

## Pythonda "Iloncha" o'yinini yaratamiz (1-qism) – Praktika

Bugun biz dunyoda bilmagan odam bo'lmagan klassik o'yin - "**Snake**" o'yinini yangi ko'rinishda yaratishga urinib ko'ramiz. Buni barchasi hammaga tushunarli bo'lishi uchun mavzuni bir necha qismlarga bo'lib ko'rsatishga qaror qildim. O'yinii amalga oshirishda **PyGame** modulidan foydalanamiz.



Bugungi qismda biz quyidagi topshiriqlarni bajaramiz:

1. O'yin oynasini yaratamiz
2. Oyna orqasiga rang berib, asosiy element - Ilonni oynada joylashtiramiz.
3. So'ngra Ilonning harakatlanishini ta'minlaymiz.
4. Ilon oyna chegarasiga kelganida dasturni yakunlaymiz.

Bunda ko'pchilikka biroz murakkab ko'rinuvchi **OYD (Obyektga-Yo'naltirilgan-Dasturlash)**dan foydalanamiz. **OYM** qiyin ko'rinishi mumkin, lekin bu usul aynan dasturchining yukini yengillatish uchun yaratilgan.

Maqolada faqatgina **Python**da o'yin yaratishdan tashqari, **OYD** haqida ham qisqacha o'rganib chiqamiz.

Yuqoridagi vazifalarni bajarish uchun 2 ta obyekt yaratamiz:

1. **Game** - unda o'yin logikasini kiritamiz
2. **Snake** - Ilonning ko'rinishi va harakatlanishini taminlaydi

Bugun faqat shularni amalga oshiramiz va bunda avval bilmagan narsalarni o'rganasiz degan umiddaman!  
:)

# Boshladik

## 1 - Bo'lim

Kod bilan ishlash va o'qish osonroq bo'lishi uchun har bir obyektни alohida faylga yozamiz, ya'ni **Game** obyektini - [main.py](#) nomli faylga (asosiy fayl bo'lgani uchun), **Snake** obyektini uchun [snake.py](#) faylida yozamiz.

Birinchi bo'lib [main.py](#) faylida kerakli modullarni ulaymiz:

```
import pygame
import sys
from pygame.locals import *
import tkinter as tk
import os
```

Har biri nima uchun kerakligini tushuntiraman:

**pygame** - bu Python dasturlash tilida yozilgan, SDL kutubxonasi asosida yozilgan, barcha platformalarda ishlaydigan va o'yin yoki multimediali interfeysli dasturlar yaratishga yo'naltirilgan modullar to'plami.

**sys** - bu Pythonning asosiy modullaridan biri bo'lib, interpretator bilan ishlashga yordam beradi. (biz dasturni yakunlash, ya'ni o'chirish uchun foydalanamiz)

**pygame.locals** - bu pygaming locals moduli bo'lib, maxsus konstantlar, elementni oyna joylashtirish va ranglar bilan ishlashda foydalanuvchi modul.

**tkinter** - bu turli interfeysli dasturlar yaratilishda foydalaniladi. (biz dastur ish tushganda dastur oynasini ekran markazida joylashtirish uchun foydalanamiz)

**os** - bu operatsion sistema bilan ishlashga yordam beruvchi modul.

### 1-qadam: Game obyektini yaratamiz:

Bu uchun [main.py](#) fayliga:

```
class Game:
    deb yozamiz.
```

### 2-qadam: So'ngra kerakli kortejlar va o'zgaruvchilarni yozamiz (tushuntirishlar bilan):

```
SIZE = WIDTH, HEIGHT = (800, 500) # oyna o'lchami
WHITE = (255, 255, 255) # oq rangning RGB formatda
BACK = (15, 15, 15) # orqa fon rangi RGB formatda
RED = (255, 0, 0) # qizil rang RGB formatda
GREEN = (0, 255, 0) # yashil rang RGB formatda
fps = 20 # 1 soniyada nechta kadr tasvirlanishi yoki Ilon tezligi
```

**3-qadam: Game obyektini initsiyalizatsiya (gavdalantiramiz) qilamiz. Ya'ni obyektga beriladigan parametrlarni kerakli o'zgaruvchiga joylashtiramiz va kodda ishlatadigan o'zgaruvchi va boshqa kerakli parametrlarni shu qismda yaratib olamiz.**

