Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

Шарттарды программалау және PyGame кітапханасын қосу

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_27\_\_\_» \_\_\_қаңтар \_\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-23

Сабақ түрі: Жаңа білімді меңгеру

**2. Мақсаты, міндеттері:**

#### **Оқу:** Білімгерлерге шарттар мен логикалық оператормен жұмыс істеуді үйрету. PyGame кітапханасы негізінде қарапайым ойындар жасау арқылы бағдарламалау дағдыларын дамыту.

#### **Дамыту:** Ойындар жасау барысында алгоритмдер мен деректер құрылымдарын пайдалану дағдыларын дамыту. Қарапайым анимациялар мен графикалық интерфейстер жасау арқылы пайдаланушының визуалды ойлау дағдыларын жетілдіру.

#### **Тәрбиелік:** Жоба барысында жүйелі және шығармашылық түрде жұмыс жасау дағдыларын қалыптастыру. Командалық жұмысқа бейімделу, қажет болған жағдайда басқа адамдармен бірге код жазу дағдыларын дамыту.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Студенттер шарттарды тиімді жазуды үйреніп, бағдарламада түрлі жағдайларға сәйкес әрекеттерді таңдай алады. PyGame кітапханасын орнату және оның негізгі мүмкіндіктерін пайдалану жолдарын меңгереді. Экранда графикалық объектілерді шығару, қозғалыс және анимация жасау дағдыларын алады.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Менен сұрақ - Сізден жылдам жауап” әдісі*** *(15 минут)*

**Шарттарды программалау және PyGame кітапханасын қосу**

**1. Шарттарды программалаудың негіздері**

Шарттарды программалау – бұл бағдарламаның белгілі бір жағдайларға немесе шарттарға сәйкес әрекет етуін анықтайтын процесті білдіреді. Python тілінде бұл үшін негізгі конструкциялар:

* **if** – Егер шарт орындалса, бір әрекет орындалады.
* **else** – Егер шарт орындалмаса, басқа әрекет орындалады.
* **elif** – Қосымша шарттарды тексеруге мүмкіндік береді.

**Мысал:**

python

x = 10

if x > 5:

print("x 5-тен үлкен")

elif x == 5:

print("x тең 5-ке")

else:

print("x 5-тен кіші")

Бұл кодта, x мәні 10 болғандықтан, "x 5-тен үлкен" деген хабарлама шығады.

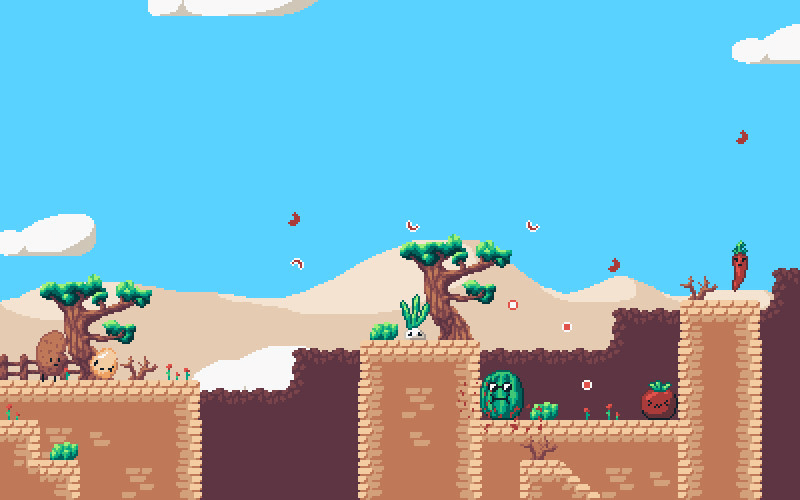
**Pygame** - компьютерлік ойындар мен мультимедиялық қосымшаларды жазуға арналған Python бағдарламалау тілінің модульдерінің (кітапханаларының) жиынтығы.

