

Колледж цифровых технологий "Академия ТОП"

Проект:
Змейка на Pygame

Выполнил:
Студент 1 курса
Группы: 9/1-РПО-23/1
Осипов Михаил
Александрович
Руководитель проекта:
Ефимов Максим
Михайлович



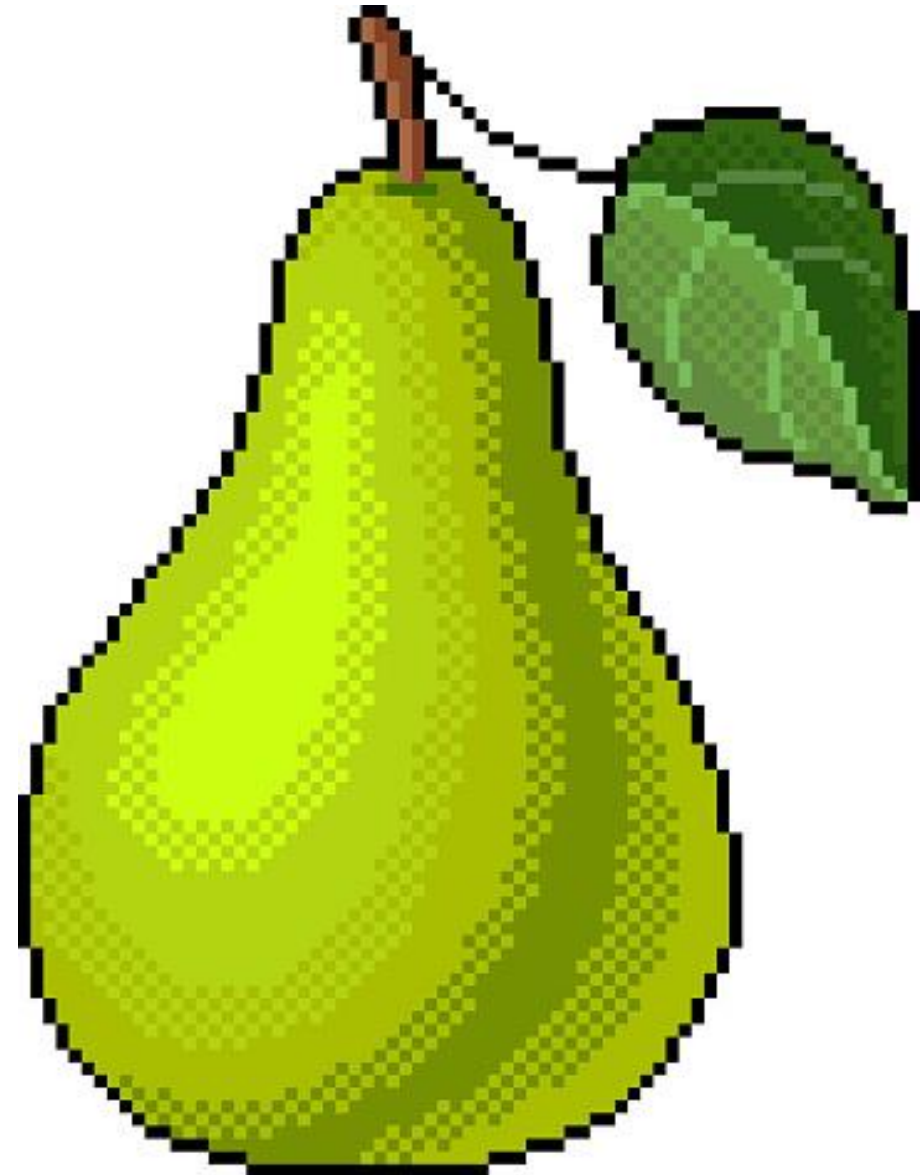
Введение:

Правила и краткое описание игры:

Змейка это жанр игр в котором игрок управляет змеей и его задача 1) съесть груши 2) не попадать на ту же клетку где хвост змеи 3) не сталкиваться со стенками.

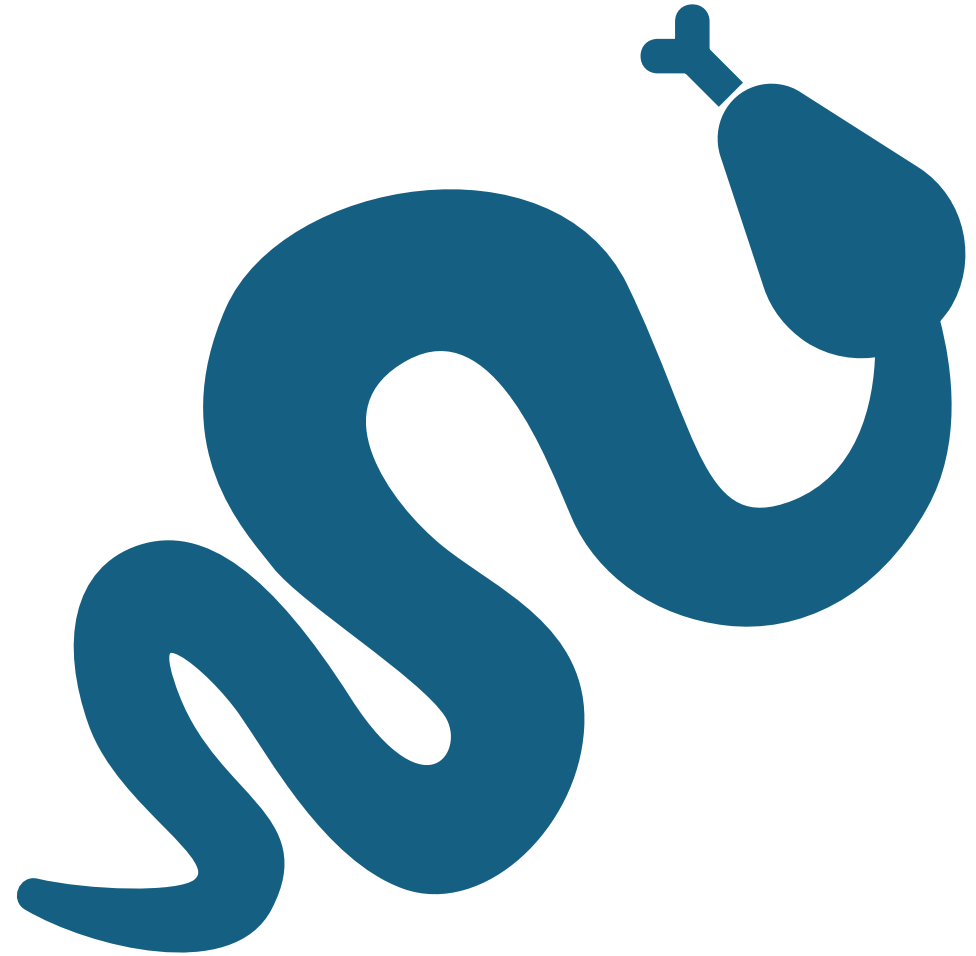
Что бы работала игра надо
переместить в её код
библиотеки Pygame и Random