Bu uchun:

```
def __init__(self):
    os.environ['SDL_VIDEO_WINDOW_POS'] = f'{(tk.Tk().winfo_screenwidth() -
self.WIDTH) // 2}, {
                                                                    f'{(tk.Tk().winfo_screenheight() -
self.HEIGHT) // 2}'
```

```

pygame.mixer.pre_init(44100, -16, 2, 2048)
pygame.mixer.init()
pygame.init()
pygame.display.set_caption('Iloncha :)')

self.surf = pygame.display.set_mode(self.SIZE)
self.clock = pygame.time.Clock()

self.snake = Snake(self.surf)

self.font = pygame.font.Font(None, 22)

self.play()

```

Agar kimdir avval OYD bilan ishlamagan bo'lsa, **self** - bu obyekt ichidagi har bir **metod** va **funksiya** qabul qiladigan majburiy **parametrdir**. U aynan shu obyektidagi **o'zgaruvchilar**, **konstantlar**, **kortejlar**, **lug'atlar** va **funksiyalar** bilan ishlash uchun foydalaniladi.

### **Endi 3-qadamni tushuntirib o'taman:**

- 1-qator: Pythondagi maxsus funksiyani yaratamiz
- 2-qator: tkinter va os modullaridan foydalanib o'yin oynasini ekran markazida joylashtiramiz
- 4-5-6-qator: PyGame modulida oynani yaratamiz
- 7-qator: Oyna sarlovhasini kiritamiz
- 8-qator: Oyna o'lchamini kiritamiz
- 9-qator: Kadrlar tezligi bilan ishlash uchun kerakli obyektни clock o'zgaruvchisiga joylashtiramiz
- 11-qator: Snake obyektі - bu yuqorida aytilgan obyekt bo'lib, uni boshqa faylda quyiroqda yaratamiz. Avvaldan unga surf o'zgaruvchisini parametr sifatida berib, qo'yamiz.
- 13-qator: Oynadagi matn o'lchamini belgilaymiz
- 15-qator: Quyida yaratadigan Ilon harakatini ta'minlovchi funksiya

### **4-qadam: Ilon harakatlanishini ta'minlovchi play funksiyasini yaratamiz**

```

def play(self):
    while True:
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                sys.exit()

        keys = pygame.key.get_pressed()
        if keys[K_LEFT] or keys[K_a]:
            if self.snake.direction != 'r':
                self.snake.direction = 'l'
        elif keys[K_RIGHT] or keys[K_d]:
            if self.snake.direction != 'l':
                self.snake.direction = 'r'
        elif keys[K_UP] or keys[K_w]:
            if self.snake.direction != 'd':
                self.snake.direction = 'u'
        elif keys[K_DOWN] or keys[K_s]:
            if self.snake.direction != 'u':
                self.snake.direction = 'd'

        if (self.snake.x < 0 or self.snake.y < 0 or self.snake.x +
self.snake.SIDE > self.WIDTH or self.snake.y + self.snake.SIDE > self.HEIGHT)
or len(self.snake.XY) != len(set(self.snake.XY)):

```

```
sys.exit()
```

```
self.draw()
```

### **Tushuntirishlar:**

- 1-qator: Tushunarli :)
- 2-qator: Harakat ma'lum takrorlanishdan so'ng to'xtab qolmasligi uchun barchasini cheksiz tsikl amalga oshiramiz
- 3-4-5-6-qator: Cheksiz rekursiya sodir bo'lmasligi uchun foydalanuvchi oynani yopmoqchi bo'lganida dasturni yakunlaymiz, ya'ni barcha eventlarni (hodisalarni) for tsiklidan o'tkazamiz va agar hodisa QUITga, ya'ni CHIQISHga teng bo'lganida dasturni yakunlaymiz
- 8-qator: Klavyaturadagi barcha bosilgan tugmalarni kuzatib, keys o'zgaruvchisiga joylashtiramiz
- 9-qatordan...20-qatorgacha: Qaysi tugma bosilganini tekshiramiz va kerakli tugmalar bosilganda quyiroqda yo'nalishini, yaratadigan Snake obyektimizga tegishli, Ilon harakatlanadigan yo'nalishni belgiluvchi direction o'zgaruvchiga joylashtiramiz
- 23-24-25-qator: Ilon oyna chegarasini kesib o'tganida dasturni tugatamiz
- 30-qator: Barcha elementlarni oynada joylashtiruvchi draw funksiyasini chaqiramiz. Uni esa quyiroqda yaratamiz.

