



алгоритмика

# PYTHON

Занятие 23

# Сегодня на уроке

---

- Взаимодействие спрайтов
- Артефакты
- Усложнение игры



# Вопросы

---

1. Как различать спрайты одного класса?
2. Как отследить взаимодействие спрайтов?
3. Для чего используется переменная `game_over` в игре?
4. Какие действия останавливались, если `game_over=0`?
5. Из чего состоит цвет RGB палитры?
6. Откуда начинается отсчет координат в окне приложения?
7. Что такое случайное значение?

# Размещение спрайта в игровом окне

Создать объект класса в указанном месте

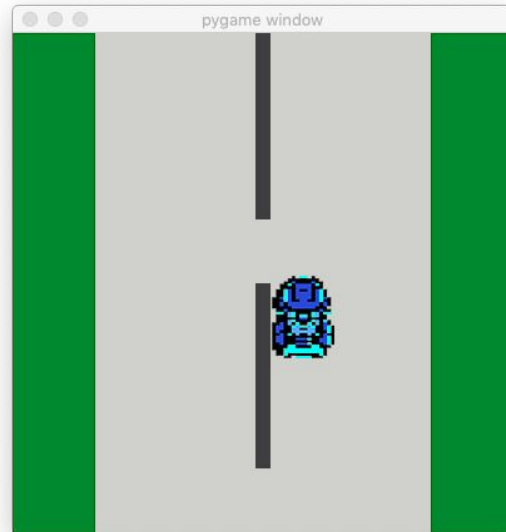
1 `car1 = Car(randint(25,375), 'Car1.png')`

Задать координату у для левого верхнего угла

2 `car1.rect.y=0`

Разместить объект сверху прямоугольника по размеру загруженной картинки

3 `sc.blit(car1.image, car1.rect)`



# Добавление группы

```
class Game_sprite(pygame.sprite.Sprite):  
    def __init__(self, x, filename, group):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.image.load(filename).convert_alpha()  
        self.rect = self.image.get_rect(center=(x, 0))  
        self.add(group)  
  
car1 = Game_sprite(100, 'Car1.png', user_car)
```

Группа для спрайта будет определена в момент создания объекта класса

# Взаимодействие спрайтов

---

Метод **`spritecollideany()`** проверяет взаимодействие конкретного спрайта с любым из спрайтов из группы. Функция принимает первым аргументом спрайт, который проверяется, вторым – группу.

```
pygame.sprite.spritecollideany(имя_спрайта, имя_группы_взаимодействия)
```

Проверяет взаимодействие гоночной машины и противника

```
pygame.sprite.spritecollideany(car1, cars)
```

# Вопрос

---

В каком случае наступает Game over?



# Вопрос-ответ

---

В каком случае наступает Game over?

В случае  
столкновения машин

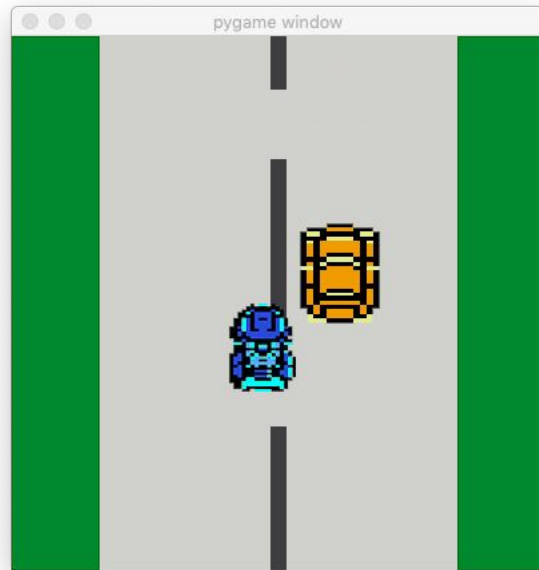




# Вопрос

---

Как долго может двигаться автомобиль в реальной жизни?



# Вопрос-ответ

---

Как долго может двигаться автомобиль в реальной жизни?

