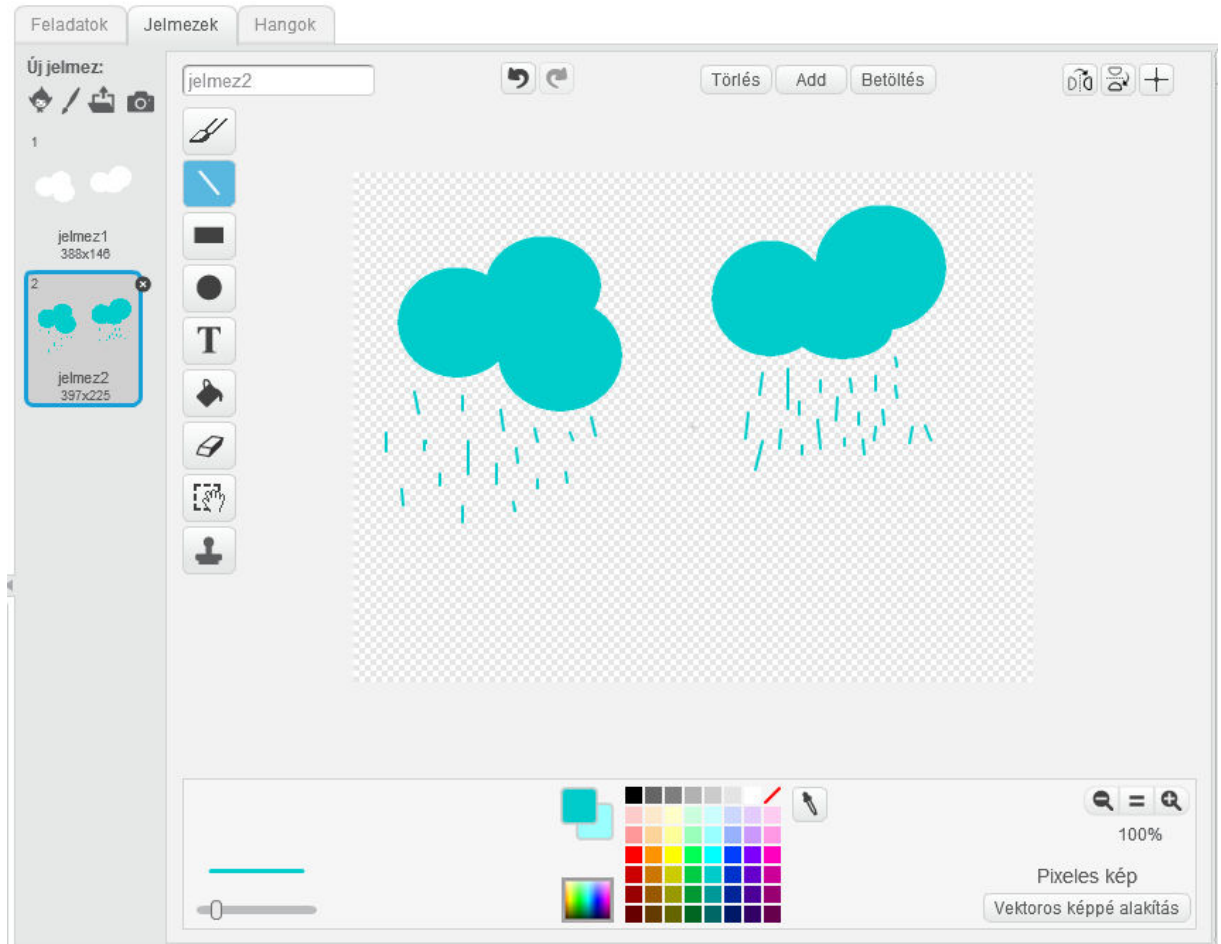
Ciklusok: ismétlődő tevékenységek megvalósítására szolgálnak. Most két ciklusfajtaival fogunk megismerkedni.


- mindig: folyamatosan ismétli a benne elhelyezett parancsokat
- ismételd: a megadott számú alkalommal ismétli a benne elhelyezett parancsokat

Most a programunkban egy elefánt fog sétálgatni, miközben hirtelen elkezd esni az eső.

Lépéseink:

- Készítsünk egy egyszínű hátteret
- Töltsük be az új szereplőnként elephant néven.
- Készítsük el magunk a másik szereplőnként: felhőket ez az első jelmez, a második jelmeze legyen, amikor esik belőle az eső.



- A parancsok a zöld zászló lenyomásával indulnak el.
- Most ebben az esetben a felhők fogják mindig, folyamatosan változtatni a jelmezüket. Ezt a már a lecke elején említett  parancssal érjük el.