Pygame бастапқыда Пит Шиннерс (Pete Shinners) жазған. 2004-2005 жылдардан бастап еркін бағдарламалық жасақтама қауымдастығы қолдайды және дамытады. Pygame қарапайым DirectMedia Layer (SDL) кітапханасын С бағдарламалау тілі мен оның туындыларының төмен деңгейлі механикасы жоқ нақты уақыттағы компьютерлік ойындардың дамуын қамтамасыз ету мақсатында пайдаланады. Бұл ойын ішіндегі ең қымбат мүмкіндіктерді ойын логикасынан абстракциялауға болады деген болжамға негізделген , бұл ойынды құрылымдау үшін Python сияқты жоғары деңгейлі бағдарламалау тілін пайдалануға мүмкіндік береді.

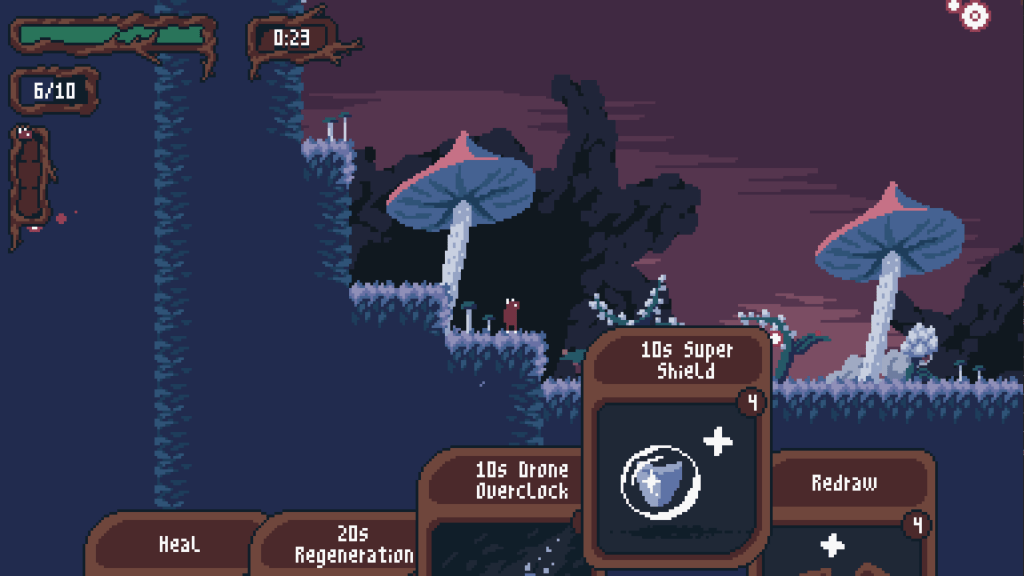
**Pygame көмегімен қандай ойындар жасалады?**

Бірқатар басқа инди қозғалтқыштарынан айырмашылығы, Pygame қатты және танымал ойындардың мысалдарымен мақтана алмайды. Бірақ онда бірнеше жобалар жазылған. Мұнда бірнеше мысалдар келтірілген.

**Super Potato Bruh** - Латук ханшайымын Черпак мырзадан құтқару үшін кедергілерді жеңіп, жауларын жеңетін картоп адамы туралы платформер. Көңілді сипаттамаға қарамастан, ойын өте күрделі, ерекше физика және bullet hell-бұл ойын барысында кейіпкерге жаудың әр түрлі снарядтары үнемі ұшатын кіші жанр.



**Drawn down Abyss** - тағы бір платформер, бұл жолы карта. Басты кейіпкер біртіндеп тұңғиыққа түсіп, оның түбіне жетуге тырысады және жолда әртүрлі жауларға тап болады. Батырдың қабілеттері Қорғаныс немесе шабуыл жасау үшін ойнауға болатын карталар ретінде ұсынылған. Бұл ойынды Super Potato Bruh сияқты әзірлеуші жасаған: оның лақап аты — DaFluffyPotato. Ол сонымен қатар Pygame-де тағы бірнеше әдемі пиксель ойындарын, негізінен платформерлерді жасады.