### **5-qadam: Elementlarni oynada joylashtiruvchi draw funksiyasini yaratamiz**

Bu uchun

```
def draw(self):
    self.surf.fill(self.BACK)
    self.surf.blit(self.font.render('by Murodov Azizmurod', 1, (255, 165, 0)), (self.WIDTH - 180, self.HEIGHT - 25))

    self.snake.move_snake(self.snake.direction)

    pygame.display.update()
    self.clock.tick(self.fps)
```

kodini [main.py](#) fayliga kiritamiz. Bu satrlar:

- 1-qator: Funksiyani yaratamiz
- 2-qator: Oyna orqasiga [main.py](#) yuqorisida kiritgan BACK kortejida sqalanuvchi, RGB formatdagi rangni beramiz
- 3-qator: Keraksiz matnni oynada joylashtiramiz (bu qatorni o'chirib tashlasa ham bo'ladi)
- 5-qator: Snake obyektiga tegishli move\_snake funksiyasiga play funksiyasidan olgan Ilon yo'nalish joylashgan o'zgaruvchini parametr sifatida beramiz
- 7-qator: Barcha aks etishi uchun oynani yangilaymiz
- 9-qator: 1 soniyadagi kadlar sonini kiritamiz
- ni anglatadi.

Oxirida **Game obyektini cheksiz tsiklda** ishga tushiramiz:

```
while True:
    Game()
```

Shunday qilib o'yin oynasini ham yaratdik. Endi Snake obyektini yaratamiz:

## 2 - Bo'lim

### 1-qadam: Snake obyektini yaratamiz!

Avvaliga [snake.py](#) nomli faylni yaratamiz va birinchi navbatda kerakli modullarni ulaymiz:

```
import pygame
```

Bunda bizga faqatgina **PyGame** moduli kerak bo'ladi.

### 2-qadam: Endi aynan Snake obyektini yaratamiz:

```
class Snake:
    pass
```

Yuqoridagi **pass** - bu hech qanday vazifani bajarmaydi. U shunchaki **None** javobini qaytaradi.

### 3-qadam: So'ngra keyinroq ishlatadigan kartej va o'zgaruvchilarni yaratamiz:

```
COLOR = [255, 255, 255]
SIDE = 15
length = 2
beh = 10
```

**COLOR** - Ilon elementining rangini saqlovchi ro'yxat RGB formatda

**SIDE** - Ilon elementi bir-nechta kvadratlardan tashkil topgan bo'ladi. Bu o'zgaruvchida shu kvadratlarning tomoni uzunligini belgilaymiz

**length** - O'yin endi boshlangandagi uzunligi. Keyingi qismlarda ball olish imkoniyatini yaratamiz va ball ortgan sari bu o'zgaruvchidagi son ham ortib boradi.

**beh** - Kvadaratlar orasidagi masofa

### 4-qadam: Endi Game obyektini kabi Snakeni ham initsiyalizatsiya qilamiz (gavdalantiramiz). Obyekt qabul qiladigan yagona parametrlarni kerakli o'zgaruvchiga joylashtiramiz.

```
def __init__(self, surface):
    self.surface = surface
    self.x = surface.get_width() // 2 # Elementni oyna eni bo'yicha o'rtada
    joylashtiramiz
    self.y = surface.get_height() // 4 # Elementni oyna bo'yi bo'yicha
    o'rtada joylashtiramiz
    self.XY = [(self.x, self.y)] # Elementning boshlang'ich nuqtasi
    self.direction = 'd' # o'yin boshlangan element harakatlanadigan
    yo'nalish
```

### 5-qadam: So'ngra harakat yo'nalishiga qarab element joylashuvini o'zgartiruvchi get\_snake funksiyasini yozib olamiz:

```
def get_snake(self): # Funksiyani yaratamiz
    self.XY += [(self.x, self.y)] # elementni harakatini o'z uzunligiga
    teng masofaga uzoqlashish orqali amalga oshiramiz
    self.XY = self.XY[-self.length:] # elementni harakatini o'z uzunligiga
    teng masofaga uzoqlashish orqali amalga oshiramiz
    for kx, ky in self.XY: # har bir o'zgarish sodir bo'lganda elementning
    joylashuvini o'zgartirib boruvchi ysikl yaratamiz
        pygame.draw.rect(self.surface, self.COLOR, (kx, ky, self.SIDE,
        self.SIDE)) # Elementni oynada chizamiz
```