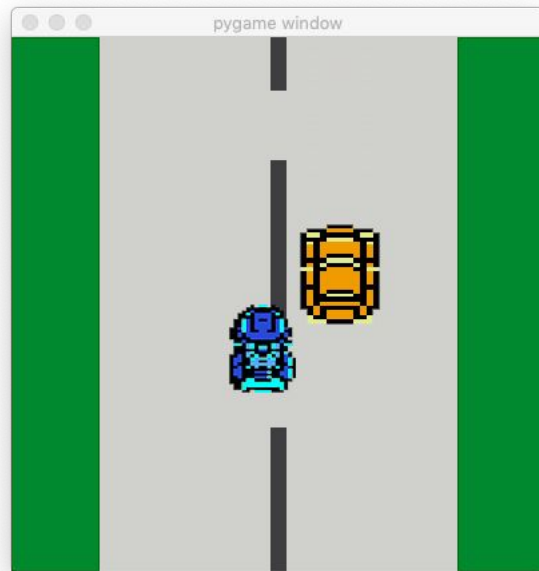
Пока не закончится  
топливо



# Вопрос

---

Что такое артефакт в компьютерных играх?



# Вопрос-ответ

---

Что такое артефакт в компьютерных играх?

Это предмет, который  
дает преимущество  
герою в чем-либо



# Вопрос

---

Какой артефакт помог бы восполнить запас топлива?



# Вопрос

---

Какой артефакт помог бы восполнить запас топлива?

Канистра/Бензоколонка



# Задание

---

Добавьте в игру спрайт **canister** с группой **canister\_group**. Изображение для файла возьмите из файл **canister.png**.



canister.png

# Задание. Решение

---

Добавьте в игру спрайт **canister** с группой **canister\_group**. Изображение для файла возьмите из файл **canister.png**.



canister.png

```
canister_group = pygame.sprite.Group()
```

```
canister=Game_sprite(200,'canister.png',canister_group)
```

```
canister.rect.y=0
```



# Вопрос

---

Как гоночная машина может получить топливо?



# Вопрос-ответ

---

Как гоночная машина может получить топливо?

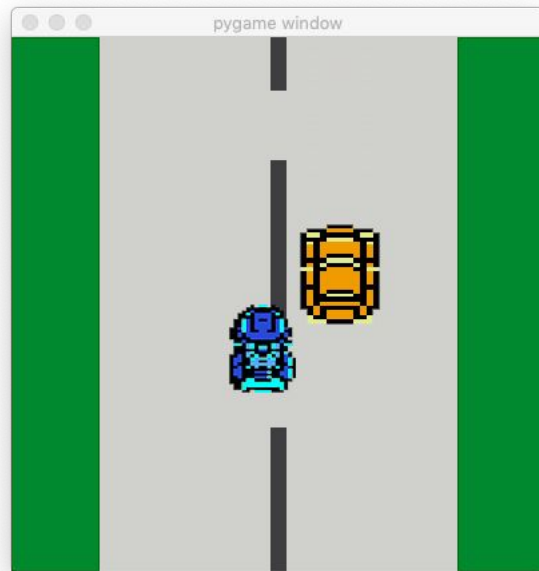
Поймать его



# Вопрос

---

Как должен двигаться спрайт canister?



# Вопрос

---

Как должен двигаться спрайт canister?



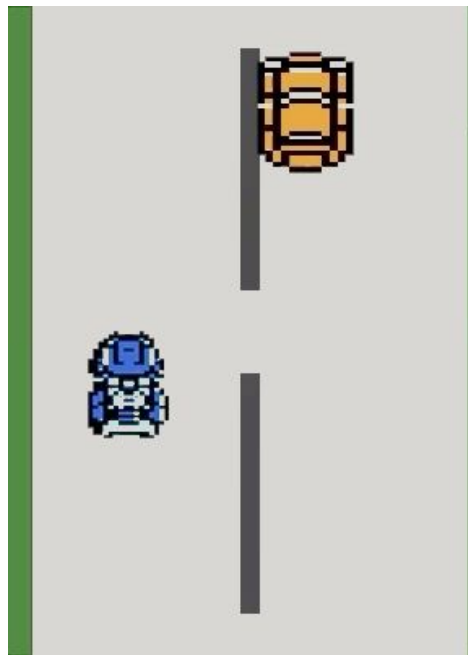
canister.png

# Вопрос-ответ

---

Как должен двигаться спрайт canister?