**Pygame-де ойын құру ерекшеліктері**

Бейне ойын-оқиғаға бағытталған бағдарлама. Бұл бағдарлама оқиғаларға жауап береді дегенді білдіреді: түймені басу, тінтуірдің қозғалысы, біраз уақыттан кейін. Оқиға болған кезде белгілі бір код іске қосылады: функция немесе қарапайым әрекет. Осылайша ойындар интерактивті болады. Түймелерге немесе тінтуірді басуға жауап беретін кез-келген мультимедиялық қосымшалар оқиғаға бағытталған болып саналады. Бірақ ойындар жағдайында мүмкін болатын оқиғалар көп-мысалы, екі объектінің соқтығысуы. Сондықтан ойынның жұмысын шартты түрде үш бөлікке бөлуге болады: оқиғаларды бақылау; оқиғаларға сәйкес ойындағы жағдайды өзгерту; көрсету, яғни жаңа параметрді ескере отырып, экранды қайта құру. Pygame-де барлық үш бөлікті жүзеге асыруға арналған құралдар мен мүмкіндіктер жиынтығы бар. Мысалы, онда нөлден жазуды қажет етпейтін көптеген кіріктірілген оқиғалар бар. Ол сондай — ақ оқиғаларды сақтайды, тіпті егер олар кадрды қайта құру кезінде болса да-олар арнайы тізімде қалады және әзірлеуші оларға қандай да бір реакция жаза алады.

Осының бәрімен Pygame өте төмен деңгейде. Ойын дамыту контекстінде бұл көптеген мүмкіндіктерді білдіреді әзірлеуші қолмен жүзеге асыруы керек. Мысалы, ойын физикасы үшін дайын функциялар жоқ-іске асыруды өзіңіз жазып алуыңыз керек, яғни формула кодын оқулықтан көшіріңіз. Бұл ыңғайлы болатын мысал ретінде кейіпкерді жеделдету, тақырыпты күрделі траектория бойынша құлату, баллистика және т.б. Ойын циклі және оны Pygame-де жүзеге асыру Ойын шексіз цикл ретінде жүзеге асырылады. Бұл циклде бағдарлама оқиғаларды тексереді, оларға жауап береді және экранды қайта сызады — жоғарыда сипатталған үш компонент. Мұның бәрі пайдаланушы "Шығу"түймесін басқанша жалғасады. Бұл циклды Pygame-де қалай жүзеге асыруға болады. Үлгі. Алдымен әзірлеуші ойын үлгісін жасайды: терезенің ұзындығы мен Шири сипаттайды, қажетті FPS — секундына кадрлар санын көрсетеді. FPS неғұрлым жоғары болса, геймплей соғұрлым жылдам болады. Ойынның "сүйегінде" Сіз Pygame-ді инициализациялауыңыз керек, яғни оны қосуға мәжбүр етіңіз. Мұнда сіз бастау экранын жасай аласыз, оған әріптерді шығара аласыз, дыбыстарды қосу және таймерді қосу мүмкіндігін қоса аласыз. Таймер-уақытты санауға мүмкіндік беретін тағы бір Pygame мүмкіндігі. Осы бөлімде циклдің өзі жасалады.

FPS. FPS бақылау үшін бөлек команда бар. Ең басында, ойынның негізін жасау кезінде сіз таймерді тағайындай аласыз-ол кадр өзгергенге дейін өтуі керек уақытқа жауап береді. Қажетті FPS жеке сандық айнымалыда жақсы көрсетілген:

timer = pygame.time.Clock() # создаем таймер

FPS = 30 # указываем нужный FPS, тут он равен 30 кадрам в секунду

Таймерді шексіз циклден тыс жасау керек, бірақ уақытты санау — оның ішінде. Мұны істеу үшін циклдің басында пәрменді тағайындау керек:

timer.tick(FPS) # объясняем Pygame, как часто обновлять экран

Pygame барлық оқиғаларды арнайы тізімге түсіреді - оған кіру үшін pygame пәрмені бар.event.get(). Оқиғаны өңдеу үшін сіз осы тізімді сұрыптап, қажет нәрсені тауып, оған реакция жазуыңыз керек. Мысалы, бірінші қадам-pygame оқиғасына реакцияны жүзеге асыру.QUIT, яғни "Шығу"түймесін басыңыз. Сіз тізім бойынша өтуіңіз керек және егер қандай да бір оқиғаның түрі pygame-ге тең болса.QUIT, шексіз циклден шығу пәрменін беріңіз. Онсыз ойынды жабу мүмкін болмайды.

Суретпен жұмыс. Pygame-де экранды түспен толтыратын, оған кейіпкерлер салатын және басқа нәрселер жасайтын командалар бар. Бірақ сурет салу-бұл баяу процесс: әр пиксельге арналған команда графикалық картаға жетуі керек. Сондықтан Қос Буферлеу бар: экранда бір кадр болғанша," сорғыштың астында " болған барлық әрекеттердің екіншісі қалыптасады. Кадрды өзгерту үшін қажетті уақыт өткенде, ол өзгереді және болған барлық өзгерістерді көрсетеді. Яғни, экран әр әрекеттен кейін қайта жасалмайды: бір кадрда бірнеше өзгерістер болуы мүмкін. Pygame - де қос Буферлеу Автоматты, ал кадрды өзгерту үшін арнайы pygame командасы бар.flip(). Оны шексіз циклдің соңына қою керек: одан кейін болатын барлық нәрсе көрінбейді.

**Pygame-де тағы қандай құралдар бар?**

Ойындар мен мультимедиялық қосымшаларды жасау кезінде қолдануға болатын Pygame мүмкіндіктерінің бірнеше мысалдары келтірілген.

* Pygame модулі.draw ойын терезесінде геометриялық фигураларды салуға мүмкіндік береді. Онда әртүрлі пішіндерге арналған функциялар жиынтығы бар: сызықтар, үшбұрыштар, тіктөртбұрыштар және т.б.
* Pygame модулі.image кейіпкерлердің спрайттарын жүктеу сияқты суреттермен жұмыс істеуге жауап береді. Pygame модулі.трансформация суреттерді әр түрлі жолмен өзгерте алады, ал pygame.қаріп қаріптермен жұмыс істеуге мүмкіндік береді.
* Pygame модульдері.mixer және pygame.mixer.музыка дыбыстарға жауап береді: біріншісі дыбыстық эффектілермен жұмыс істеу үшін қажет, екіншісі фондық музыкамен жұмыс істейді.
* Pygame сыныбы.Surface графикалық редактордағыдай сурет салу үшін беттерді — "қабаттарды" жасауға мүмкіндік береді. Мысалы, фонның" негізгі "бетінде кейіпкерлердің фигуралары бар" қабаттар " құрып, оларды фонның үстіне салуға болады.

**2. PyGame кітапханасын қосу**

PyGame – Python тілінде ойындарды және графикалық интерфейстерді жасау үшін арнайы әзірленген кітапхана. PyGame арқылы экранға суреттерді шығару, пернетақта мен тышқан оқиғаларын өңдеу және анимация жасауға болады.

**PyGame орнату:**

PyGame кітапханасын орнату үшін терминалда келесі пәрменді орындаңыз:

bash

КопироватьРедактировать

pip install pygame

**PyGame кітапханасын қосу:**

PyGame бағдарламасында қолданылатын негізгі импорттар:

python

import pygame

from pygame.locals import \*

**3. PyGame бағдарламасының құрылымы**

PyGame бағдарламасының негізгі құрылымы келесі қадамдардан тұрады:

1. **PyGame инициализациясы:** Барлық модульдерді іске қосу.
2. **Ойын терезесін орнату:** Терезенің өлшемін және атауын анықтау.
3. **Ойынның негізгі циклі:** Оқиғаларды өңдеу, графиканы жаңарту және экранды көрсету.

