

**ОТЧЕТ**

по модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»

по специальности 09.02.03 «Программирование в компьютерных системах»

Выполнил студент группы П2-18

Сумин Константин Евгеньевич

\_\_\_\_\_\_\_\_\_\_ (подпись)

Принял преподаватель

Гусятинер Леонид Борисович

\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2021

**Оглавление**

[**Текстовое описание задачи** 3](#_Toc76155296)

[**Диаграмма прецедентов** 4](#_Toc76155297)

[**Диаграмма классов** 5](#_Toc76155298)

[**Определения классов по диаграмме** 6](#_Toc76155299)

[Листинг 1 setting (wight, position,sizr\_check,no\_wight) 6](#_Toc76155300)

[Листинг 2 draw\_circle (wight) 6](#_Toc76155301)

[Листинг 3 menu (position, wight) 6](#_Toc76155302)

[**Проект с классами для выполнения задания.** 8](#_Toc76155303)

[Листинг 1 Главный модуль. 8](#_Toc76155304)

[**Отладку одного из модулей при разработке** 10](#_Toc76155305)

[**Тестовые наборы и тестирование одного из модулей.** 12](#_Toc76155306)

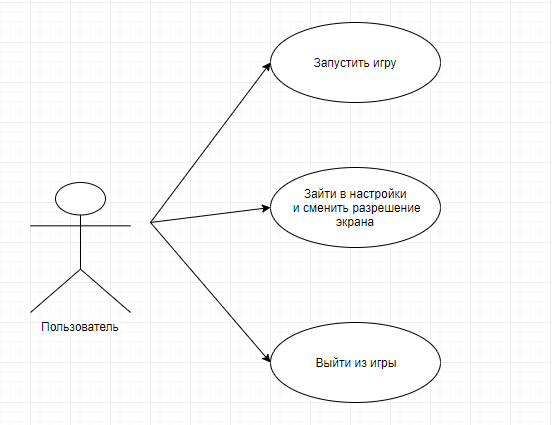
[**Значение временной сложности O для одного из методов. Значение O для оптимального кода.** 13](#_Toc76155307)

# **Текстовое описание задачи**

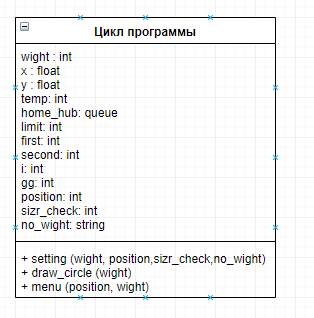
Сделать игру в которой объект, похожий на НЛО, должен будет лететь и пытаться не врезаться в встречающиеся ему на пути объекты.

|  |  |
| --- | --- |
| ESC | Выход из игры |
| UP | Движение вверх |
| DOWN | Движение вниз |
| LEFT | Движение влево |
| RIGHT | Движение вправо |

# **Диаграмма прецедентов**



# **Диаграмма классов**



# **Определения классов по диаграмме**

## Листинг 1 setting (wight, position,sizr\_check,no\_wight)

функция переключения на экран настроек, а так же работы с этим экраном. А именно сканирование движение курсора и ввод нужного разрешения экрана.

def setting(wight, position,sizr\_check,no\_wight):  
 font = pygame.font.SysFont('couriernew', int(wight \* 0.06))  
 size = font.render(str('Size:'), True, THECOLORS['green'] if (sizr\_check == 1) else THECOLORS['pink'])  
 screen.blit(size, (wight \* 0.4, wight \* 0.2))  
 if event.type == pygame.MOUSEMOTION and event.pos[0] > wight \* 0.4 and event.pos[0] < wight \* 0.57 and event.pos[1] > wight \* 0.2 and event.pos[1] < wight \* 0.25:  
 sizr\_check = 1  
 if event.type == pygame.KEYUP and event.key == pygame.K\_RETURN:  
 position = 0  
 wight = INTEGER(no\_wight,wight)  
 no\_wight = ''  
 elif event.type == pygame.KEYUP and event.key == pygame.K\_BACKSPACE:  
 no\_wight = no\_wight[:-1]  
 elif event.type == pygame.KEYUP:  
 no\_wight += event.unicode  
 size = font.render(str(no\_wight), True, THECOLORS['pink'])  
 screen.blit(size, (wight \* 0.6, wight \* 0.2))  
 return wight, position,sizr\_check,no\_wight