Сверху-вниз



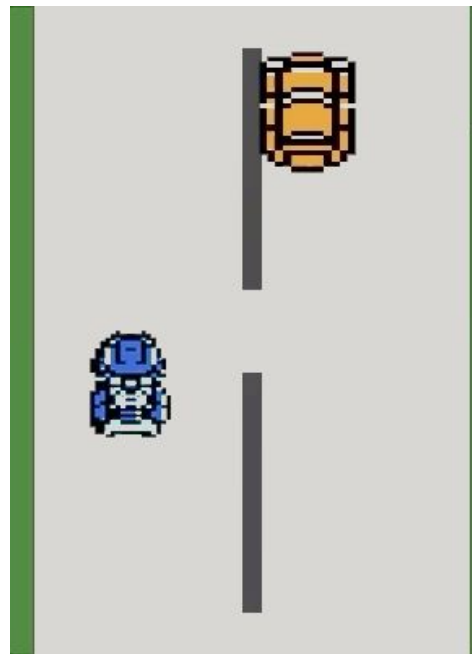
canister.png

# Вопрос

---

Как должен двигаться спрайт canister?

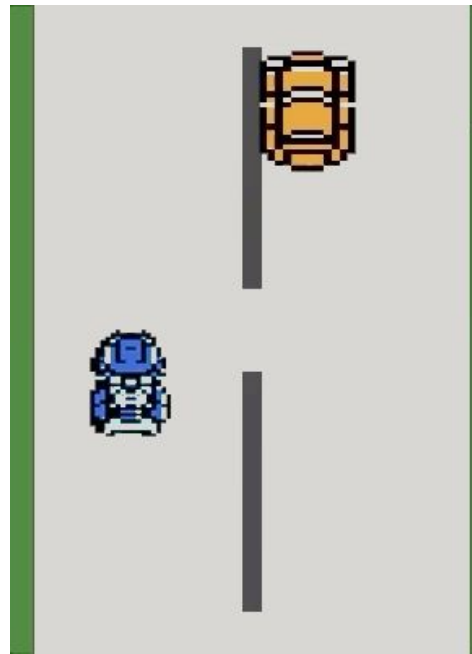
Сверху-вниз



# Вопрос

---

В каком месте на игровом окне может двигаться спрайт машины-противника?



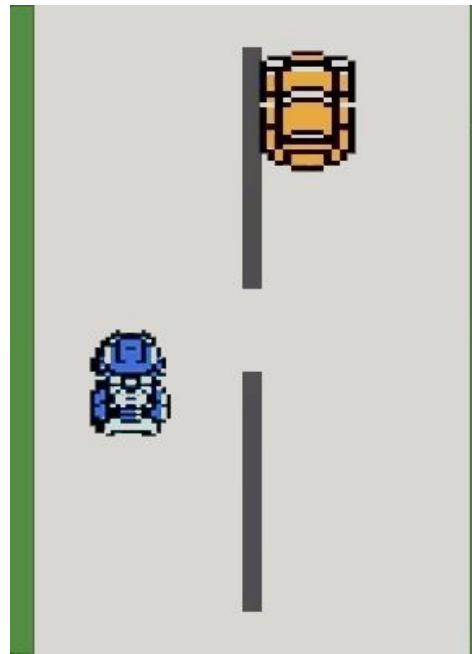
# Вопрос-ответ

---

В каком месте на игровом окне может двигаться спрайт машины-противника?