### 6-qadam: Hozir esa qaysi yo'nalish tugmasi bosilganini tekshirishlar orqali aniqlaymiz va unga qarab elementni o'sha tomonga harakatlantiramiz va bu uchun nomi ham vazifasini anglatuvchi

***move\_snake funksiyasini yaratamiz. U faqatgina bosilgan 1ta parametr sifatida tugma haqidagi ma'lumotlarni saqlovchi key o'zgaruvchisini beramiz:***

```
def move_snake(self, key):  
    if key == 'l':  
        self.x -= self.beh  
        self.get_snake()  
    if key == 'r':  
        self.x += self.beh  
        self.get_snake()  
    if key == 'u':  
        self.y -= self.beh  
        self.get_snake()  
    if key == 'd':  
        self.y += self.beh  
        self.get_snake()
```

**Snake** obyektini yozib bo'lganimizdan so'ng **Game** obyektida aytib o'tilgan qismlar tushunarli bo'ldi deb o'ylayman.

**7-qadam: Endi [main.py](#) faylining yuqorisiga, barcha modullarni ulovchi qismning pastiga quyidagi satrni kiriamiz:**

```
from snake import *
```

Bu bilan biz [snake.py](#) faylidagi Snake obyektiga tegishli barcha funksiya va o'zgaruvchilardan foydalana olamiz.

Tekshirish uchun [main.py](#) faylini ishga tushuramiz va natijaga ko'rishingiz mumkin.

## Xulosa

Xullas, shu bilan bugungi maqolamda sizga ko'rsatmoqchi va o'rgatmoqchi bo'lgan narsalarim shularda iborat edi.

Agar, sizga yoqqan bo'lsa va yangi bilimlarga ega bo'lsangiz - bundan faqatgina xursandman. Alohida o'rinda maqolamni oxirigacha o'qiganlarga rahmat aytmoqchiman va sizni tabriklayman. Aynan siz anchagina katta bo'lgan maqolani o'qishda o'zingizda kuch topa olgan ekansiz, demak, sizda aniq belgilangan maqsad bor va bu maqsad sayin harakatlanmoqdasiz va albatta, unga erishasiz. Men bunda sizga tilakdoshman.

Agar sizga maqolam maqul kelgan bo'lsa, albatta, bundan xursandman.

iXayr omon bo'ling!

## Pythonda "Iloncha" o'yinini yaratamiz (2-qism) – Praktika

**Hammaga salom!** Bugun biz klassik "Snake" o'yinini yaratishda davom etamiz. Agar maqolaning [1-qismi](#)ni o'qimagan bo'lsangiz, sizda tushunmovchiliklar bo'lishi mumkin. Lekin, o'qigan bo'lsangiz biz bosladik!



**Bugungi qismda biz quyidagi topshiriqlarni bajaramiz:**

1. "Food"larini (ozuqalarni) ekranda random joylashishini ta'minlaymiz.
2. Asosiy ilon elementi oyna chegarasiga yetganda "Game Over" oynasini yaratamiz.
3. Ilon elementi "food" elementi bilan to'qnashganda hisobni oshirib borish imkoniyatini yaratamiz.

**Boshladik:**

"Food" elementi uchun alohida Food obyektini yaratamiz va bu uchun [main.py](#) va [snake.py](#) fayllari joylashgan katalogda yangi [food.py](#) faylini yaratamiz.

