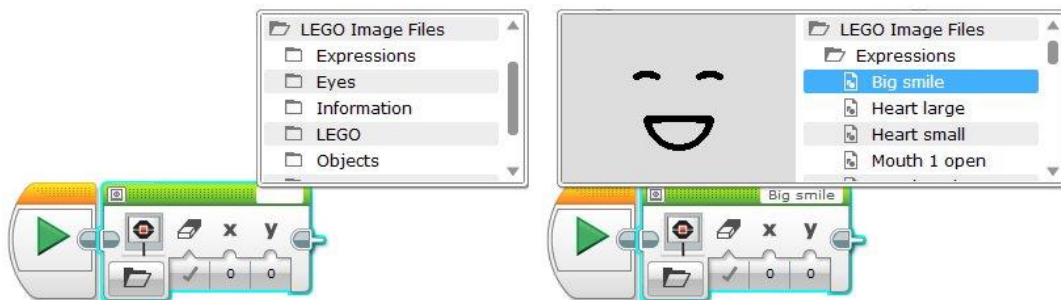


Dinamikus rajzok a robot képernyőjén

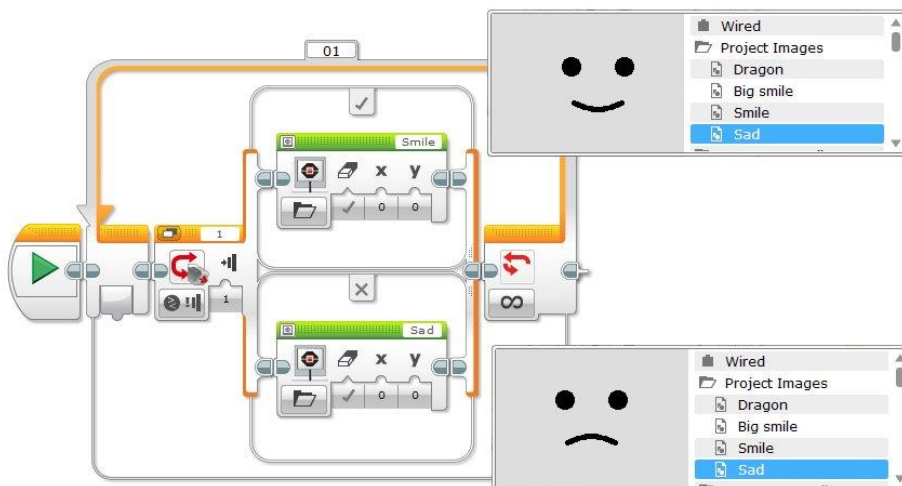
A képernyőre nem csak alakzatok és szöveg kerülhet, hanem egyszerű két színnel elkészített rajzok is. Alapértelmezésben sok előre elkészített piktogram, egyszerű grafika áll a rendelkezésünkre, amelyet a képernyőre rajzolhatunk. A korábban bemutatott rajzoló eszközzel magunk is készíthetünk ilyen *rbf* kiterjesztésű képeket.

A *Display* modul beillesztésével ha a működési módot *Image*-re állítjuk, akkor ezek a rajzok az ikon jobb felső sarkában látható szövegdobozra kattintva egy listából választhatók. Az előre elkészített rajzok témájukat tekintve mappákba rendezve jelennek meg a listán. Valamelyiket kiválasztva a képről egy villámnézetet is kapunk.



A használatot egy egyszerű, de látványos programmal mutatjuk be.

Írjon programot, amelyet végrehajtva a robot egy mosolygó smileyt rajzol a képernyőjére, ha az ütközésérzékelője be van nyomva, és egy szomorú smileyt, ha nincs benyomva! Mindezt kikapcsolásig ismétlje!

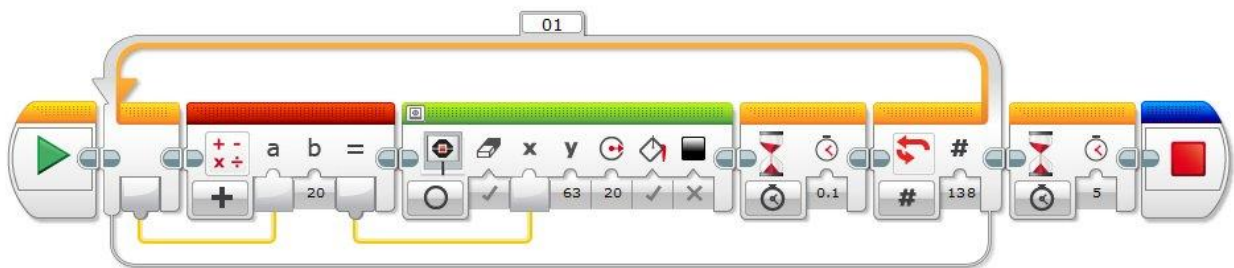


Összetettebb program esetén már komplexebben kihasználhatjuk a korábban tanult programozási eszközöket.