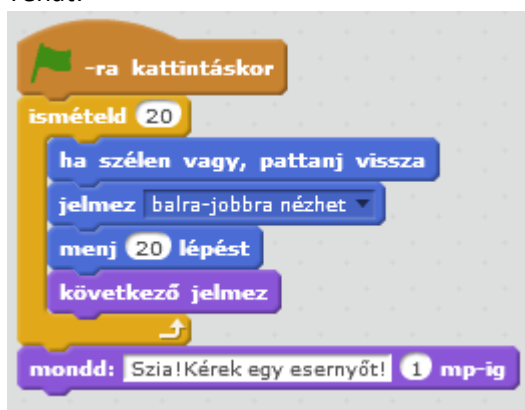
Tehát:



- Az elefánt feladata, hogy sétálgat, és közben mond valamit. Tehát az elefánt ismétli a feladatát valahányszor (pontosabban amennyit megadunk neki), vagyis nem végtelen sokszor, mint a felhő. Ezt a következő kirakó darab megadásával tudjuk elérni:



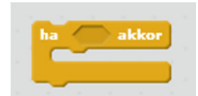
Tehát:



Kész program: [felhos_elefant](#)

5. lecke: A „Ha” feltétel

A játékok készítésénél a szereplők nem mindig ugyanúgy viselkednek minden helyzetben. Ahhoz hogy ezen változtatni tudjunk, a parancsokat feltételekhez fogjuk kötni. Ebben a leckében a feltétellel fogunk megismerkedni, dolgozni.



Ebben a programunkban egy autót fogunk elvezetni a parkolóba.

Lépéseink:

- Készítsünk egy hátteret, amely egy labirintushoz hasonlít vagy töltsük be a **parkoló** nevű fájlt.
- Töltsük be az új szereplőnként egy tetszőleges autót.
- A szereplőnk ebben az esetben fel-le illetve balra és jobbra fog mozogni, így ezt a legkönnyebben egy koordináta-rendszerben fogjuk tudni megvalósítani, ahol az x és az y tengelyen fog mozogni. Állítsuk be a gombokat:



- Ahhoz, hogy a játékot el tudjuk indítani az autónknak a kiinduló pozícióban kell, hogy álljon, a mi esetünkben a sárga négyzeten. Legegyszerűbben úgy tudjuk megállapítani a kezdő koordinátákat, ha az autónkat a kezdő pozícióba állítjuk és leolvassuk a koordinátákat. És a következőképpen állítjuk be:



- Most már tudjuk mozgatni az autót, de sajnos bámerre tudom irányítani, nem csak a kijelölt útvonalon. Ezt a problémát tudom kiküszöbölni a „ha” feltétellel. pl. ha kék színt érinti az autó, akkor ugorjon vissza a kezdő pozícióba.



6. lecke: A „Ha – különben” feltétel

Az 5. leckét folytatjuk tovább, de most már egy másik feltételes szerkezettel is megismerkedünk:



Ebben a leckében, ha sikerült ügyesen az autót a parkolóba juttatni, akkor dicsérjük meg magunkat, hogy milyen ügyesek voltunk, tehát írja ki, hogy: Ügyes voltál! Ehhez egymásba kell ágyaznunk a parancsokat.

Lépéseink:

- Választunk egy másik szereplőt, amely a mi esetünkben egy P betű, azaz parkoló (ezzel is jelezve, hogy jó helyen vagyunk).
- Ennek az új szereplőnek fogjuk az előbb leírt feladatot megadni, vagyis ha elérünk hozzá, akkor írja ki, hogy „Ügyes vagy!” különben, amíg oda nem érünk, villogjon.



Ezt a játékot különféleképpen is még tudjuk kicsit bonyolítani. Például:

1. Akadályokat tehetünk bele.
2. „Vakon” kell megkeresni a parkolót.
3. Időre is játszhatjuk.

Kész program: [parkolo](#)

7. lecke: Üzenetek

Az eddigi leckékben a szereplők akkor mozogtak, ha lenyomtuk egy billentyűt, vagy ha a zászló indításakor adtuk nekik feladatokat. Ebben a leckében akkor fognak valamit csinálni a szereplők, ha valamilyen üzenetet kapnak vagy üzenetben kapnak feladatot.

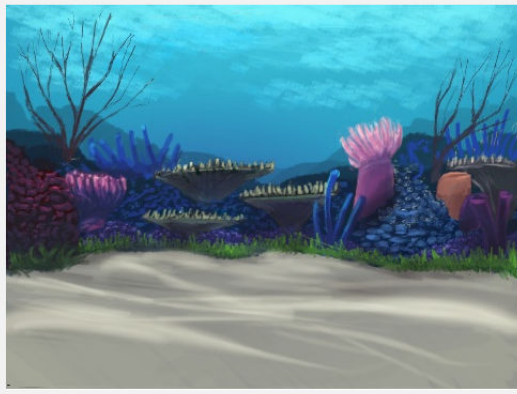
Ha több szereplős játékot szeretnénk készíteni, elengedhetetlen, hogy kommunikáljanak egymással a szereplők, erre valók az üzenetek.

De lássuk mit is jelent ez a gyakorlatban.

Ebben a leckében három halacska fog egymással kommunikálni üzenetben. Ha a fish1 halacskára kattintunk a fish2 halacska ránéz és közelebb úszik hozzá és köszön neki: Hello! Ha a fish3 halacskára kattintunk, akkor a fish2 halacska ránéz és közelebb úszik hozzá és köszön neki: Szia!

Lépéseink:

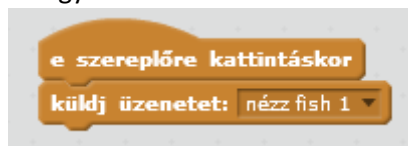
- Töltsünk be egy víz alatti háttérképet (underwater2).



- Töltsünk be 3 db halacskát, melyek különbözőek (fish1, fish2, fish3).



- Fish1 és fish3 halacskának ugyanaz a feladata:



Ha rájuk kattintunk abban az esetben a fish2 halacska ránéz és odaúszik hozzá, miközben köszön nekik.

