Itt a piros, hol a piros

A projekt célja

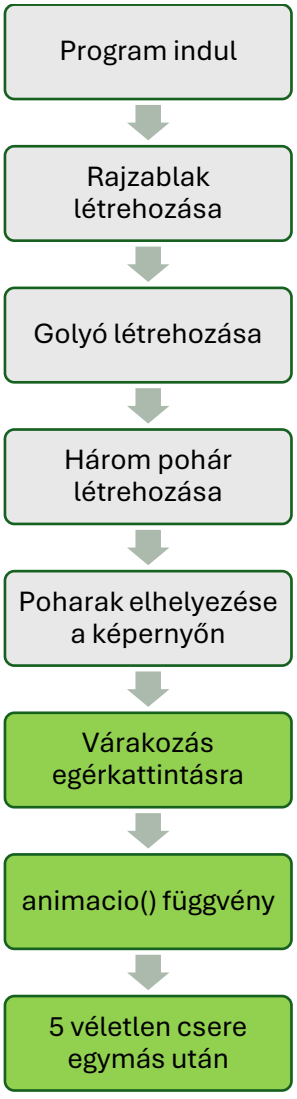
Itt a piros, hol a piros animáció készítése:

A projektben egy „Itt a piros, hol a piros” animációt készítünk el közösen, amelyben három pohár közül az egyikben van egy piros golyó, a poharakat véletlenszerűen cserélgetjük, a piros golyó követi annak a pohárnak a mozgását, amelyben benne van.

A projekt elkészítése során használt fogalmak:

- ☑ listák
- ☑ gif betöltése
- ☑ teknőcök animálása (onclick + callback)
- ☑ folyamatok összehangolása
- ☑ véletlenszám generálás

A program felépítése



Projekt előkészítése

Rajzlap

```
rajzlap = turtle.Screen()
rajzlap.tracer(0) # Automatikus frissítés kikapcsolása
rajzlap.addshape("pohar.gif") # Saját alak betöltése
```

Golyó

```
golyo = turtle.Turtle()
golyo.shape("circle")
golyo.color("red")
golyo.penup()
```

Poharak

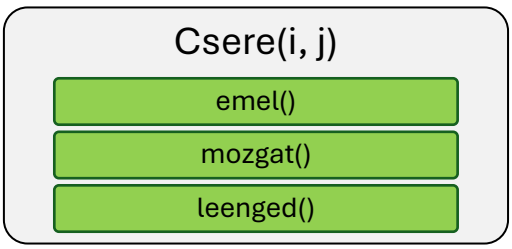
```
for _ in range(3):
   p = turtle.Turtle()
   p.penup()
   p.shape("pohar.gif")
   poharak.append(p)
# Kezdő pozíciók
poziciok = [-100, 0, 100]
for i in range(3):
   poharak[i].setx(poziciok[i])
rajzlap.update() # Manuális képernyőfrissítés
```

Golyó helyének frissítése

```
# A golyó kezdőpozíciója
golyos_pohar = 1
# A golyót mindig a megfelelő pohár alá helyezzük
def frissit_golyo():
     golyo.setx(poharak[golyos_pohar].xcor())
```