По ширине  
дороги

```
car2 = Game_sprite(randint(100,300),'Car2.png',cars)
```





# Задание

---

Добавьте движение спрайту canister. Спрайт должен появляться только в районе дороги.



canister.png

# Задание. Решение

---

Добавьте движение спрайту canister. Спрайт должен появляться только в районе дороги.

```
sc.blit(canister.image,canister.rect)
```

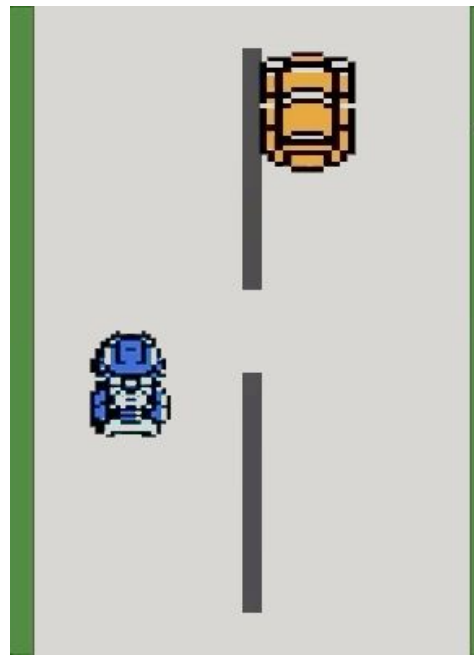
```
if canister.rect.y<H:  
    canister.rect.y=canister.rect.y+2  
else:  
    canister.rect.y=0
```



# Вопрос

---

С каким спрайтом будет взаимодействовать спрайт canister?

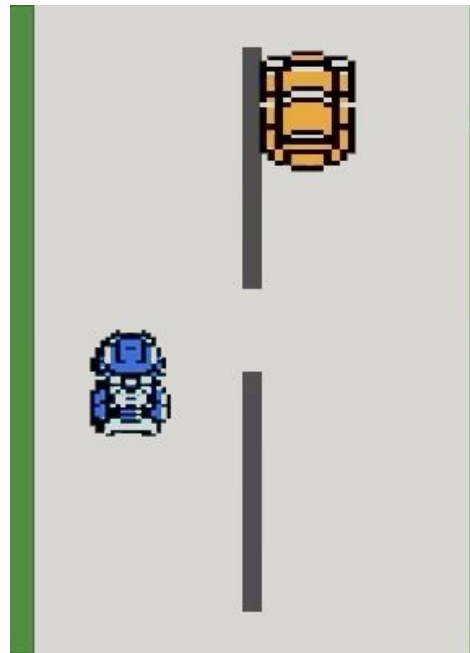


# Вопрос-ответ

---

С каким спрайтом будет взаимодействовать спрайт canister?

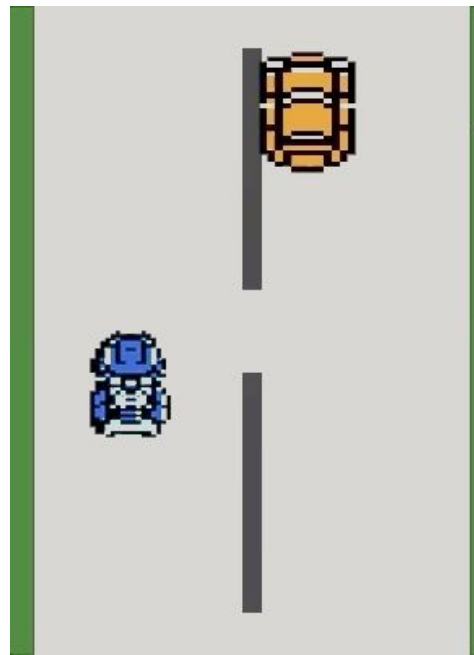
Гоночной  
машиной (car1)



# Вопрос

---

Что должно произойти после касания спрайтов?

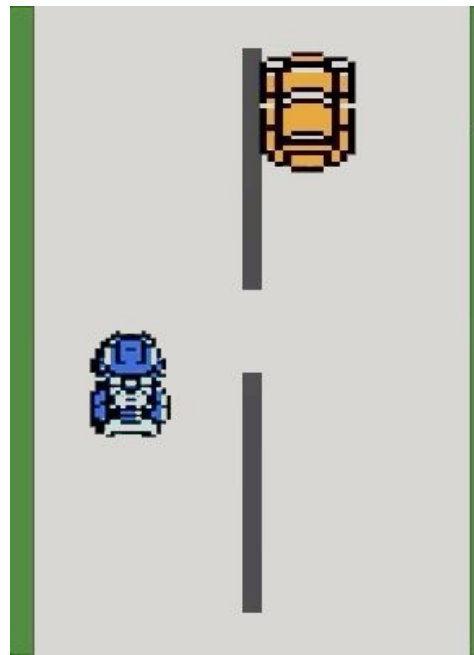


# Вопрос-ответ

---

Что должно произойти после касания спрайтов?