Írjon programot, amelyet végrehajtva a robot a képernyőjén vízszintesen mozgat egy 20 pixel sugarú fekete színnel kitöltött kört! A kör a bal oldali képernyőszéltől a jobb oldali képernyőszélig egyenletes sebességgel (programozói beavatkozás nélkül) haladjon, majd ott álljon meg!

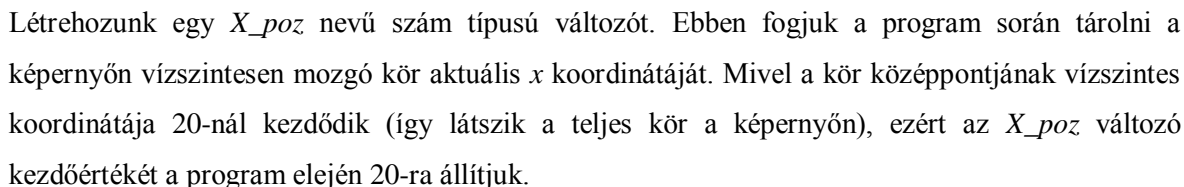
Mivel a kör sugara 20 pixel, ezért a középpontjának a koordinátája vízszintesen 19 és 157 között egyenletesen növekszik, tehát összesen legfeljebb 138 értéket vehet fel. A függőleges koordinátája pedig állandó, pl.: 63.

Egy 138-szor lefutó ciklust hozunk létre, amely minden egyes végrehajtás során eggyel növeli a vízszintes koordinátát. Erre alkalmas a ciklusváltozó, amelyhez 20-at adva (a kör középpontjának indulási pozíciója 20) már meg is kapjuk a kör középpontjának aktuális x koordinátáját. Ezt a paramétert adjuk át a képernyő modul bemeneti x paraméterének. A képernyőre rajzolás esetén mindig töröljük az előző ábrát, és a ciklus után várakozunk 5 mp-ig a program befejezése előtt. Mivel a végrehajtás nagyon gyors, ezért úgy lassítunk a kör mozgásán, hogy 0,1 mp-ig várakozunk minden egyes rajzolás után.

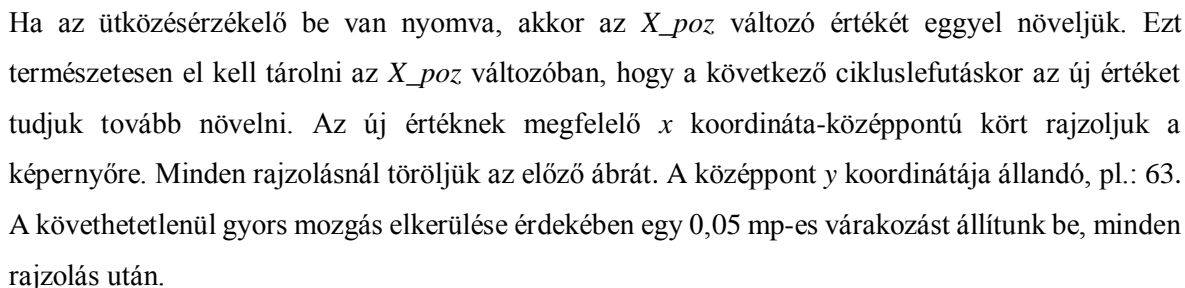


Természetesen a képernyőn történő mozgást egyéb szenzorokkal is vezérelhetjük. A következő program mutat erre ötletet.

Írjon programot, amelyet végrehajtva a robot ütközésérzékelőjének benyomásával szabályozható a képernyőn megjelenő 20 pixel sugarú fekete színnel kitöltött kör vízszintes mozgása! Ha az ütközésérzékelő be van nyomva, akkor a kör x koordinátája folyamatosan 1 pixelenként növekszik (így a kör balról jobbra mozog). Ha a kör elérte a képernyő szélét (teljesen eltűnt a képernyőről), akkor újra jelenjen meg bal oldalon és folytassa a vízszintes balról jobbra mozgást. Ha az ütközésérzékelő nincs benyomva, akkor álljon a kör az aktuális pozícióján. Mindezt kikapcsolásig ismételje!