## Листинг 2 draw\_circle (wight)

отрисовка круга (UFO) и внутреннего круга

def draw\_circle(wight):  
 pygame.draw.circle(screen, THECOLORS['gray'], (x, y), wight \* 0.07)  
 pygame.draw.circle(screen, (100, 0, 245), (x, y), (wight \* 0.07 \* 0.5))

## Листинг 3 menu (position, wight)

отрисовка меню, а так же работы с меню. А именно сканирование движение курсора.

def menu(position, wight):  
 font = pygame.font.SysFont('couriernew', int(wight \* 0.06))  
 start = font.render(str('START'), True, THECOLORS['green'] if (position == 1) else THECOLORS['pink'])  
 settings = font.render(str('SETTINGS'), True, THECOLORS['green'] if (position == 2) else THECOLORS['pink'])  
 exitt = font.render(str('EXIT'), True, THECOLORS['green'] if (position == 3) else THECOLORS['pink'])  
 screen.blit(start, (wight \* 0.4, wight \* 0.2))  
 screen.blit(settings, (wight \* 0.35, wight \* 0.3))  
 screen.blit(exitt, (wight \* 0.425, wight \* 0.4))  
 # event.type = pygame.MOUSEBUTTONDOWN, event.button = 3, event.pos = (100, 100).  
  
 if event.type == pygame.MOUSEMOTION and event.pos[0] > wight \* 0.4 and event.pos[0] < wight \* 0.57 and event.pos[  
 1] > wight \* 0.4 and event.pos[1] < wight \* 0.45:  
 position = 3  
 elif event.type == pygame.MOUSEBUTTONUP and event.pos[0] > wight \* 0.4 and event.pos[0] < wight \* 0.57 and \  
 event.pos[1] > wight \* 0.4 and event.pos[1] < wight \* 0.45:  
 position = 33  
 elif event.type == pygame.MOUSEMOTION and event.pos[0] > wight \* 0.4 and event.pos[0] < wight \* 0.57 and event.pos[  
 1] > wight \* 0.2 and event.pos[1] < wight \* 0.25:  
 position = 1  
 elif event.type == pygame.MOUSEBUTTONUP and event.pos[0] > wight \* 0.4 and event.pos[0] < wight \* 0.57 and \  
 event.pos[1] > wight \* 0.2 and event.pos[1] < wight \* 0.25:  
 position = 11  
 elif event.type == pygame.MOUSEMOTION and event.pos[0] > wight \* 0.33 and event.pos[0] < wight \* 0.65 and event.pos[  
 1] > wight \* 0.3 and event.pos[1] < wight \* 0.35:  
 position = 2  
 elif event.type == pygame.MOUSEBUTTONUP and event.pos[0] > wight \* 0.33 and event.pos[0] < wight \* 0.65 and event.pos[  
 1] > wight \* 0.3 and event.pos[1] < wight \* 0.35:  
 position = 22  
 return position

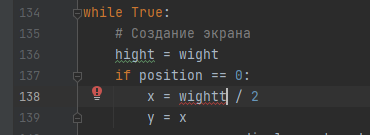
# **Проект с классами для выполнения задания.**

## Листинг 1 Главный модуль.

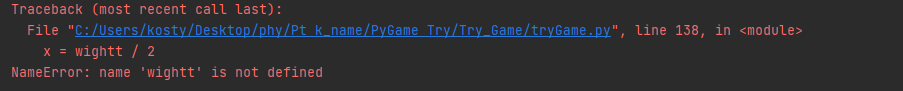
while True:  
 # Создание экрана  
 hight = wight  
 if position == 0:  
 x = wight / 2  
 y = x  
 screen = pygame.display.set\_mode((wight, hight))  
 # Ограничение FPS  
 pygame.time.delay(10)  
 # Условия выключения приложения  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT or event.type == pygame.KEYUP and event.key == pygame.K\_ESCAPE:  
 pygame.quit()  
 sys.exit()  
 # Проигрыш  
 if gg == 1:  
 game\_over(gg)  
 if event.type == pygame.MOUSEBUTTONUP:  
 hight = wight  
 x = wight / 2  
 y = x  
 temp = 0  
 home\_hub = queue.Queue()  
 limit = 0  
 first = 0  
 second = 0  
 i = 0  
 gg = 0  
 position = 0  
 sizr\_check=0  
 continue  
  
 keys = pygame.key.get\_pressed()  
  