Bu fayl birinchi bo'lib, kerakli modullarni ulaymiz:

```
import pygame
from random import randint
```

**PyGame** - avvalgi maqolada aytilganidek, o'yin oynasi bilan ishlash uchun foydalanadigan modulimiz.

**Random** - yuqorida aytilgan random, ya'ni to'g'ri kelgan sonni olishda kerak bo'luvchi modul.

### 1 - Bo'lim

**1-qadam: Obyektini yaratamiz:**

```
class Food:
```

**2-qadam: Kerakli kortejni va o'zgaruvchini yozamiz.**

```
COLOR = (255, 0, 0) # food elementining rangi
SIDE = 20 # kvadrat shaklidagi elementning tomoni uzunligi
```

**3-qadam: Game obyektini initsializatsiya qilamiz. Ya'ni obyektga beriladigan parametrlarni kerakli o'zgaruvchiga joylashtiramiz va kodda ishlatadigan o'zgaruvchi va boshqa kerakli parametrlarni shu qismda yaratib olamiz. Bu yerda obyektimiz 1 ta parametr qabul qiladi.**

```
def __init__(self, surface):
    self.surface = surface # o'yin oynasi bilan aloqalashishi uchun ulaymiz
    self.x = self.SIDE # elementning x o'qi bo'yicha o'lchamini kiritamiz
    self.y = self.SIDE # elementning y o'qi bo'yicha o'lchamini kiritamiz
    self.set_x = set() # set() funksiyasi orqali x o'qida takrorlanuvchi
koordinatalarni o'chiramiz
    self.set_y = set() # set() funksiyasi orqali y o'qida takrorlanuvchi
koordinatalarni o'chiramiz
```

**4-qadam: Food elementini oynada chizuvchi `add_food` funksiyasini yaratib olamiz:**

```
def add_food(self):
    pygame.draw.rect(self.surface, self.COLOR, (self.x, self.y, self.SIDE,
self.SIDE))
    self.set_x = set(range(self.x, self.x + self.SIDE + 1))
    self.set_y = set(range(self.y, self.y + self.SIDE))
```

**5-qadam: Yangi elementni to'g'ri kelgan joyda joylashtiruvchi `new_food()` funksiyasini yaratamiz va bunda random modulining `randint` funksiyasidan foydalanamiz:**

```
def new_food(self, snake_side):
    self.x = randint(snake_side, self.surface.get_width() - snake_side)
    self.y = randint(snake_side, self.surface.get_height() - snake_side)
```

**6-qadam: Ilon elementi food elementi bilan to'qnashgan yoki to'qnashmaganini tekshiruvchi `is_eaten()` funksiyasini yaratamiz:**

```
def is_eaten(self, snake_x, snake_y, snake_side):
    if set(range(snake_x, snake_x + snake_side + 1)) && self.set_x
and set(range(snake_y, snake_y + snake_side + 1)) && self.set_y:
        return True
    return False
```

Ushbu funksiya **True** yoki **False** javobini qaytaradi:

- To'qnashdi - **True**
- To'qnashmadi - **False**

Shu bilan Food obyektining kodini yozib bo'ldik. Endi faqagina [main.py](#) fayliga [food.py](#) faylini ulaymiz va yuqoridagi funksiyalarni `Game` obyektida qo'llaymiz.

## 2 - Bo'lim: "Game Over" xabarini chiqaramiz.

[food.py](#) faylidagi barcha funksiyalardan foydalanish uchun [main.py](#) fayli yuqorisiga usgbu faylni ulaymiz:

```
from food import *
```

Bu uchun [main.py](#) faylining pastgi qismida `game_over()` funksiyasini yaratamiz:

```
def game_over(self):
    while True: # cheksiz tsikl yaratamiz
        self.surf.blit(self.font_end.render(f'Jami hisobingiz:
{self.score}', 1, (255, 165, 0)),
                        (self.WIDTH//2 - 130, self.HEIGHT//3)) # Hisob
to'g'risidagi yozuvni oyna markazida aks ettiramiz
```



```

        mouse_coord = pygame.mouse.get_pos() # sichqonchaning oyna ustidagi
holatini (koordinatlarini) olamiz
        mouse_events = pygame.mouse.get_pressed() # sichqoncha bosilganda
ko'rsatgichning holatini olamiz
        if self.WIDTH//2 - 50 < mouse_coord[0] < self.WIDTH//2 + 50 and
self.HEIGHT//2 - 50 < mouse_coord[1] < self.HEIGHT//2 + 10:
            pygame.draw.rect(self.surf, self.GREEN, (self.WIDTH//2 - 50,
self.HEIGHT//2-50, 100, 40)) # sichqoncha RETRY tugmasi ustiga kelganda hover
effektini qilamiz ya'ni tugmaning orqa foni rangini o'zgartiramiz
            if mouse_events[0]:
                break
        else:
            pygame.draw.rect(self.surf, self.RED, (self.WIDTH//2 - 50,
self.HEIGHT//2 + 50, 100, 40))
            self.surf.blit(self.font_button.render('RETRY', 1, self.BACK),
                           (self.WIDTH // 2 - 33, self.HEIGHT // 2 + 40))
            self.clock.tick(self.fps)
            pygame.display.update() # oynani yangilab boramiz
            for event in pygame.event.get(): # har bir oynada sodir bo'ladigan
event - hodisalarni tekshiri boramiz va u CHIQQISH buyrug'i bo'lganda oynani
yopamiz
                if event.type == QUIT:
                    pygame.quit()
                    sys.exit()

    Game()