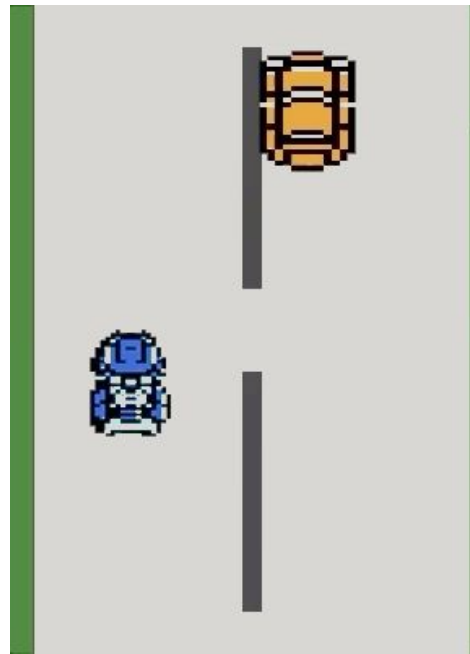
Канистра должна  
пропасть и снова начать  
движение сверху вниз



# Задание

---

Добавьте проверку условия касания спрайта car1 и canister\_group.

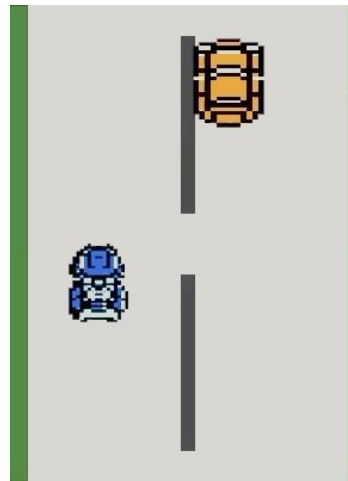


# Задание. Решение

---

Добавьте проверку условия касания спрайта car1 и canister\_group.

```
if pygame.sprite.spritecollideany(car1, canister_group):  
    canister.rect.y = 0
```

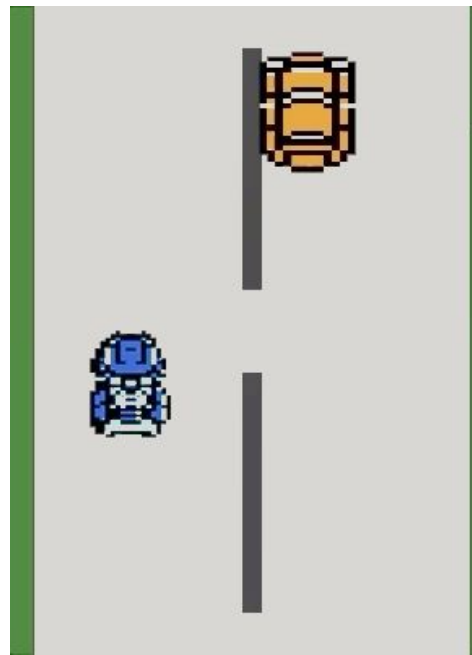




# Вопрос

---

С какой высоты начинается движение канистры?

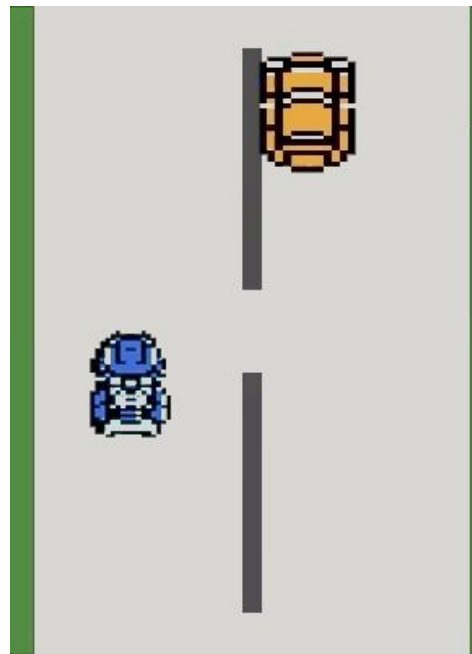


# Вопрос-Ответ

---

С какой высоты начинается движение канистры?