 # Создание меню  
 if position == 22:  
 wight, position,sizr\_check,no\_wight = setting(wight, position,sizr\_check,no\_wight)  
 elif position == 33:  
 pygame.quit()  
 sys.exit()  
 elif position == 11:  
  
 # Отрисовка круга  
 draw\_circle(wight)  
  
 # Создание преград  
 temp += 1  
 if temp == 20:  
 rx = 0  
 ry = random.randrange(round(wight \* 0.07), round(wight - (wight \* 0.08)))  
 temp = random.randrange(-40, -20)  
 if limit != 5:  
 limit += 1  
 home\_hub.put(ry)  
 home\_hub.put(rx)  
 # Отрисовка и обработка преград  
 if not home\_hub.empty():  
 for i in range(limit):  
 first = home\_hub.get()  
 second = home\_hub.get()  
 if first != 0 and second != first:  
 pygame.draw.rect(screen, THECOLORS['gray'], (first, second, wight \* 0.1, wight \* 0.1))  
  
 # Сканирование столкновений преград с кругом  
 # (X - Xo)^2 +(Y - Yo)^2 <= R^2  
  
 if (first - x) \*\* 2 + (second - y) \*\* 2 <= (wight \* 0.07) \*\* 2 or (first + wight \* 0.1 - x) \*\* 2 + (  
 second - y) \*\* 2 <= (wight \* 0.07) \*\* 2 or (first - x) \*\* 2 + (  
 second + wight \* 0.1 - y) \*\* 2 <= (  
 wight \* 0.07) \*\* 2 or (first + wight \* 0.1 - x) \*\* 2 + (second + wight \* 0.1 - y) \*\* 2 <= (  
 wight \* 0.07) \*\* 2:  
 gg = 1  
 # Уничтожение преград, что вышли за экран  
 if second < wight:  
 home\_hub.put(first)  
 home\_hub.put(second + wight \* 0.004)  
 else:  
 limit -= 1  
  
 # Движение круга  
 x, y = move(x, y)  
 elif position != 11:  
 position = menu(position, wight)  
  
 pygame.display.flip()

# **Отладку одного из модулей при разработке**

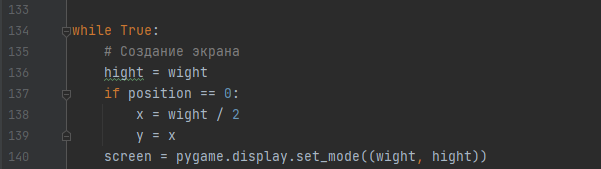
Код до отладки:



Ошибка:



Код после отладки:



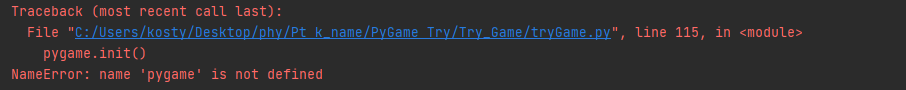
Запуск программы:



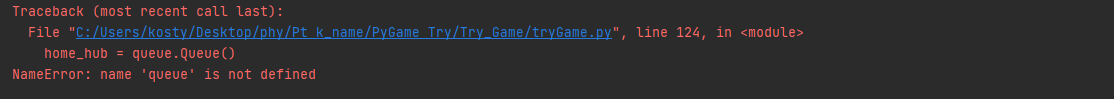
# **Тестовые наборы и тестирование одного из модулей.**

|  |  |
| --- | --- |
| Тест 1 Отсутствие import pygame | Ошибка |
| Тест 2 Отсутствие import queue | Ошибка |

Тест 1



Тест 2



# **Значение временной сложности O для одного из методов. Значение O для оптимального кода.**

Главный модуль является O(1), т.к. алгоритм не должен пройти все строки БД, и все данные заранее введены.