- A fish2 halacsának a feladata egy kicsivel bonyolultabb:



És már kész is!

Kész program: [halacska](#)

8. lecke: Változók és véletlenszámok

Elértünk most már, hogy igazi izgalmas játékokat tudjunk készíteni. Ebben a játékban pontokat szerzünk miközben időre fogunk játszani.

A játékunkban egy denevér fog repkedni és közben labdákat fog dobálni. Nekünk az a feladatunk, hogy ezeket a labdákat elkapjuk (lehetőleg minél többet), mert ezekből lesznek a pontszámok. Erre 30 másodpercünk lesz.

Lépéseink:

- Válasszunk kettő tetszőleges háttérrel (woods and bench).

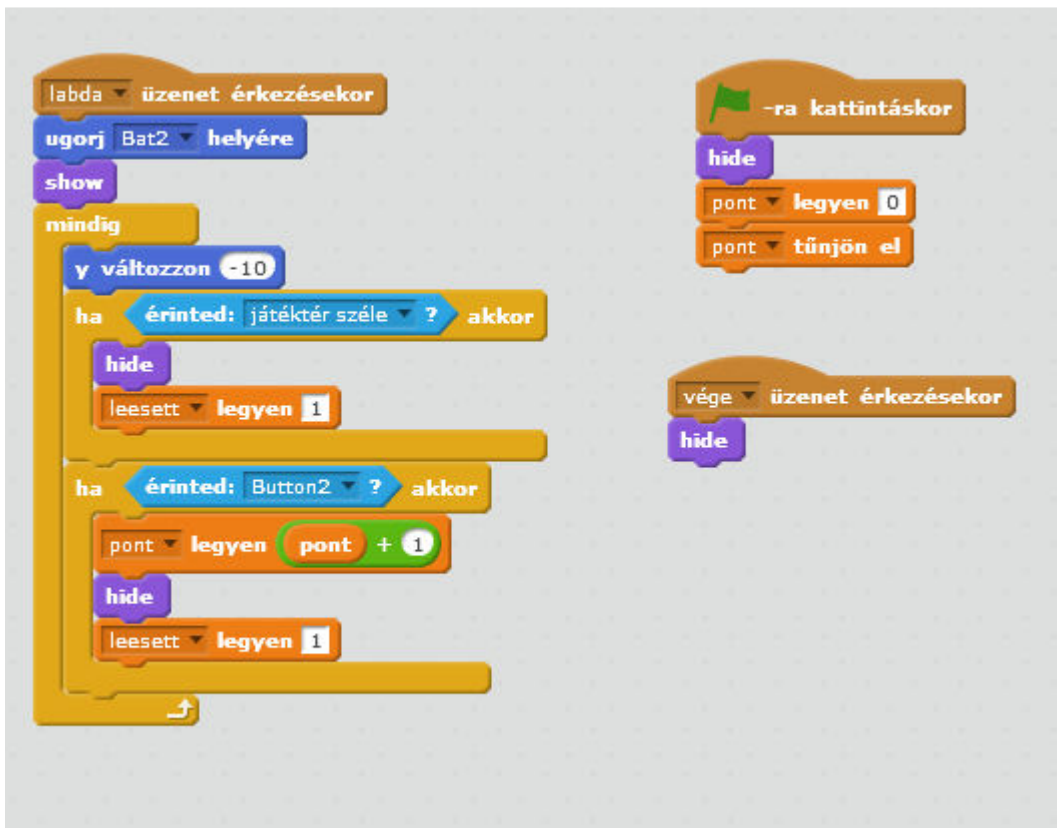


- Válasszunk szereplőket, egy olyat, ami tud repülni (bat2). Egy labdát és egy „kosarat”, amellyel elkapjuk a lepotyogó labdákat.
- Az első lépésünk, hogy ismertetjük a játékszabályokat. Ezt most úgy oldjuk meg, hogy egy másik háttér is jelenjen meg. Egy szereplő pedig bemutatja a játékszabályokat:

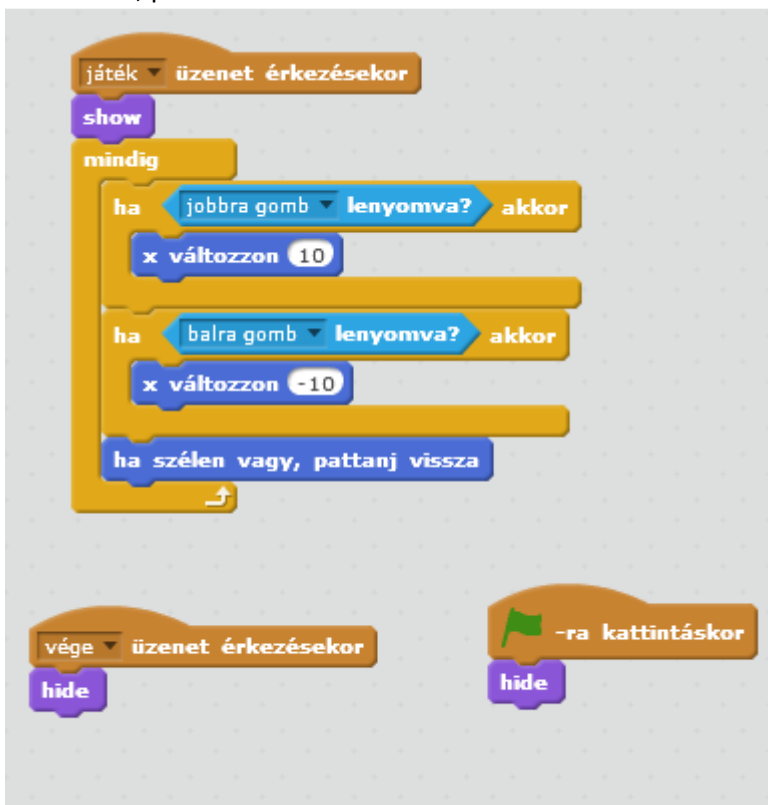


- A következő lépésünk, hogy a labda feladatait megadjuk. Ehhez változókra lesz szükségünk, hisz a lepotyogó labdákat számolni kell.
Mi is az a változó?
A változó olyan hely a számítógépen, ahol valamilyen mennyiséget tárolunk. Van neve és értéke. A mi esetünkben ez lesz a pontszám. A pontszám a játék indulásakor 0. Eggyel nő, ha elkap egy labdát a kosár.
Első lépésként beállítjuk, hogy a pontszám kezdetben legyen 0.

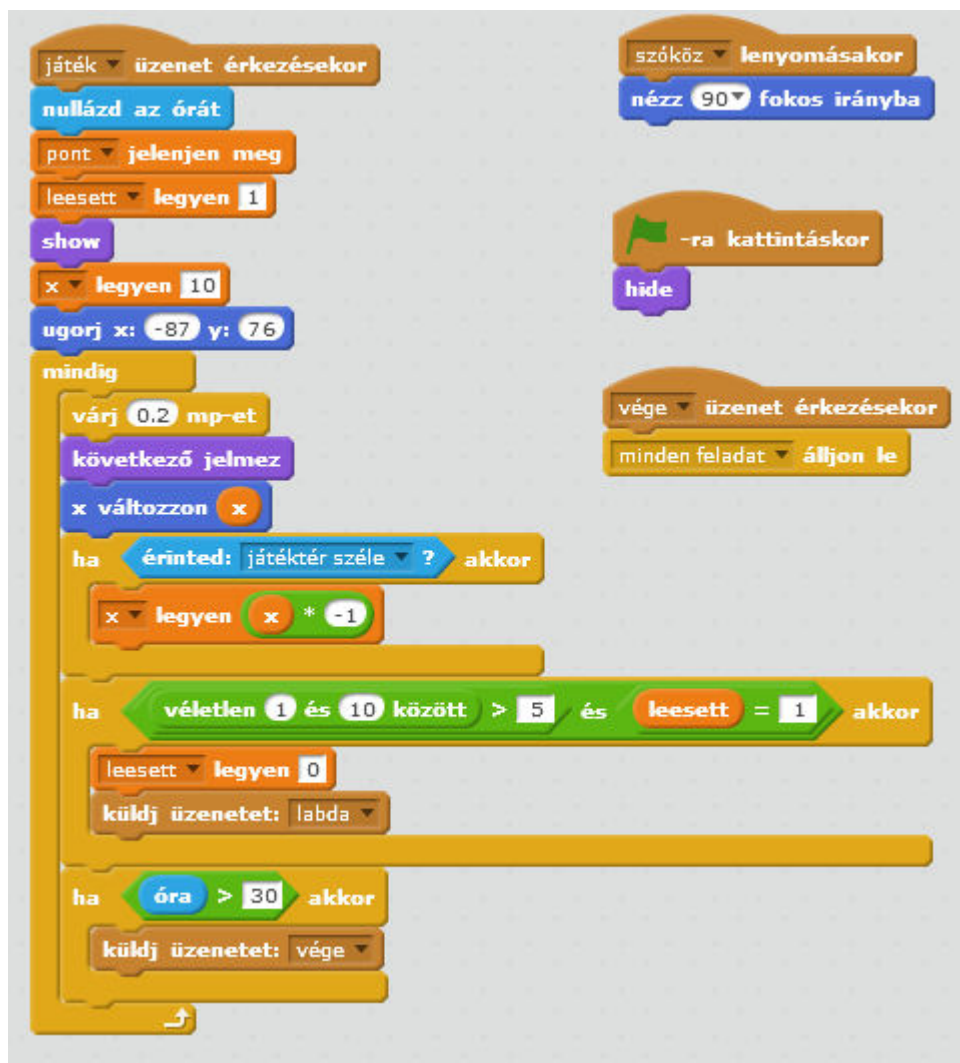
A labda feladatai, tehát:



- Ezután következik a kosár feladatainak megadása. Ebben a részben a már eddig megismert eszközöket, parancsokat kell használni.



- Most már csak a főszereplőnk, a denevér feladatait kell megadnunk, hogy repülés közben szálljon jobbra-balra és eközben véletlenszerűen dobálja a labdákat. Itt ebben a részben jelenik meg a véletlenszám. Ezzel tudjuk kiszámíthatatlanná tenni a labdák potyogását. A Scratch-ben a véletlenszámokat két szám között tudjuk megadni. Pl. 1 és 10 között, ami azt jelenti, hogy ez 1 és 10 közötti számot ad és mindegyiket egyforma eséllyel.



Kész program: [denever_jatek](#)

9. lecke: Kecske és káposzta, vagyis amikor már (majdnem) mindent tudunk 😊

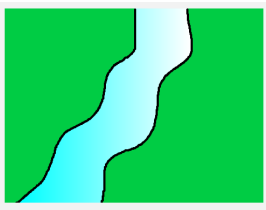
Elértünk most már ahhoz a részhez, amikor is már szinte mindenféle játékot meg tudunk valósítani.