Animáció alapjai

Csere



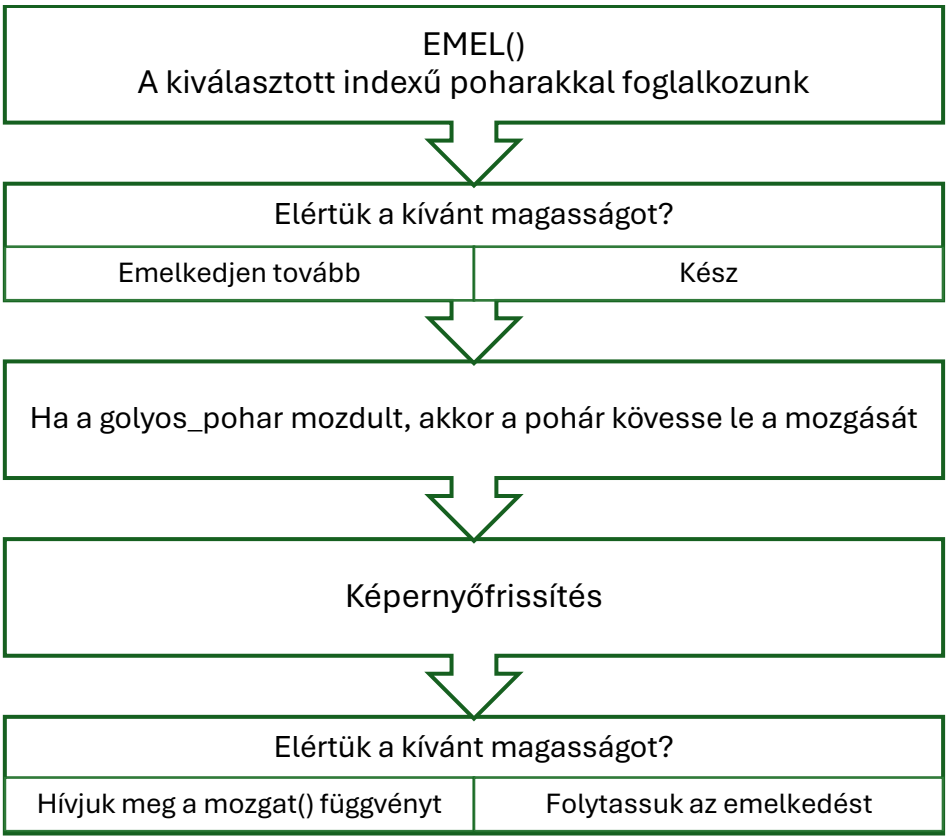
A `csere` függvény a paraméterként kapott két poharat animálva megcseréli. Az animáció 3 lépésből áll:

1. a poharak felemelkedéséből,
2. egymás helyére mozgatásából,
3. majd a leengedésből.

Ezeket lokális függvények segítségével valósítjuk meg. A `csere` függvény elején beállítjuk mindkét pohár célhelyét, azaz leendő `x` koordiinátáját. A csere függvényen belül csak az emel függvényt kell elindítanunk, a többi függvényt úgy készítjük el, hogy meghívják sorban egymást, ha befejeződtek.