```

So'ngra `play()` funksiyasiga o'zgartirishlar kiritamiz:

Birinchidan **food** elementiga to'qnashganda:

- Ilon elementinig uzunligini oshiramiz,
- Yangi food yaratamiz,
- Hisobni 1ga oshiramiz,
- O'yin tezligini, ya'ni soniyasiga freymrlar sonini oshirib boramiz

```

if self.food.is_eaten(self.snake.x, self.snake.y, self.snake.SIDE):
    self.snake.add_lenght()
    self.food.new_food(self.snake.SIDE)
    self.score += 1
    self.fps += 1

```

Ikkinchidan avvalgi maqolada qilingandek Ilon elementi oyna chegarasiga yetganda oynani yopmasdan **Food** obyektidagi `game_over()` funksiyasini chaqiramiz:

```

if (self.snake.x < 0 or self.snake.y < 0 or self.snake.x + self.snake.SIDE
> self.WIDTH
    or self.snake.y + self.snake.SIDE > self.HEIGHT) or
len(self.snake.XY) != len(set(self.snake.XY)):
    self.game_over()

```

Endi **"food"** ham oynada boshqa elementlar kabi "chizilishi" uchun uni `draw()` fubksiyasida chaqiramiz.

```

def draw(self):
    self.surf.fill(self.BACK)
    self.surf.blit(self.font_score.render(f'Hisobingiz: {self.score}', 1,
(255, 165, 0)), (self.WIDTH - 105, 5))
    self.surf.blit(self.font_score.render('by Murodov Azizmurod', 1, (255,
165, 0)), (self.WIDTH - 180, self.HEIGHT - 25))
    self.food.add_food()

    self.snake.move_snake(self.snake.direction)

```

```
pygame.display.update()  
self.clock.tick(self.fps)
```

## **Xulosa**

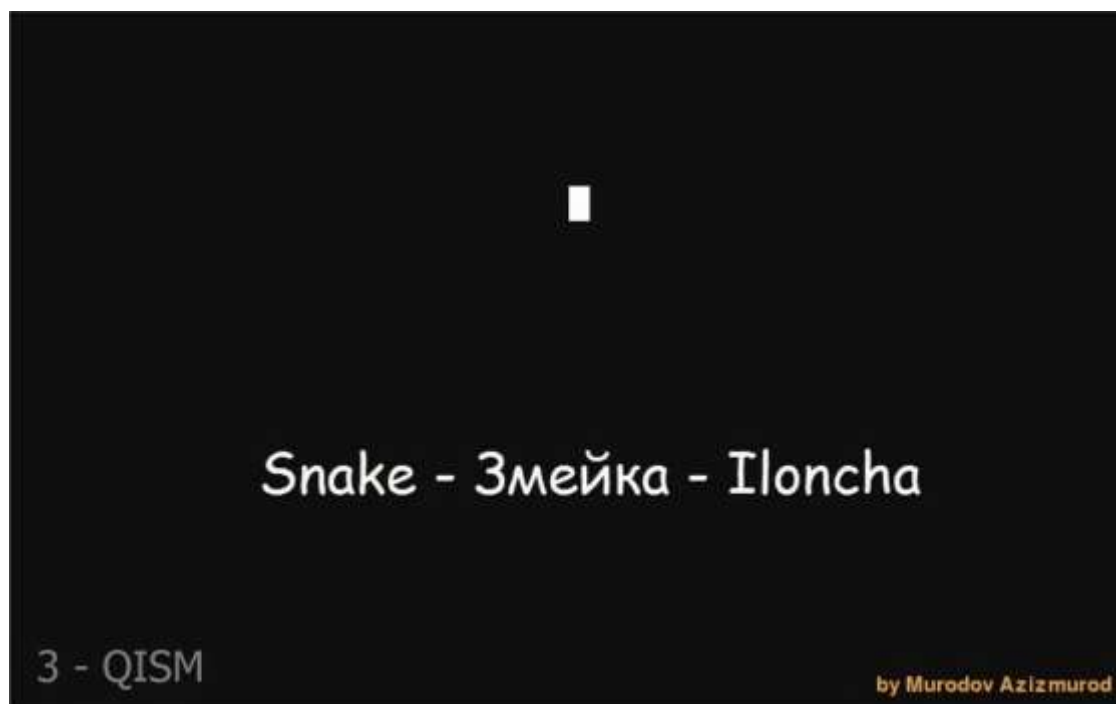
**Xullas, shu bilan bugungi maqolamda sizga ko'rsatmoqchi va o'rgatmoqchi bo'lgan narsalarim shularda iborat edi.**