Ebben a részben egy ismert fejtörőt fogunk megvalósítani, mégpedig azt, amikor egy csónakkal kell átvinni a folyó egyik oldaláról a másikra a játék szereplőit. Egy bányát, egy káposztát és egy farkast. Mindezt úgy, hogy senki ne egye meg a másikat. Vagyis a farkas nem maradhat egyedül a bányával, mert akkor meg fogja enni, de a bányát sem maradhat a káposztával, mert akkor ő is megeszi a káposztát.

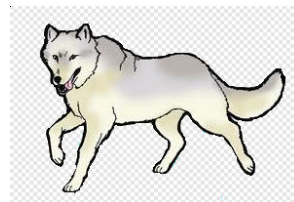
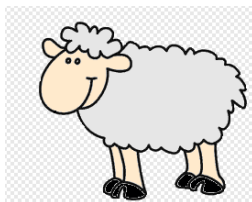
Akkor lássuk, hogyan is fog felépülni a játék.

Lépéseink:

- Először is meg kell rajzolnunk a háttérrel. Ebben az esetben egy folyót, aminek látszik mindkét partja.



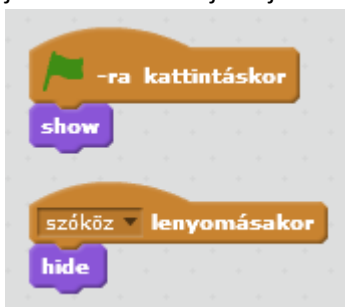
- Azután meg kell a szereplőinket alkotni. Tehát keresünk a beépített szereplők közül egy farkast, egy bányát és egy káposztát. (ha nincs ilyen, akkor vagy mi magunk rajzolunk 😊, vagy letöltünk az internetről egy képet).



- És még kell egy csónak is, ami szállítani fogja a szereplőket a folyó egyik partjáról a másikra.



- Minden játék egy játékmertetővel kezdődik, amelyben leírjuk, hogy mit is kell csinálni ebben a feladatban. Tehát leírjuk és aztán nincs más, mint azt a feladatot adni neki, hogy a játék indításakor jelenjen meg majd pedig a szóköz billentyű lenyomásakor pedig tűnjön el.

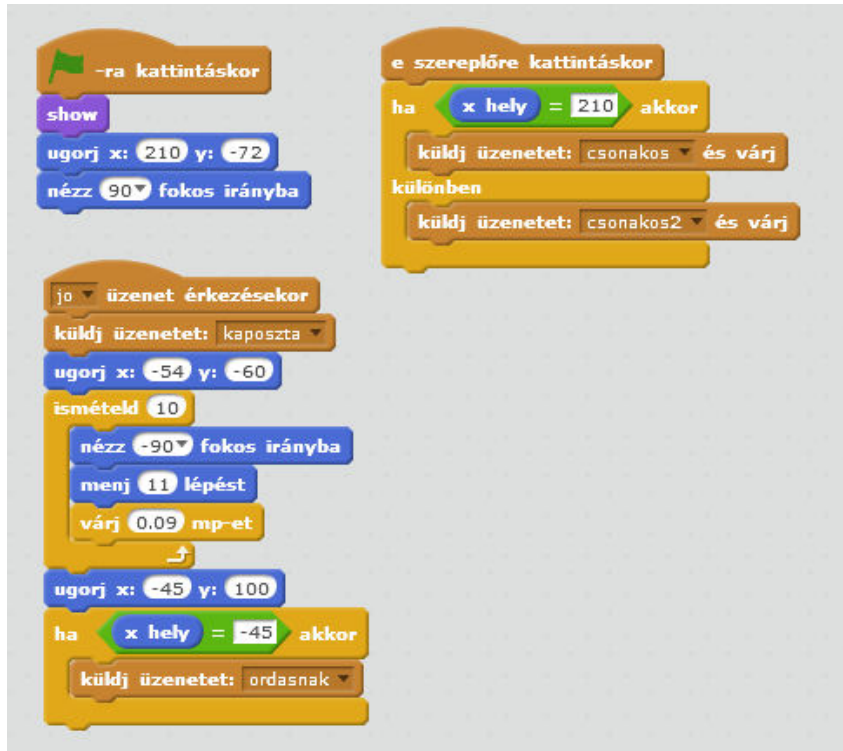


- A legtöbb feladata a csónakosnak lesz. Ő kapja a legtöbb utasítást, hisz neki kell tudnia, hogy melyik szereplő érkezésekor mit kell majd csinálnia. Tehát mind a három szereplőre való feladatlistát át kell gondolnunk.

