**PYTHON ТІЛІНДЕ ОЙЫН ЖАСАУ ҚЫЗЫҚТЫ ЖӘНЕ ОҢАЙ!**

***Бекбулатова Әсел Бекбулатқызы***

***“Т.Айбергенұлы атындағы №19 мектеп гимназиясы”КММ***

***Информатика пәні мұғалімі***

***Педагог-модератор***

Python — бағдарламалау тілі, ол өз қарапайымдылығымен және қуатты кітапханаларымен танымал. Python тілінде ойын жасау — жаңа бастағандар үшін өте ыңғайлы және қызықты тапсырма. Бұл мақалада біз Python тілінде ойын жасау үшін не керек екенін, қандай құралдар мен кітапханаларды қолдануға болатынын және алғашқы ойын жасаудағы қадамдарды қарастырамыз.

1. Ойын жасауға қажетті құралдар мен кітапханалар

Python тілінде ойын жасауда кеңінен қолданылатын бірнеше кітапханалар бар. Олардың ішіндегі ең танымалдары:

1. Pygame:

- Pygame — Python тілінде 2D ойындарын жасауға арналған ең кеңінен қолданылатын кітапхана. Бұл кітапхана графикамен, дыбыспен жұмыс істеуге, пайдаланушының кірісін өңдеуге және ойын логикасын жүзеге асыруға мүмкіндік береді.

2. Arcade:

- Arcade — Pygame сияқты ойын жасау үшін қолданылатын басқа бір құрал, бірақ ол жаңа пайдаланушыларға көбірек бағытталған және өте оңай пайдаланылады.

3. Pyglet:

- Pyglet — мультимедиалық қосымшалар, оның ішінде ойындар жасау үшін қолданылатын Python кітапханасы. Ол 2D және 3D графика, дыбыстық басқару және мультимедиялық контент үшін қолайлы.

2. Python-да ойын жасауға арналған алғашқы қадамдар

Енді Python тілінде ойын жасау үшін ең қарапайым қадамдарды қарастырайық. Біз осы жерде Pygame кітапханасын пайдаланып, шағын 2D ойын жасауды үйренеміз.

Қадам 1: Pygame кітапханасын орнату

Pygame кітапханасын орнату үшін төмендегі команданы Python командалық жолында орындау қажет:

bash

pip install pygame

Қадам 2: Ойын терезесін жасау

Алғашқы қадам — ойын терезесін құру. Бұл үшін Pygame кітапханасының бірнеше негізгі функциясын пайдалану керек.

python

import pygame

Pygame кітапханасын бастапқы күйге келтіру

pygame.init()

Ойын терезесінің өлшемдері

screen\_width = 800

screen\_height = 600

Ойын терезесін жасау

screen = pygame.display.set\_mode((screen\_width, screen\_height))

pygame.display.set\_caption("Алғашқы Python ойыным")

Ойынның басты циклін жасау

running = True

while running:

# Оқиғаларды өңдеу (жабу түймесі)

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# Экранды түсіру (қайтадан бояу)

screen.fill((0, 0, 0)) # Экранды қара түспен бояу

# Экранды жаңарту

pygame.display.update()

Pygame кітапханасын жабу

pygame.quit()

[16:04, 12.03.2025] ChatGPT: Бұл код арқылы біз қарапайым қара түсті экран көрсететін терезе жасаймыз. Ойынды аяқтау үшін терезе жапқышын басқан кезде pygame.quit() командасы арқылы бағдарламадан шығады.

Қадам 3: Ойынның кейіпкерін жасау

Келесі қадам — экранда кейіпкер (мысалы, шар немесе квадрат) көрсететін код жазу.

```python

import pygame

Pygame кітапханасын бастапқы күйге келтіру

pygame.init()

Ойын терезесінің өлшемдері

screen\_width = 800

screen\_height = 600

Ойын терезесін жасау

screen = pygame.display.set\_mode((screen\_width, screen\_height))

pygame.display.set\_caption("Шарды жылжыту ойыны")

Кейіпкердің бастапқы координаттары

x, y = screen\_width // 2, screen\_height // 2

speed = 5

Ойынның басты циклін жасау

running = True

while running:

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# Пайдаланушының пернелерді басуы арқылы қозғалыс

keys = pygame.key.get\_pressed()

if keys[pygame.K\_LEFT]:

x -= speed

if keys[pygame.K\_RIGHT]:

x += speed

if keys[pygame.K\_UP]:

y -= speed

if keys[pygame.K\_DOWN]:

y += speed

# Экранды бояу

screen.fill((0, 0, 0)) # Қара фон

# Кейіпкерді (шарды) салу

pygame.draw.circle(screen, (255, 0, 0), (x, y), 20) # Қызыл шар

[16:04, 12.03.2025] ChatGPT: # Экранды жаңарту

pygame.display.update()

Pygame кітапханасын жабу

pygame.quit()

Бұл кодты орындаған кезде экранда қызыл шар пайда болып, оны пернелер арқылы жылжытуға болады. Бұл ойынның негізгі қозғалыс механикасын жасайды.

\*Қадам 4: Ойынның қосымша элементтері\*

Енді ойынды қызықты ету үшін кейбір қосымша мүмкіндіктерді қосуға болады:

- \*Дыбыс эффектілерін\* қосу (мысалы, шар қозғалғанда дыбыс шығару).

- \*Уақыт немесе ұпай жүйесі\* жасау.

- \*Кедергілерді қосу\* (мысалы, экранда кездейсоқ орналасатын бөгеттер).

python

Дыбыстық файлды қосу

bounce\_sound = pygame.mixer.Sound("bounce.wav")

Пернелерді басқанда дыбыс шығару

if keys[pygame.K\_LEFT] or keys[pygame.K\_RIGHT] or keys[pygame.K\_UP] or keys[pygame.K\_DOWN]:

bounce\_sound.play()

```

3. Ойын жасаудың негіздері мен кеңестер

1. Ойын логикасын жасау: Ойынды жасаудағы маңызды аспектілердің бірі — ойынды басқару логикасын дұрыс құру. Ойынның ережелерін, мақсаттарын және қиындық деңгейлерін анықтау өте маңызды.

2. Графика мен дыбысты қолдану: Ойындарда графика мен дыбыстың рөлі үлкен. Pygame кітапханасы графикамен жұмыс істеуге арналған қарапайым әдістерді ұсынады.

3. Тест пен оңтайландыру: Ойын аяқталған соң, оны бірнеше рет тестілеу керек. Қателер мен ақауларды түзету, тиімділікті арттыру үшін оңтайландыру жұмыстары қажет.

Қорытынды

Python тілінде ойын жасау — қызықты әрі тиімді процесс. Pygame сияқты кітапханалар көмегімен ойындарды бағдарламалау өте оңай. Қарапайым ойын жасаудан бастау алған соң, күрделі ойындарға өту үшін, алгоритмдер мен дизайн мәселелерін тереңірек түсіну қажет болады. Бұл процесс қызығушылық пен шығармашылықты талап етеді, әрі Python тілі ойын жасаушылар үшін өте ыңғайлы тіл болып табылады.