**Қарапайым бағдарлама мысалы:**

python

import pygame

# PyGame инициализациясы

pygame.init()

# Терезенің өлшемін орнату

screen = pygame.display.set\_mode((800, 600))

pygame.display.set\_caption("PyGame шарттар бағдарламасы")

# Түстер анықтамасы

WHITE = (255, 255, 255)

BLUE = (0, 0, 255)

# Негізгі цикл

running = True

while running:

# Оқиғаларды тексеру

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# Экранды толтыру

screen.fill(WHITE)

# Жағдайларды тексеру және графиканы жаңарту

if pygame.mouse.get\_pressed()[0]: # Егер тышқанның сол батырмасы басылса

pygame.draw.circle(screen, BLUE, pygame.mouse.get\_pos(), 50)

# Экранды жаңарту

pygame.display.flip()

# PyGame ресурстарын босату

pygame.quit()

**4. Шарттарды PyGame-де қолдану**

PyGame-де шарттарды программалау оқиғаларды өңдеу (event handling), графикалық элементтерді басқару және пайдаланушы әрекетіне жауап беру үшін пайдаланылады.

**Оқиғалар мен шарттарды басқару:**

1. **Тышқан:** Тышқанның батырмалары басылғанда немесе қозғалғанда әрекет орындау.
2. **Пернетақта:** Пернетақтадан түймелерді басуды өңдеу.

**Мысал: Төртбұрышты жылжыту**

python

import pygame

# PyGame инициализациясы

pygame.init()

# Экран параметрлері

screen = pygame.display.set\_mode((800, 600))

pygame.display.set\_caption("Төртбұрышты жылжыту")

# Түстер

BLACK = (0, 0, 0)

GREEN = (0, 255, 0)

# Төртбұрыш параметрлері

rect\_x, rect\_y = 400, 300

rect\_width, rect\_height = 50, 50

speed = 5

# Негізгі цикл

running = True

while running:

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# Пернетақтаны тексеру

keys = pygame.key.get\_pressed()

if keys[pygame.K\_LEFT]:

rect\_x -= speed

if keys[pygame.K\_RIGHT]:

rect\_x += speed

if keys[pygame.K\_UP]:

rect\_y -= speed

if keys[pygame.K\_DOWN]:

rect\_y += speed

# Экранды толтыру және жаңарту

screen.fill(BLACK)

pygame.draw.rect(screen, GREEN, (rect\_x, rect\_y, rect\_width, rect\_height))

pygame.display.flip()

pygame.quit()

Бұл бағдарламада пернетақтаны қолдана отырып, төртбұрышты экранда жылжытуға болады.

**АНИМАЦИЯ**

import pygame

import sys

# Pygame-ді іске қосу

pygame.init()

# Экран параметрлері

WIDTH, HEIGHT = 800, 600

screen = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("Үй мен табиғат анимациясы")

# Түстер анықтамасы

sky\_blue = (135, 206, 250)

grass\_green = (34, 139, 34)

sun\_yellow = (255, 223, 0)

brown = (139, 69, 19)

white = (255, 255, 255)

black = (0, 0, 0)

gray = (169, 169, 169)

red = (255, 0, 0)

# Күн параметрлері

sun\_x, sun\_y = 100, 100

sun\_radius = 50

sun\_speed = 2

# FPS орнату

clock = pygame.time.Clock()

# Ойын циклі

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

# Күннің қозғалысы

sun\_x += sun\_speed

if sun\_x - sun\_radius > WIDTH:

sun\_x = -sun\_radius # Күн қайтадан шығады

# Фон суреттерін салу

screen.fill(sky\_blue) # Аспан

pygame.draw.rect(screen, GRASS\_GREEN, (0, HEIGHT // 2, WIDTH, HEIGHT // 2)) # Шөп

pygame.draw.circle(screen, SUN\_YELLOW, (sun\_x, sun\_y), sun\_radius) # Күн

# Тау

pygame.draw.polygon(screen, GRAY, [(200, 300), (400, 100), (600, 300)]) # Тау

pygame.draw.polygon(screen, WHITE, [(380, 140), (400, 100), (420, 140)]) # Тау басындағы қар

