Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Спрайттар соқтығысуын анықтау, әзірлеу.**

(сабақ тақырыбы)

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

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_сәуір\_\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-1-22, 3БҚ-2-22

Сабақ түрі: Тәжірибелік

**2. Мақсаты, міндеттері:**

Оқу: Білімгерлерге PyGame кітапханасы негізіндеспрайттар соқтығысуын анықтау, әзірлеуді жүзеге асыру. 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

<https://translated.turbopages.org/proxy_u/ru-kk.ru.ebe5a83d-67fca360-fa400ee9-74722d776562/https/infourok.ru/prezentaciya-na-temu-upravlenie-s-pomoshyu-klaviatury-5823776.html>

<https://ust.kz/word/prezentaciya_keiipkerlerdi_animaciyalay_9synyp-203831.html>

***5. Сабақтың барысы: (90 минут)***

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

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

***“Бірлескен сауалнама” әдісі*** *(15 минут)*

**6. Жаңа тақырып**

Спрайттар соқтығысуын анықтау, әзірлеу.

Белгілі бір сценарий бойынша әзерленген ойынның кейіпкерлерін таңдап, оларды қозғалту және ойының нәтижелерін есептеп отыру кез келген ойынның негізгі мәселесі болып саналады. Кез келген ойынның түпкі идеясында ойыншылардың біреуі жеңу, біреуі жеңілу немесе тең ойнауы керек. Яғни ойын белгілі бір қорытынды нәтижеге жетуі тиіс. Мұндай болмаған жағдай қарастырылып жатқан программаны ойын деп қарауға болмайды. Өткен тақырыпта ойын алаңындағы графикалық нысандардың қозғалысымен таныстық. Ендігі мәселе белгілі бір ойын мақсатына сай кейіпкердің басқа кейіпкермен әсерлесуін анықтау жолдарын қарастырайық. Енді спрайтқа толық тоқталып өтейік. Спрайт екі негізгі қасиетке ие:

**image** –спрайт арқылы бейнеленген сурет

**rect** – спрайтты қамтитын тіктөртбұрышты аймақ

Сурет **Pygame** дегі функциялар көмегімен салынады немесе файлдан жүктеледі. **Pygame**кітапханасының sprite модулі бізге  **Sprite** класын береді. Әдетте программада класпен тікелей жұмыс істемейді. Бізде pygame.sprite.Sprite класымен тікелей жұмыс істемей, осы класқа негізделген Robots кіші класын құрайық.

**Def \_\_init\_\_(self, image\_file, location, speed)** қатарында алдынғы программа кодынан ерекшелігі, speed қосылған. Төмендегі катарларда роботтың бастапқы орнымен бірге, оның жылдамдығын да беріп отырады.

**Speed = [choice([-2, 2]), choice([-2,2])**

**Robot = Robots (img\_file, location, speed)**

**Pygame** – модулінде sprite класында spritecollide() функциясы спрайттардың соқтығысуын анықтайды. Спрайттардың соқтығысуын анықтау 3 кезеңнен тұрады.

1.      Спрайтты спрайттар тобынан өшіреміз.

2.      Бұл спрайттың топтағы басқа спрайттармен соқтығысуын тексереміз.

3.      Спрайтты қайта топқа қосамыз

**Код**

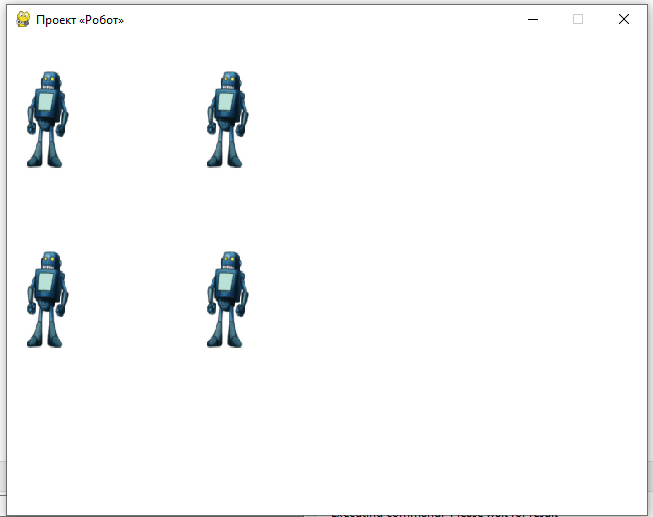
import pygame; pygame.init()  
from random import randint  
import pygame as pg  
import sys  
   
pg.time.set\_timer(pg.USEREVENT, 3000)  
   