**Agar, sizga yoqqan bo'lsa va yangi bilimlarga ega bo'lsangiz - bundan xursandman.**

**Agar sizga maqolam maqul kelgan bo'lsa, albatta, xursandman. Keyingi maqolada o'yin oynasining orqasiga harakatlanuvchi fon va tovushli effektlar beramiz.**

**Xayr, omon bo'ling!**

## Pythonda "Iloncha" o'yinini yaratamiz (3-qism) – Praktika



Hammaga salom! Bugun biz klassik **"Snake"** o'yinini yaratishda davom etamiz. Agar maqolaning [birinchi](#) va [ikkinchi](#) qismlarini o'qimagan bo'lsangiz, sizda tushunmovchiliklar bo'lishi mumkin. Lekin, o'qigan bo'lsangiz biz bosladik!

Bugungi qismda biz quyidagi topshiriqlarni bajaramiz:

1. O'yin oynasining orqasiga **"Ilon"** elementining yo'nalishiga qarab harakatlanuvchi fon yaratamiz.
2. Turli musiqali effektlar beramiz.

### Boshladik:

Birinchi bo'lib qolgan fayllar joylashgan kataloga yangi [background.py](#) faylini yaratamiz. Ushbu faylda esa Background obyektini joylashtiramiz.

[background.py](#) fayliga kerakli modullarni ulaymiz.

```
import pygame
from random import randint, randrange
```

**PyGame**- o'yin oynasi bilan ishlash uchun foydalanadigan modul.

**Random**- To'g'ri kelgan sonni olish uchun foydalanamiz.

### 1 - Bo'lim

#### 1-qadam: Obyektni yaratamiz:

```
class Background:
```

**2-qadam: Background obyektini initsiyalizatsiya qilamiz. Ya'ni obyektga beriladigan parametrni kerakli o'zgaruvchiga joylashtiramiz va kodda ishlatadigan o'zgaruvchi va boshqa kerakli parametrlarni shu qismda yaratib olamiz. Bu yerda obyektimiz 1 ta parametr qabul qiladi.**

```
def __init__(self, surface):
    self.surf = surface # fonni joylashtirish uchun o'yin oynasini obyektga
    ulaymiz
    self.max_obj = 500 # Fondagi harakatlanuvchi kvadratning soni
    self.width = surface.get_width() # oynaning eni o'lchamini olamiz
    self.height = surface.get_height() # oynaning bo'yi o'lchamini olamiz
    self.x = [randrange(-self.width, 2 * self.width) for i in
range(self.max_obj)] # kvadratni for loop tsikli yordamida ko'paytirib, x o'qi
bo'yicha random joyda joylashtiramiz
    self.y = [randrange(-self.height, 2 * self.height) for i in
range(self.max_obj)] # kvadratni for loop tsikli yordamida ko'paytirib, y o'qi
bo'yicha random joyda joylashtiramiz
    self.side = [randint(2, 5) for i in range(self.max_obj)] #
kvadratlarning o'lchamini kiritamiz
    self.speed = [randint(1, 4) for i in range(self.max_obj)] #
kvadratlarning tezligini belgilaymiz
    self.color = [(randrange(255), randrange(255), randrange(255)) for i in
range(self.max_obj)] # kvadratlarning rangini kiritamiz
```