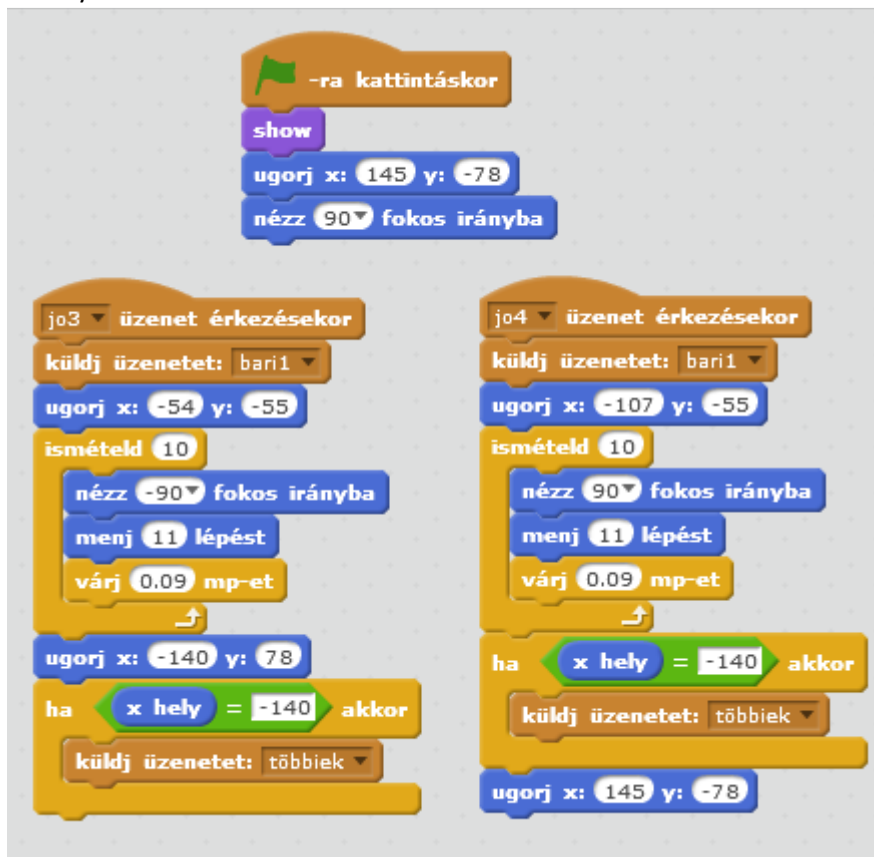
The image shows two Scratch code blocks. The first block, titled '-ra kattintáskor', starts with a green flag icon, followed by 'ugorj x: -25 y: -72' and 'nézz 90 fokos irányba'. The second block, titled 'kaposzta üzenet érkezésekor', starts with 'játszd le: Plunge', followed by a 'ha x hely = -25 akkor' condition. Inside this condition, it says 'nézz -90 fokos irányba', 'várj eddig: érinted: kaposzta?', and a loop of 10 iterations containing 'menj 11 lépést', 'várj 0.09 mp-et', and 'következő jelmez'. A 'különben' block follows, with 'nézz 90 fokos irányba', 'várj eddig: érinted: kaposzta?', and another loop of 10 iterations with the same movement and sound blocks. The block ends with 'minden hangot állíts le'.

The image shows a single Scratch code block titled 'farkas1 üzenet érkezésekor'. It starts with 'játszd le: Plunge', followed by a 'ha x hely = -25 akkor' condition. Inside this condition, it says 'nézz -90 fokos irányba', 'várj eddig: érinted: farkas1?', and a loop of 10 iterations containing 'menj 11 lépést', 'várj 0.09 mp-et', and 'következő jelmez'. A 'különben' block follows, with 'nézz 90 fokos irányba', 'várj eddig: érinted: farkas1?', and another loop of 10 iterations with the same movement and sound blocks. The block ends with 'minden hangot állíts le'.

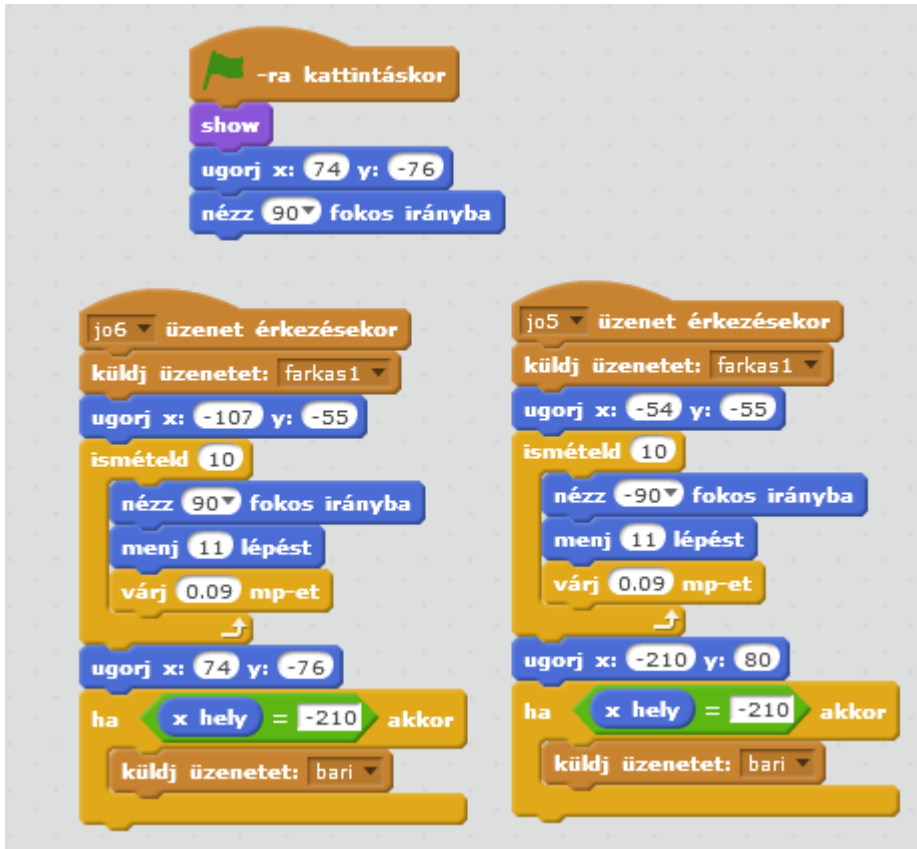
- Káposzta feladatai



- Bárány feladatai



- Farkas feladatai



- Utolsó lépésként feliratot készítünk, ha jól dolgoztunk, akkor írja ki a képernyőre, hogy: Sikerült! Ügyes vagy!



