

5.3 Robot changyutkich to'siqqa duch kelsa nima bo'ladi? - Holat

Bu uchinchi, "Robot changyutgich" misoli. Bu misolda biz ilgari foydalanmagan Actor ob'yektlarining qo'shimcha usullari (harakatlari) keltirilgan, ammo oldingi boblarda o'rganilgan asosiy tamoyillar unchalik o'zganligi sababli, bu juda qiyin bo'lmasligi kerak.

```
{% code lineNumbers="true" %}
```

```
from pgzhelper import *

WIDTH = 960
HEIGHT = 540

robot = Actor('cleaner', (WIDTH / 2, HEIGHT / 2))
robot.scale = 0.3
robot.angle = 90
robot.brush_init((WIDTH, HEIGHT), 50, 'white')

def draw():
    screen.blit('pol', (0, 0))
    robot.brush_draw()
    robot.draw()

def update():
    robot.move_forward(10)

    if robot.top < 0 or robot.right > WIDTH \
        or robot.bottom > HEIGHT or robot.left < 0:
        robot.move_back(10)
        robot.angle += 133
```

```
{% endcode %}
```

Birinchidan, 3-4 qatorlarda ekran o'lchami mavjud o'lchamdan ikki barobarga oshirildi (480 kenglik, 270 balandlik). Buning alohida sababi yo'q, bu faqat ijro natijasi ekranini kattaroq hajmda ko'rish uchundir.

6-qatorda robot changyutgichni ekranning o'rtasiga joylashtirish uchun markaziy koordinata gorizonta va vertikal uzunliklarning yarmiga o'rnatildi.

8-qatorda robot changyutgich ob'ekti birinchi marta ishga tushirilganda ekranda yuqoriga qarab harakatlanishi uchun uning boshlang'ich burchagi 90 darajaga o'rnatiladi.

9-qatordan robot changyutgichning harakat izlarini bo'yashning asosiy sozlamalari **brush_init** funksiyasi (metodi) orqali amalga oshiriladi. Uchta argument talab qilinadi: cho'tka bilan bo'yalgan umumiy maydon, qalinligi va rangi.

12-qatorda men nihoyat birinchi marta o'yin fonini chizish uchun kodni qo'lladim. **Ekrani** deb nomlangan oldindan o'rnatilgan obyektga turli xil funksiyalar (metodlar) mavjud bo'lib, **ular o'yin ekraniga ishora qiladi**

([batafsil ma'lumot](#) uchun PygameZero qo'llanmasiga qarang) va blit metodi ulardan biri bo'lib, tasvirni to'g'ridan-to'g'ri ekranga chiqaradi va bizning kodda pol tasvirini chiqaradi.

16-qatorda paydo bo'lgan **update qayta chaqirish funktsiyasi o'yin tsiklini tashkil etuvchi asosiy funktsiyadir. Bu o'yinda doimiy ravishda o'zgarib turadigan narsalar (masalan, Actor obyekti harakatlanayotganda) yoki doimiy tekshirilishi kerak bo'lgan narsalar (masalan, qaysi klaviatura yoki sichqoncha tugmasi bosilgan) uchun kod yozadigan qismdir.** Shuning uchun robot changyutgich har safar 10 o'lchamdagi harakatni takrorlashi uchun 17-qatorda **move_forward(10)** ishlatilgan.

Va nihoyat, 19-qatordan so'ng, robot changyutgich o'yin ekranining oxiriga yetganda nima bo'lishi kerakligini tavsiflovchi shart bilan kod keladi. Bu Entry muhitida odatiy kodlashdan biroz farq qiladi. *Entry-da o'yin ekranining chegaralari "devor" deb nomlangan va obyektlar sifatida nazarga olingan. Bu devor chegarasiga tegganligini (chiqib ketganini) aniqlash uchun obyektlar orasidagi to'qnashuvni tekshirishdan foydalanilgan. Biroq, Pygame Zero-da ekran chegaralari obyektlar emas. Shuning uchun ekran chegarasiga tegganligini aniqlash uchun doimiy ravishda harakatlanadigan robot changyutgichning hozirgi holati (joylashuvi) orqali amalga oshiriladi.*

Masalan, robot.top qiymatining 0 dan kichik bo'lishi robot changyutgich obyektlarining o'yin ekrani yuqori chegarasidan chiqib ketayotganini anglatadi. Shu bilan birga, robot.right qiymatining ekran kengligidan kattalashishi o'yin ekrani o'ng chetiga yetib borganini bildiradi. Ushbu holatda, robotni biroz (10 birlik miqdorida) orqaga qaytarib, keyin yo'nalishni o'zgartirishga urinayotganini yodda tutish kerak. Bunday algoritmnining zarurati shundaki, ekranning chetiga yetganda darhol yo'nalishni o'zgartirishga urinish, ba'zan yo'nalish o'zgartirilgan paytda ham obyektlar hali ham ekranning chekkasida bo'lib qolishi mumkin. Natijada, robot doim bir joyda aylanib qolib, boshqa yo'nalishda harakat qila olmaydi. Buning oldini olish uchun robotni bir oz orqaga qaytarib, keyin yo'nalishni o'zgartirishga harakat qilinadi.

Ma'lumot uchun, 19-qator oxirida joylashgan '\n' belgisiga qiziqayotganlar bo'lishi mumkin. 19~20-qatorlarga bo'lingan kod aslida bitta uzun qator sifatida 19-qatorga joylashishi kerak edi. Ammo muallif bu kodni ikki qatorga ajratdi, bu esa kodni o'qishda tez va intuitiv tarzda tushunish imkonini berish, ya'ni kodni o'qish qulayligini oshirish uchun qilingan. Ammo, **kodlashda, agar bitta uzun qatorni bir necha qatorlarga ajratish kerak bo'lsa, ajratilgan qatorning oxirida '\n' belgisi qo'yilishi shart. Bu belgidan foydalanish ajratilgan qator hali tugamaganligini va keyingi qator bilan bog'langanligini bildiradi.**

Nihoyat, biz robot changyutgichning yakuniy natijalarini ko'rib chiqish orqali ushbu bobni yakunlaymiz.