# Үй

pygame.draw.rect(screen, RED, (300, 400, 200, 150)) # Үйдің денесі

pygame.draw.polygon(screen, BROWN, [(300, 400), (500, 400), (400, 300)]) # Шатыр

pygame.draw.rect(screen, WHITE, (370, 450, 60, 100)) # Есік

# Ағаш

pygame.draw.rect(screen, BROWN, (150, 400, 30, 100)) # Сабақ

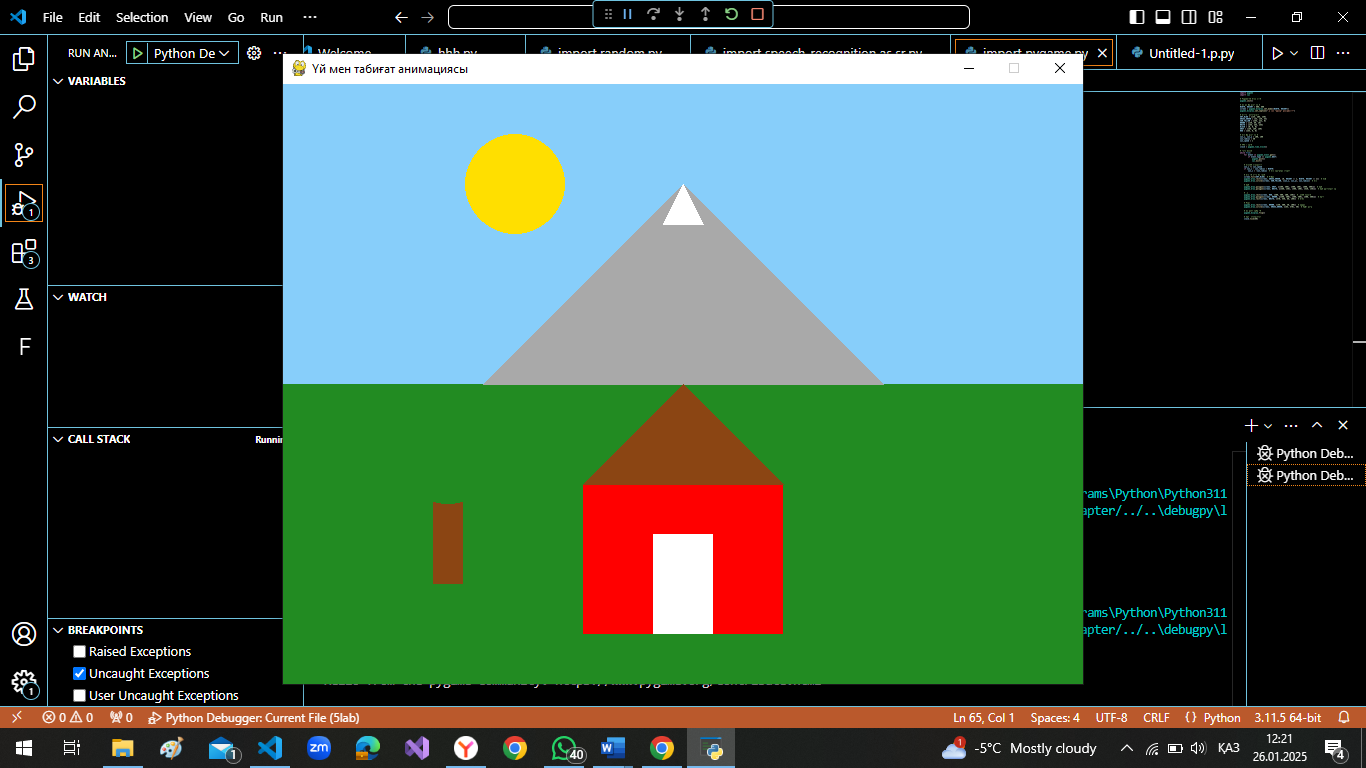
pygame.draw.circle(screen, GRASS\_GREEN, (165, 370), 50) # Жапырақтар