W = 400  
H = 400  
WHITE = (255, 255, 255)  
ROBOTS = ('robot.png', 'robot.png', 'robot.png')

ROBOTS\_SURF = []

sc = pg.display.set\_mode((W, H))  
   
for i in range(len(ROBOTS)):  
    ROBOTS\_SURF.append(  
        pg.image.load(ROBOTS[i]).convert\_alpha())  
class Robot(pg.sprite.Sprite):  
    def \_\_init\_\_(self, x, surf, group):  
        pg.sprite.Sprite.\_\_init\_\_(self)  
        self.image = surf  
        self.rect = self.image.get\_rect(  
            center=(x, 0))   
      self.add(group)  
        self.speed = randint(1, 3)  
    def update(self):  
        if self.rect.y < H:  
            self.rect.y += self.speed  
        else:         
            self.kill()  
robots = pg.sprite.Group()

Robot(randint(1, W),  
    ROBOTS\_SURF[randint(0, 2)], robots)  
while 1:  
    for i in pg.event.get():  
        if i.type == pg.QUIT:  
            sys.exit()  
        elif i.type == pg.USEREVENT:  
            Robot(randint(1, W),  
                ROBOTS\_SURF[randint(0, 2)], robots)  
    sc.fill(WHITE)  
    robots.draw(sc)  
    pg.display.update()  
    pg.time.delay(20)  
    robots.update()

**Сурет**



**Сұрақтар**

1.      Спрайттың қандай қасиеттері бар?

2.      1-программа кодында Robots класын құру қалай іске асады?

3.      Animate() функциясының қызметі не?

4.      Спрайттардың соқтығысуын анықтау алгоритмі қанша қадамнан тұрады?

**1. Def \_\_init\_\_(self, image\_file, location, ???)**

Выберите один вариант из списка

1. Screen
2. Width
3. Speed
4. pip init pygame

**2. Спрайттар соқтығысуын анықтау**

class Robot(pg.sprite.Sprite):  
    def \_\_**???**\_\_(self, x, surf, group):  
        pg.sprite.Sprite.\_\_init\_\_(self)  
        self.image = surf  
        self.rect = self.image.get\_rect(  
            center=(x, 0))  
        self.add(group)  
        self.speed = randint(1, 3)

Выберите один вариант из списка

1. Pygame
2. Import
3. Init
4. pip

**3. есеп**

import **???**  
class Robots(pygame.sprite.Sprite):  
  def \_\_init\_\_(self, image\_file, location):  
     pygame.sprite.Sprite.\_\_init\_\_(self)  
     self.image = pygame.image.load('robot.png')  
     self.image = pygame.transform.scale(self.image, (80, 140))  
     self.rect = self.image.get\_rect()  
     self.rect.left, self.rect.top = location

Выберите один вариант из списка

1. Init
2. Def
3. Pip
4. pygame

**4-есеп**

while running:  
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
           running = False  
pygame.**???**

Выберите один вариант из списка

1. Import
2. quit()
3. pip
4. quit

**Практикалық тапсырмалар:**

1. **Pygame** немесе басқа графикалық платформада кейіпкердің жай жүріс анимациясын жасаңыз (кем дегенде 4 кадр).
2. Бір анимациялық қозғалыс циклін (мысалы: секіру, шабуыл, айналу) жасап, оны визуал түрде көрсетіңіз.

**Жаңа тақырыпты бекіту**

1. Анимациялаудың негізгі түрлерін атаңыз және әрқайсысына мысал келтіріңіз.
2. «Скелеттік анимация» және «Keyframe анимация» ұғымдарын салыстырып жазыңыз.
3. Қозғалыстың 12 қағидасынан 5-еуін таңдап, оларды түсіндіріңіз.
4. Анимацияда «anticipation» (алдын ала дайындық) не үшін қажет?
5. Кейіпкерді анимациялауға дайындық кезеңдерін атаңыз.

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы**

Өзіңе ұнаған тақырыпта ойын кейіпкерін таңдап, оны қозғалту программасын құрастыр