- Ha pedig elrontottuk, vagyis rossz sorrendben akartuk áthozni a szereplőket, akkor pedig azonnal legyen vége a játéknak és írja ki, hogy: Vesztettél!



Kész program: [kecske es kaposzta](#)

10. lecke: Keresd a párját!

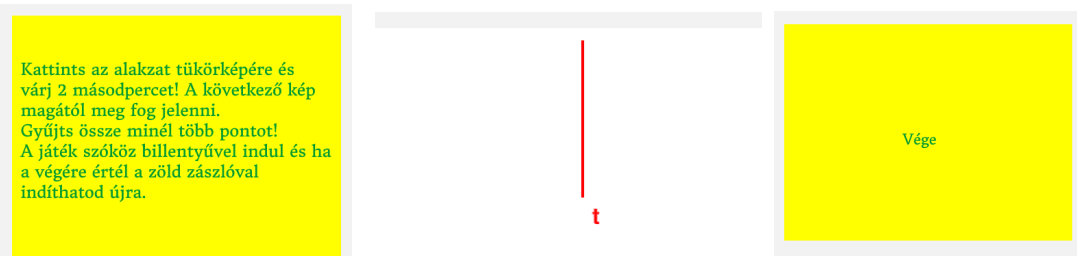
Ebben a leckében egy teljesen más típusú játékot fogunk elkészíteni, mint amit eddig készítettünk.

A „Keresd a párját!” játéknak az lesz a lényege, hogy a különböző formáknak meg kell keresni majd a hozzájuk tartozó tükörképét. Talán mindközül ez lesz a legösszetettebb feladat, amit nagyon át kell gondolni, hisz sok szereplője lesz. Viszont igaz, hogy sok szereplője lesz, de ezt úgy kell érteni, hogy egy szereplőnek lesz sok különböző jelmeze.

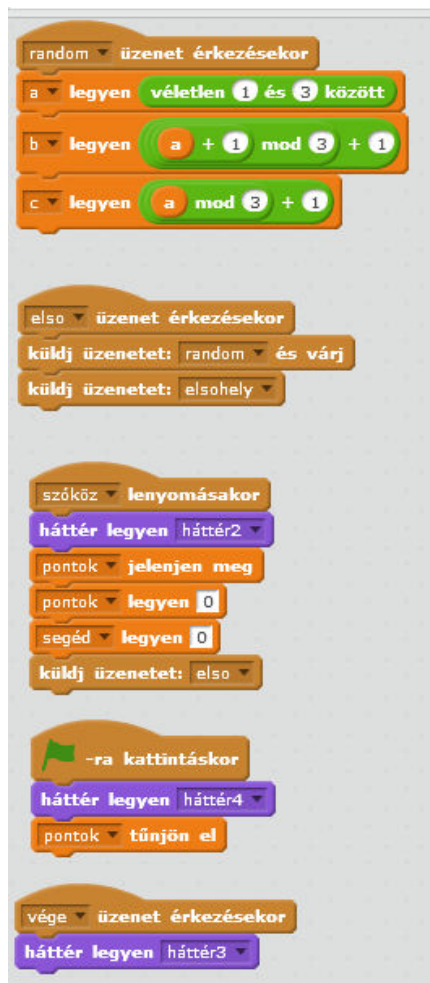
Lássuk ennek a játéknak a felépítését!

Lépéseink:

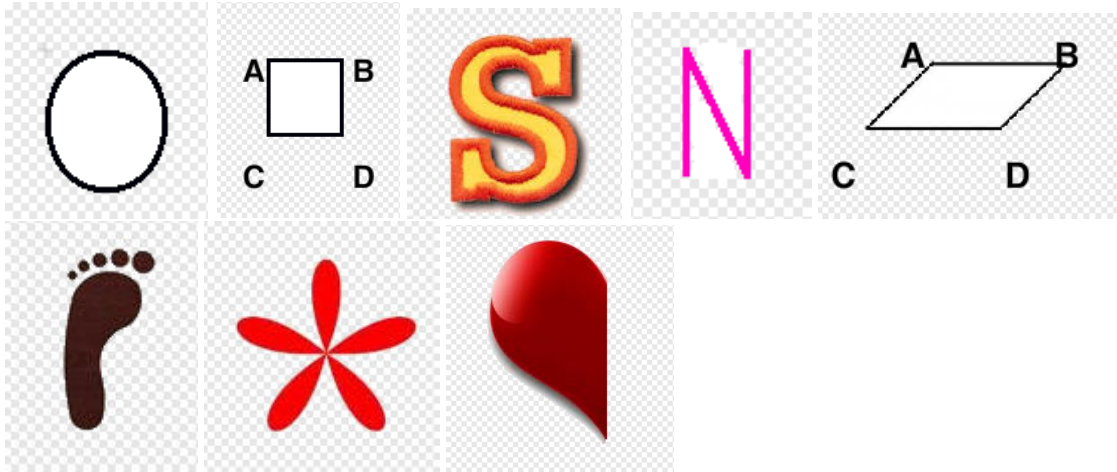
- Mint mindig először kezdjük a háttérak megajzolásával. Kell egy olyan háttér, ahol ismertette lesz a játékszabály, kell egy olyan, amin a tükrözés tengelye látható és kell egy olyan, ami a vége feliratot tartalmazza.