- Pygame используется для создания игр и визуализации в то время Random - он нужен для работы с рандомными значениями например: случайным выбором элемента из списка (например выбрал грушу) или генерацией случайной числовой последовательности, элемента (например в какой то части карты заспавнилась груша).



Цвета, параметры экрана,
переменная для
координат змейки.

- За цвета отвечают переменные: `pear_color`, `snake_color`, `background`, `mesh_color`.
- За параметры экрана отвечают переменные: `screen_width`, `screen_height`, `mesh_size` и `fps`.
- Переменная для координат змейки:
`snake = [(x, y)]`



Создание окна и установка времени

`clock.tick(fps)` - используется для управления частотой обновления кадров в игровом цикле функция `tick()` указывает сколько кадров должно отображаться в секунду (в моём случае начальное "`fps = 5`" это означает что игра обновляется 5 раз в секунду увеличивая или уменьшая это значение я могу регулировать скорость игры)

`win = pygame.display.set_mode((screen_width, screen_height))` - создание окна

○

+



Отрисовка груши и Змейки на экране



- 1) pear_random: генерация случайной позиции для груши
- 2) drawing_pear: отрисовка груши на экране
- 3) drawing_snake: отрисовка змейки на экране
- 4) snake_pieces: перемещение частей змейки

Управление Змейкой с помощью Scroll, wasd:

если вы нажмёте клавишу влево `pygame.K_LEFT` и если текущее направление не является "RIGHT" то направление будет установлено на "LEFT" (потому что справа налево нельзя двигаться)

(так же работает и с wasd)

```
51     for event in pygame.event.get():
52         if event.type == pygame.QUIT:
53             pygame.quit()
54             quit()
55
56         elif event.type == pygame.KEYDOWN:
57             if event.key == pygame.K_LEFT and direction != 'RIGHT':
58                 direction = 'LEFT'
59             elif event.key == pygame.K_RIGHT and direction != 'LEFT':
60                 direction = 'RIGHT'
61             elif event.key == pygame.K_UP and direction != 'DOWN':
62                 direction = 'UP'
63             elif event.key == pygame.K_DOWN and direction != 'UP':
64                 direction = 'DOWN'
65             elif event.key == pygame.K_a and direction != 'RIGHT':
66                 direction = 'LEFT'
67             elif event.key == pygame.K_d and direction != 'LEFT':
68                 direction = 'RIGHT'
69             elif event.key == pygame.K_w and direction != 'DOWN':
70                 direction = 'UP'
71             elif event.key == pygame.K_s and direction != 'UP':
72                 direction = 'DOWN'
```

Увеличение скорости,
длина змейки при
съедании груши

При съедании груши (если условие **if pear == snake[0]:**(то есть координата головы змейки равняется координаты груши) выполнено) переменная **pear_check** становится равной **false** что приводит к вызову функции **drawing_snake ()** затем внутри условия **if pear_check == false:** добавляется новый элемент в список **snake** с помощью **snake.append(snake[0])** что увеличивает длину змейки.

Переменная **acceleration_counter** увеличивается на 1 fps только после 3 съеденных груш (1 груша = одному очку) и получается когда мы съедем 3 груши то вместо 5 фпс будет 6.

```
87     if pear_check == False:
88         pear = pear_random ()
89         pear_check = True
90         pear_counters += 1
91         snake.append(snake[0])
92         acceleration_counter += 1
93     elif pear_check == True:
94         drawing_pear (pear)
95         if snake[0] in snake[1:]:
96             pygame.quit()
97             quit()
98         if acceleration_counter == 3:
99             acceleration_counter = 0
100            fps += 1
101
102     if pear == snake[0]:
103         pear_check = False
104     drawing_snake ()
```


Проверка столкновений и условий завершения игры

Проверка столкновения змеи с самой собой

если координата змеи будет равняться какой то координаты тела то игра закроется.

```
95         if snake[0] in snake[1:]:
96             pygame.quit()
97             quit()
```

Условия завершения игры при выходе змеи за границы экрана: это отвечает за то что бы голова змеи не заходила за границы - стенки

```
106         if snake[0][0] - mesh_size < - mesh_size or snake[0][0] + mesh_size > screen_width:
107             pygame.quit()
108             quit()
109
110         if snake[0][1] - mesh_size < - mesh_size or snake[0][1] + mesh_size > screen_height:
111             pygame.quit()
112             quit()
```

Конец