01



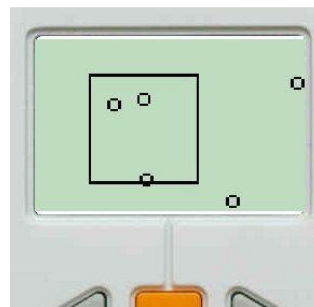
Ezzel az elágazásnak és a ciklusnak is vége.

A programot három részre darabolva mutattuk be, de természetesen ezt csak a könnyebb áttekinthetőség kedvéért tettük.

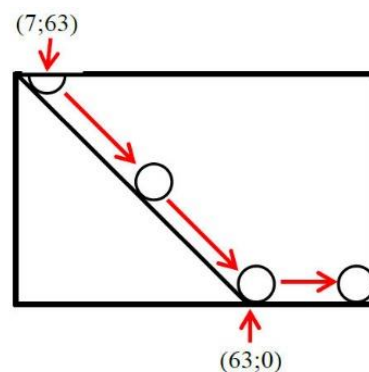
A program továbbfejlesztéseként a függőleges mozgás vezérlésére használhatunk egy másik ütközésérzékelőt, így a kör a képernyőn tetszőleges pozícióba mozgatható.

Feladatok

1. Írjon programot, amelyet a robot végrehajtva képernyőjére rajzol egy 40 pixel oldalú négyzetet, amely bal alsó sarkának koordinátái (20;10)! A program futása alatt a négyzet végig látszódjon a képernyőn! A robot sorsoljon véletlenszerűen két számot, amely koordinátákhoz tartozó pont körül a képernyőre rajzol egy 2 pixel sugarú kört. A sorsolást úgy végezze, hogy a rajzolandó körvonalnak legalább a negyed része látsszon a képernyőn. Ha a sorsolt pont a négyzeten belül van, akkor sípoljon egyet, egyébként ne. 1 másodperc várakozás után ismétlje a sorsolást, összesen 5-ször. Valamennyi pont maradjon a képernyőn! A program befejezése előtt a robot várakozzon 5 mp-ig!



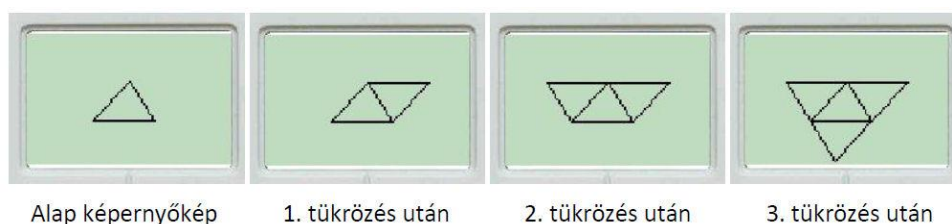
2. Írjon programot, amelyet végrehajtva a robot egy 5 pixel sugarú kört mozgat megadott pályán! A kör útvonala egy 45°-os lejtővel kezdődik, majd a lejtő aljától vízszintesen folytatódik. A kör nem metszheti át a lejtőt szimbolizáló szakaszt és a vízszintes szakaszon is teljes terjedelmében látszania kell. A körvonal képe és a lejtő, illetve a vízszintes szakaszon a képernyő aktív alsó széle között ne legyen szemmel látható hézag. (Tehát a kör „legurul” a lejtőn, majd a vízszintes szakaszon halad tovább, eltekintve a gyorsulástól, egyenletes sebességgel.) Az útvonalat értelmezi az ábra.



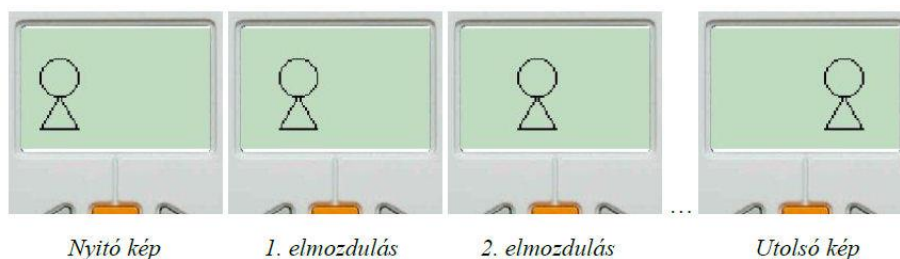
A kör középpontjának kezdő koordinátája: (7;63). A lejtő egyik végpontjának koordinátái: (63;0), és a lejtő 45°-os. A kör egyenletes sebességgel haladjon végig az útvonalon, és a képernyő szélét elérve álljon meg! A program a befejezés előtt 10 másodpercig várakozzon!