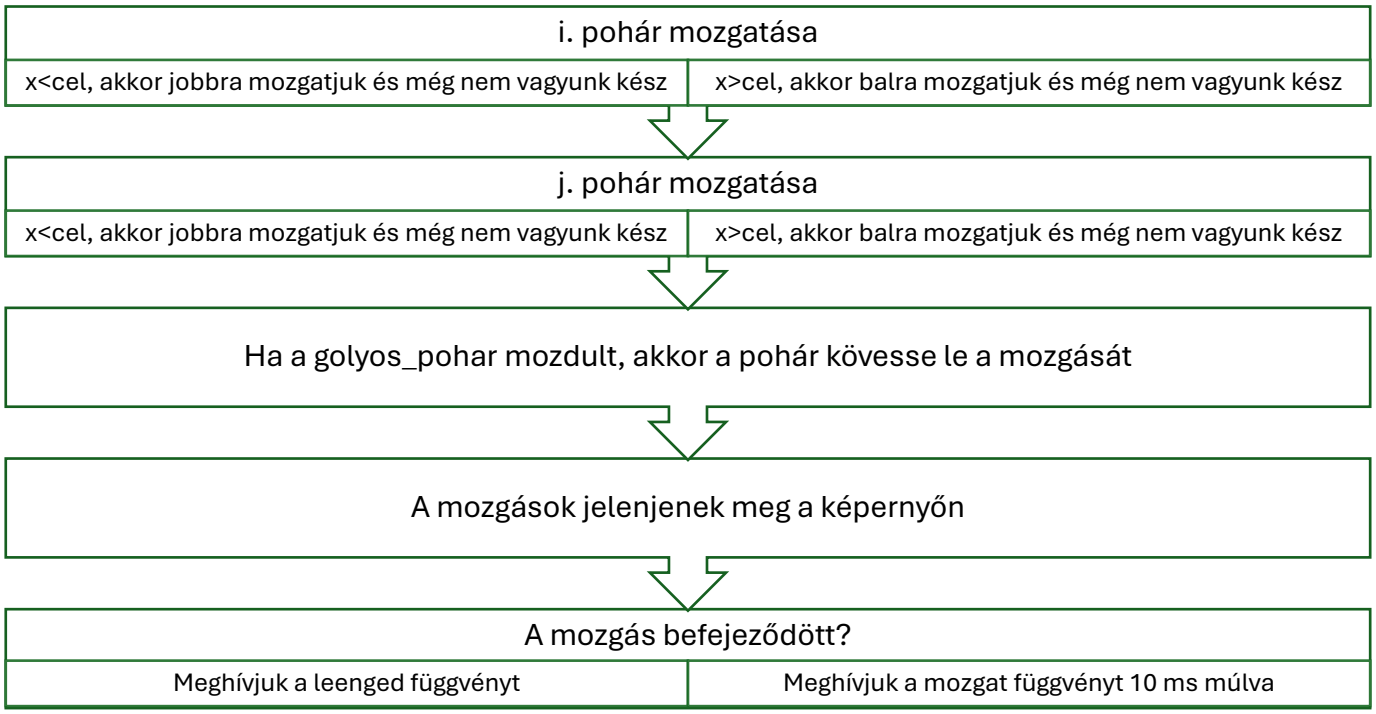
Emel

Az `emel` függvény a `csere` paramétereként megadott `i, j` indexű poharak mindegyikét felemeli a kívánt magasságba. Mivel az `emel()` függvény a csere függvény lokális függvénye, ezért rálát az `i, j` paraméterekre.