Mind a három háttér feladata a következő lesz:



- Most következhetnek a szereplők. A szereplőket a beépített készletből válogatjuk össze.



A szereplőknek most nem egyesével lesz külön-külön jelmeze, hanem egy feladathoz, „karakterhez” lesznek a különböző formák beállítva.
Minden szereplőnek négy – négy jelmeze van.

Ezek alapján lesz szereplő1, szereplő2, szereplő3, szereplő4, szereplő5.

(folytatás a következő oldalakon...)

- A szereplő1 feladata a következő lesz:

The image shows a Scratch script for a character named 'szereplő1'. The script is organized into several sections:

- Initial Setup (Első helyen):**
 - When the green flag is clicked, set the costume to 'jelmez1', show the character, and send a message 'válassz'.
- Message Received (Következő):**
 - When the message 'segéd = 1' is received, set the costume to 'jelmez2', send a random message, and send the message 'negyzet'.
 - When the message 'segéd = 2' is received, set the costume to 'jelmez3', send a random message, and send the message 'sbetű'.
 - When the message 'segéd = 3' is received, set the costume to 'jelmez4', send a random message, and send the message 'nbetű'.
 - When the message 'segéd = 4' is received, set the costume to 'k', send a random message, and send the message 'para'.
- Message Received (Further down):**
 - When the message 'segéd = 5' is received, set the costume to 'Lábnyom', send a random message, and send the message 'láb'.
 - When the message 'segéd = 6' is received, set the costume to 'szirom', send a random message, and send the message 'virag'.
 - When the message 'segéd = 7' is received, set the costume to 'sz', send a random message, and send the message 'szív'.
 - When the message 'segéd = 8' is received, send the message 'vége'.
- Click Event:**
 - When the character is clicked, hide the character.
- Message Received (End):**
 - When the message 'vége' is received, hide the character.

- A szereplő2 feladatai:

```

válassz üzenet érkezésekor
jelmez legyen jelmez1
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

-ra kattintáskor
hide

következő üzenet érkezésekor
hide

e szereplőre kattintáskor
küldj üzenetet: rossz2 és várj
segéd változzon 1
küldj üzenetet: következő

sbetű üzenet érkezésekor
jelmez legyen images
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

```

```

vége üzenet érkezésekor
hide

para üzenet érkezésekor
jelmez legyen k
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

láb üzenet érkezésekor
jelmez legyen Lábnyom
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

virag üzenet érkezésekor
jelmez legyen szírom
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

szív üzenet érkezésekor
jelmez legyen sz
show
ugorj x: -14 y: 150
csússz 1 mp-ig x: item a of x y: -100

```


- A szereplő3 feladatai:

válassz üzenet érkezésekor
 jelmez legyen jelmez1
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100
 -ra kattintáskor
 hide
 e szereplőre kattintáskor
 küldj üzenetet: rossz és várj
 segéd változzon 1
 küldj üzenetet: következő
 vége üzenet érkezésekor
 hide
 negyzet üzenet érkezésekor
 jelmez legyen jelmez2
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100
 sbetű üzenet érkezésekor
 jelmez legyen images
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100

nbetű üzenet érkezésekor
 jelmez legyen jelmez3
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100
 para üzenet érkezésekor
 jelmez legyen k
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100
 rossz üzenet érkezésekor
 küldj üzenetet: poz1
 várj 2 mp-et
 láb üzenet érkezésekor
 jelmez legyen cic
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100

virag üzenet érkezésekor
 jelmez legyen szírom
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100
 szív üzenet érkezésekor
 jelmez legyen sz
 show
 ugorj x: -14 y: 150
 csússz 1 mp-ig x: item b of x y: -100

- A szereplő4 feladatai:

The image displays a series of Scratch scripts for a character named 'szereplő4'. Each script is triggered by a 'üzenet érkezésekor' (when a message is received) event. The scripts are as follows:

- válassz**: jelmez legyen *jelmez1*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- következő**: hide.
- ra kattintáskor**: hide.
- e szereplőre kattintáskor**: küldj üzenetet: *jó* és várj, pontok változzon 1, segéd változzon 1, küldj üzenetet: *következő*.
- negyzet**: jelmez legyen *jelmez2*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- jó**: csússz 1 mp-ig x: 60 y: 74, várj 2 mp-et.
- vége**: hide.
- sbtű**: jelmez legyen *images*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- nbtű**: jelmez legyen *jelmez3*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- para**: jelmez legyen *k*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- láb**: jelmez legyen *Lábnym*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- virag**: jelmez legyen *szírom*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.
- szív**: jelmez legyen *sz*, show, ugorj x: -14 y: 150, csússz 1 mp-ig x: *item c of x* y: -100.

- A szereplő5 feladata, (ez a rossz megoldások esetén jelenik meg):



Kész program: [Keresd a parjat](#)