3. Írjon programot, amelyet végrehajtva a robot a képernyőjére rajzol egy háromszöget! A háromszög csúcsának koordinátái: (32;21); (62;21); (50;40). Az ütközésérzékelő benyomására jelenjen meg a képernyőn az eredeti háromszög egyik oldalfelező pontjára vonatkozó tükörképi háromszög is! Az ütközésérzékelő további megnyomásaira jelenjen meg egy másik oldalfelező pontra vonatkozó tükörkép, valamint a harmadik oldalfelező pontra vonatkozó tükörkép is! Minden háromszög látszódjon egy időben képernyőn és a program ütközésérzékelő megnyomására álljon le!

A robot képernyőjén látszó kép:

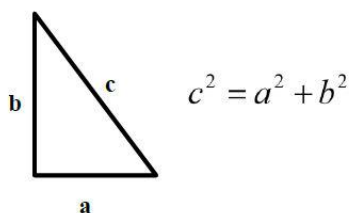


4. Írjon programot, amelyet végrehajtva a robot a képernyőjére rajzol egy szabályos háromszögből és egy körből álló ábrát a képnek megfelelően! A háromszög bal alsó csúcsának koordinátái: (10;10), az oldalának hossza 20 pixel. A kör sugara 10 pixel. A kör érinti a háromszög felső csúcsát és középpontja a háromszög szimmetria tengelyére esik. Ütközésérzékelő benyomására a teljes ábrát (háromszög és kör) tolja el 10 pixellel jobbra úgy, hogy a régi ábra már ne látszódjon a képernyőn. Mindezt hatszor ismételje! Az új ábra mindig az ütközésérzékelő megnyomására jelenjen meg! A hatodik rajz után a program várjon 5 másodpercet, majd álljon le!



Matematikai segítség a háromszög csúcskoordinátáinak meghatározásához (Pitagorasz tétele):

Derékszögű háromszögben az átfogó hosszának négyzete egyenlő a befogók hosszainak négyzetösszegével.

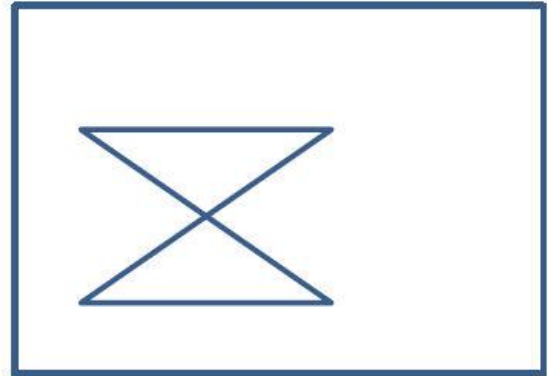


5. Írjon pont összekötő játékot! Egy állományban az összekötendő pontok x és y koordinátáit tároljuk. Olvasd be a pontok koordinátáit, és a másodiktól kezdve minden pontot köss hozzá egy szakasszal az előző ponthoz! Minden pont a robot képernyőjének belső pontja. Az utolsó ponthoz kösd hozzá az elsőnek megadottat. A képernyőre rajzolt kép ütközésérzékelő benyomásáig maradjon meg.

pl.:

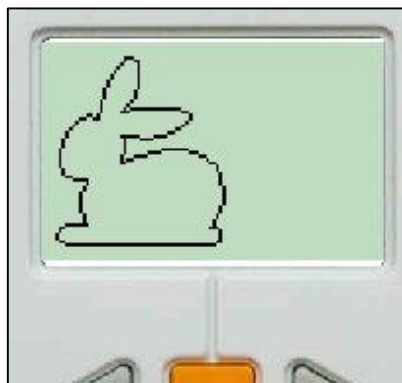
10
10
50
50
10
50
50
10

eredménye:



A *kep.txt* (NXT-G) illetve *kep.rtf* (EV3-G) fájlban összesen 65 pont koordinátái találhatók x; y sorrendben. Minden koordináta külön sorban szerepel.

A fájlban található pontokból kirajzolódó kép:



A fájlkezeléshez segítség található a Kreatív robotika című könyveben, amely letölthető: http://www.hdidakt.hu/adat/dw_anyagok/dw_74.pdf címről.