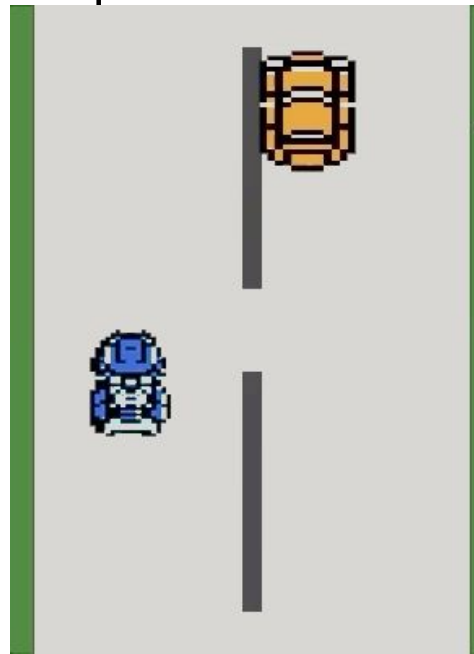
$$y=0$$



# Вопрос

---

Как изменить значение высоты, чтобы оно изменялось и было случайным для игрока?

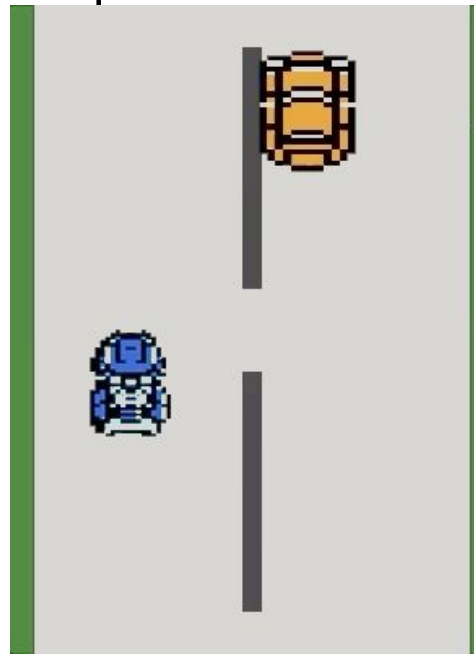


# Вопрос-Ответ

---

Как изменить значение высоты, чтобы оно изменялось и было случайным для игрока?

Использовать  
случайное значение.

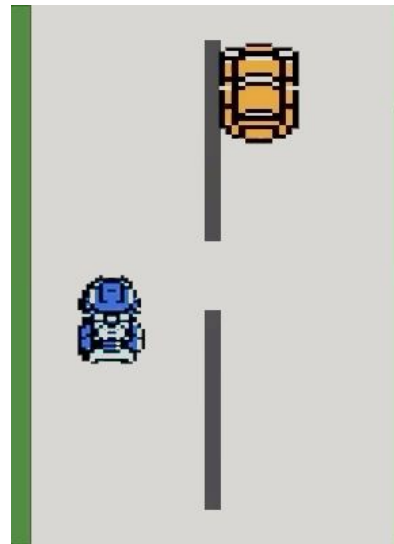


# Задание

---

Измените высоту падения канистры, чтобы её появление в игровом окне не было известно заранее игроку.