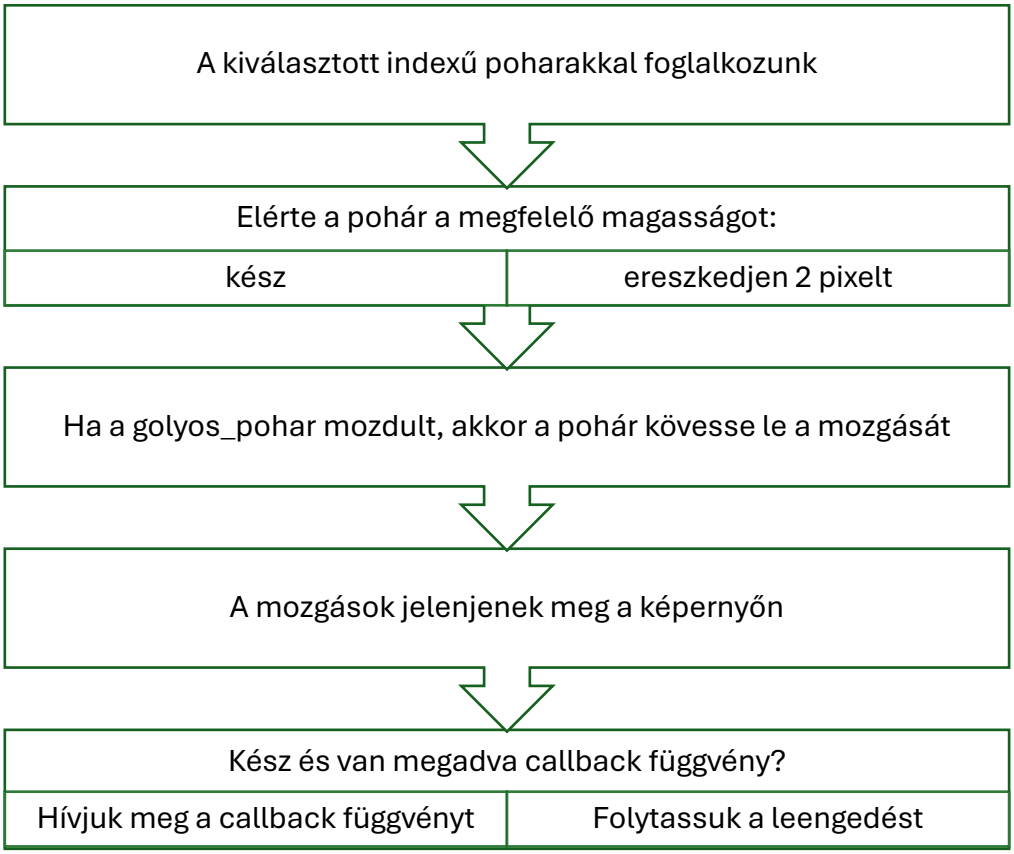
Mozgat

A kiválasztott, felemelt poharak helyet cserélnek, a két kiválasztott pohár egyszerre mozog. Ha a `golyos_pohar` mozog, akkor a golyó mozogjon vele együtt.



Leenged

A két megcserélt poharat leengedi a golyóval együtt.



A callback függvény fogalma

A callback (visszahívási) függvény olyan függvény, amelyet paraméterként adunk át egy másik függvénynek vagy eseménykezelőnek, és amely egy meghatározott esemény bekövetkezésekor vagy a program futásának egy későbbi pontján automatikusan meghívásra kerül.

Példák

Például egy teknőc onclick eseményéhez is egy callback függvénnyel tudunk hozzárendelni műveletet:

```
import turtle

t = turtle.Turtle()

def kattintas(x, y):
    turtle.goto(x, y)
    turtle.dot(10)

turtle.onclick(kattintas)
```

Különböző callbackek is megadhatóak ugyanahhoz a függvényhez, az alábbi példában a `szamol()` működése attól függ, melyik callbacket adjuk át neki.:

```
def szamol(a, b, muvelet):
    print(muvelet(a, b))

def osszead(x, y):
    return x + y
```

```
def szoroz(x, y):
    return x * y

szamol(6, 4, osszead)
szamol(6, 4, szoroz)
```

A következő cserére lépés

```
# Lehetséges cserék:
parok = [(0, 1), (1, 2), (0, 2)]

# Következő csere pár kiválasztása, ha még kell
def kovetkezo(db):
    # Ha db = 0 álljon le
    if db == 0:
        return
    # Véletlenszerűen válasszunk egy párt a parok közül
    i, j = random.choice(parok)
    # Cseréljük meg a kiválasztott poharakat, majd indítsuk el a következő cserét
    csere(i, j, lambda: kovetkezo(db - 1))
```

A `callback` ebben a példában a `lambda: kovetkezo(db - 1)` függvény. Ez egy névtelen függvény, amelynek egyetlen feladata, hogy a megfelelő időpontban meghívja a `kovetkezo(db - 1)` függvényt. A `lambda` használata azért szükséges, mert a `callback` paraméternek egy **függvényt** kell átadni, nem pedig egy **függvényhívás eredményét**. Ha a kódban ezt írnánk:

```
csere(i, j, kovetkezo(db - 1))
```

akkor a `kovetkezo(db - 1)` azonnal végrehajtódna, és nem a csere befejezése után. A `lambda` viszont "becsomagolja" ezt a hívást, így a `csere()` függvény csak akkor hajtja végre, amikor a csere animációja vagy művelete befejeződött.

Eseménykezelés

A golyóra való egérkattintás indítja a keverést:

```
def animacio(x, y):
    kovetkezo(5)

golyo.onclick(animáció)
```

A program működési folyamata