# Экранды жаңарту

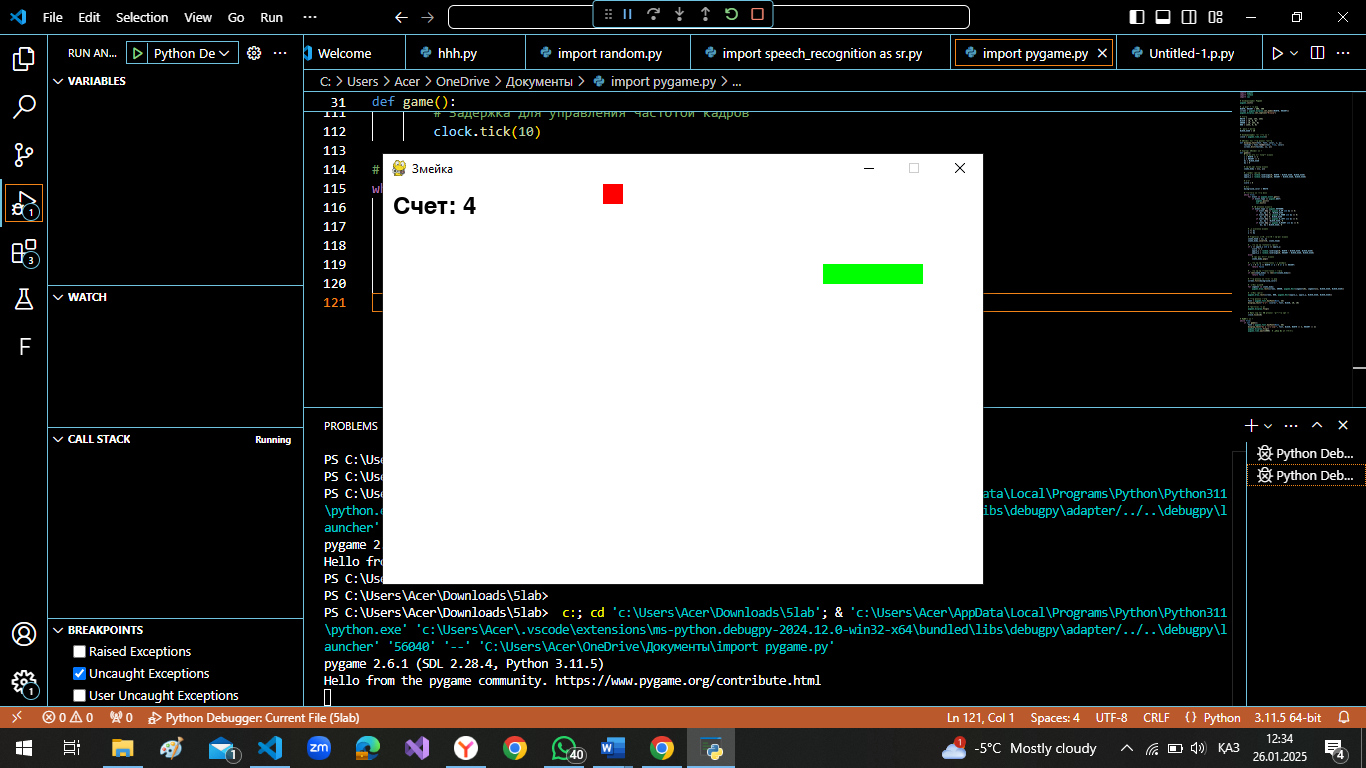
pygame.display.flip()

# Кадр жылдамдығы

clock.tick(60)



**Тапсырма: Қарапайым Snake ойынын жасау**



**Үй тапсырмасы:** Зертханалық жұмыс №1 орындау, қорғау