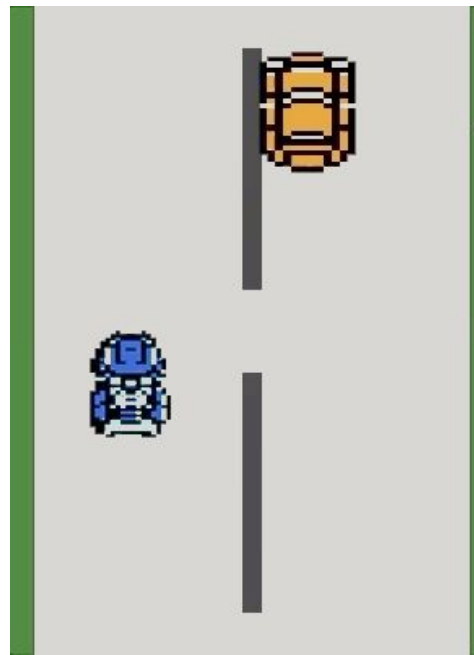
```
if canister.rect.y < H:  
    canister.rect.y = canister.rect.y + 2  
  
else:  
    canister.rect.y = 0 - randint(200, 1000)
```



# Вопрос

---

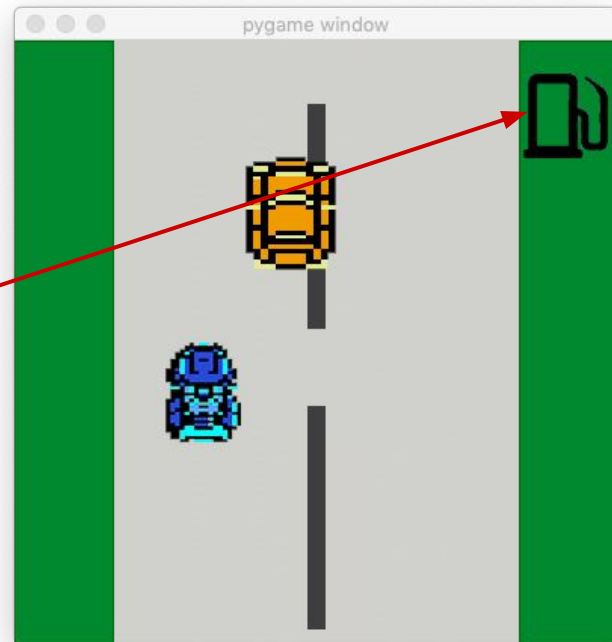
Как игроку узнать, сколько топлива осталось у гоночной машины?



# Вопрос-ответ

Как игроку узнать, сколько топлива осталось у гоночной машины?

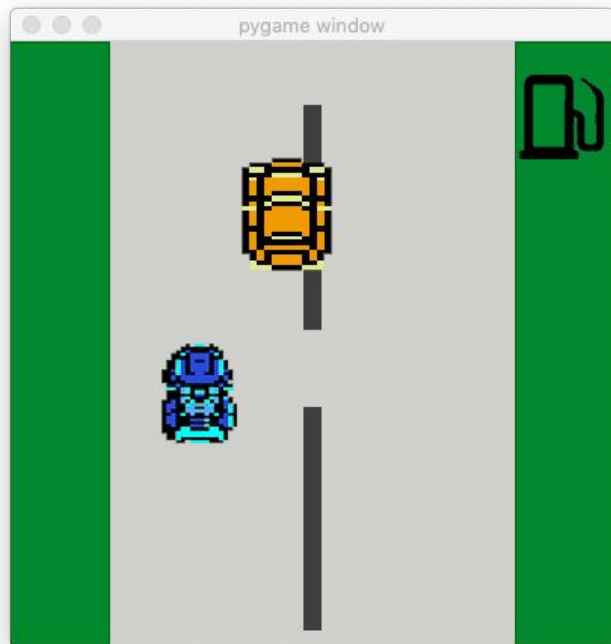
Необходимо добавить индикатор, который будет показывать уровень топлива



# Задание

---

Добавьте спрайт **fuel** с группой **fuel\_group** из файла `fuel.png`. Разместите спрайт, как показано на рисунке.