**3-qadam: Background obyektida fonni oynada joylashtiruvchi draw funksiyasini yaratamiz:**

```
def draw(self, directon):
    if directon == 'l': # agar yo'nalish chap tomon bo'lsa
        self.x = [x + r for x, r in zip(self.x, self.speed)] # x o'qi
bo'yicha harakatlantiramiz
    elif directon == 'r': # agar yo'nalish o'ng tomon bo'lsa
        self.x = [x - r for x, r in zip(self.x, self.speed)] # x o'qi
bo'yicha harakatlantiramiz
    elif directon == 'u': # agar yo'nalish yuqori tomon bo'lsa
        self.y = [y + r for y, r in zip(self.y, self.speed)] # y o'qi
bo'yicha harakatlantiramiz
    elif directon == 'd': # agar yo'nalish past tomon bo'lsa
        self.y = [y - r for y, r in zip(self.y, self.speed)] # y o'qi
bo'yicha harakatlantiramiz
    [pygame.draw.rect(self.surf, self.color[i], (self.x[i], self.y[i],
self.side[i], self.side[i])) if i % 5
else pygame.draw.rect(self.surf, (randrange(255),
randrange(255), randrange(255)), (
self.x[i], self.y[i], self.side[i], self.side[i])) for i in
range(self.max_obj)]
```

**4-qadam: [main.py](#) fayliga o'zgartirishla kiritamiz:**

**a)** [background.py](#) faylini asosiy faylga ulaymiz:

```
from background import *
```

**b)** `__init__` funksiyasida Background **obyektini** o'zgaruvchiga joylashtiramiz:

```
self.bg = BG(self.surf)
```

**c)** fon ham boshqa elementlar kabi oynada aks etishi

uchun Background **obyektining** draw funksiyasini Game **obyektining** draw funksiyasida ishga tushiramiz

```
self.bg.draw(self.snake.direction)
```

`main.py` faylini ishga tushiramiz va natijani ko'rishimiz mumkin.

## 2 - Bo'lim

**1-qadam: Kerakli tovushlardan foydalanishni osonlashtirish uchun ularni kerakli o'zgaruvchilarga ulaymiz. Buni ham yuqoridagidek Game obyektining `__init__` funksiyasida amalga oshiramiz**

**Aytgacha barcha tovush va fayllarni quyiroqdagi link orqali yuklab olishigiz mumkin.**

O'zgaruvchilarni oynani yaratganimizdan so'ng yaratib olamiz:

```
pygame.mixer.music.load('sounds/music.mid') # O'yin davomidagi asosiy musiqa
pygame.mixer.music.play()
self.sound_game_over = pygame.mixer.Sound('sounds/game_over.wav') # o'yin tugaganida tovush
self.sound_eat_food = pygame.mixer.Sound('sounds/eat_food.wav') # Ilon elementi "food" bilan to'qnashgani ishlatiladigan tovush
```

Endi esa o'yin tugaganida asosiy musiqani to'xtatamiz va `sound_game_over` o'zgaruvchisidagi tovushni eshittiramiz. Bu uchun barcha ishni `game_over` funksiyasida amalga oshiramiz ya'ni barchasi oson:

```
pygame.mixer.music.stop()
self.sound_game_over.play()
```

"Food"ga urilgan yoki urilmaganini

tekshiruvchi `play` funksiyasidagi `if` tekshiruvida `sound_eat_food` o'zgaruvchisidagi tovushni ishga tushiramiz:

```
self.sound_eat_food.play()
```

Shu bilan **3ta** maqoladan iborat bo'lgan **Python dasturlash tilida 2D** o'yin yaratish haqidagi ketma-ket darslar tugadi. **Yana qo'shimcha qo'shish to'g'risidagi fikrlar bo'lsa, izohlarga yozing. Albatta, 4-qismni ham tayyorlayman :)**

**Xullas, shu bilan bugungi maqolamda sizga ko'rsatmoqchi va o'rgatmoqchi bo'lgan narsalarim shularda iborat edi.**

**Agar, sizga yoqqan bo'lsa va yangi bilimlarga ega bo'lsangiz - bundan xursandman.**

**Agar sizga maqolam maqul kelgan bo'lsa, xursandman.**

**Xayr, omon bo'ling!**