# Задание. Решение

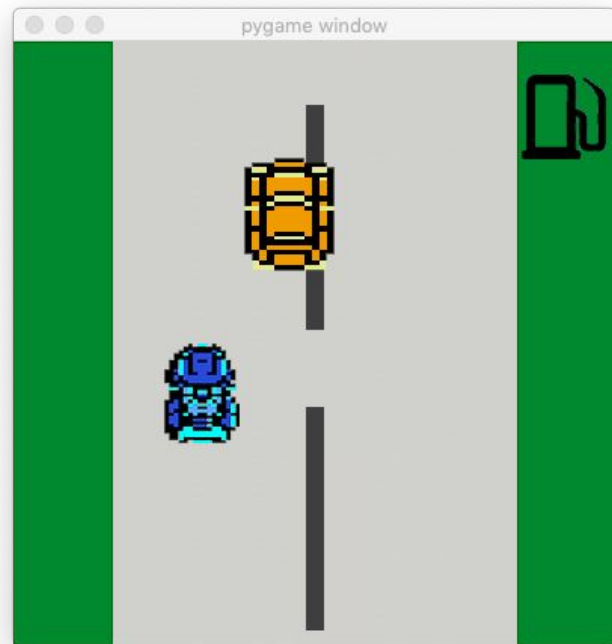
---

Добавьте спрайт **fuel** с группой **fuel\_group** из файла **fuel.png**. Разместите спрайт, как показано на рисунке.

```
fuel_group = pygame.sprite.Group()
```

```
fuel=Game_sprite(365,'fuel.png',fuel_group)
```

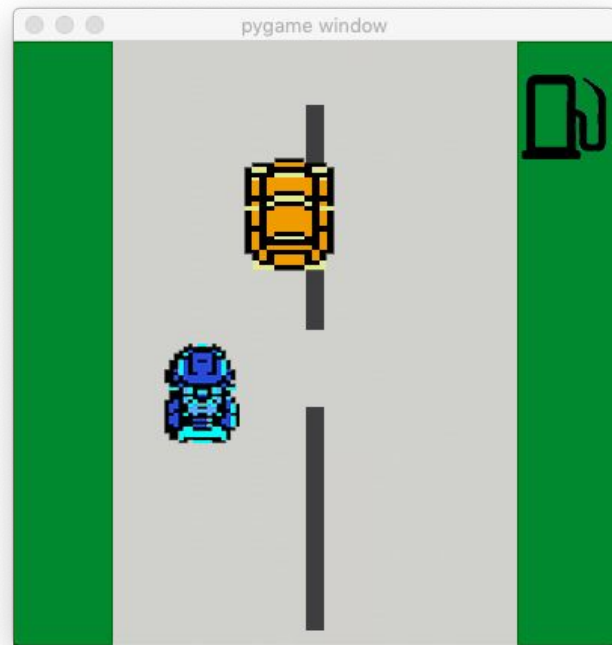
```
sc.blit(fuel.image,fuel.rect)
```



# Вопрос

---

Каким должен быть бензобак, если он полностью будет заполнен топливом?

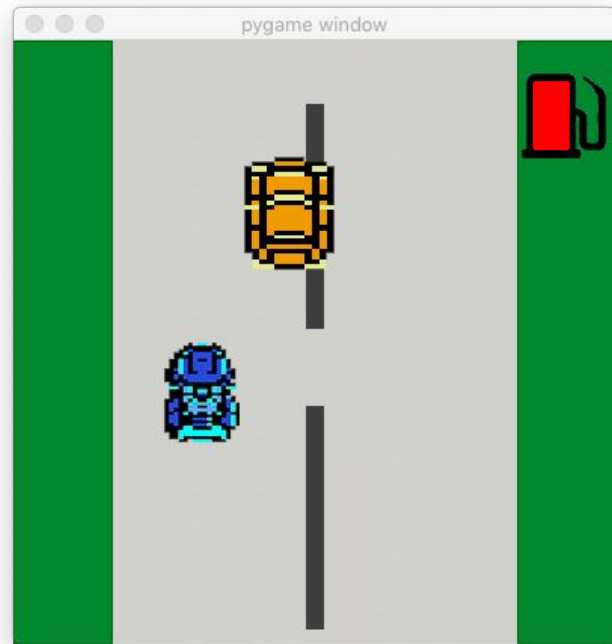


# Вопрос-ответ

---

Каким должен быть бензобак, если он полностью будет заполнен топливом?

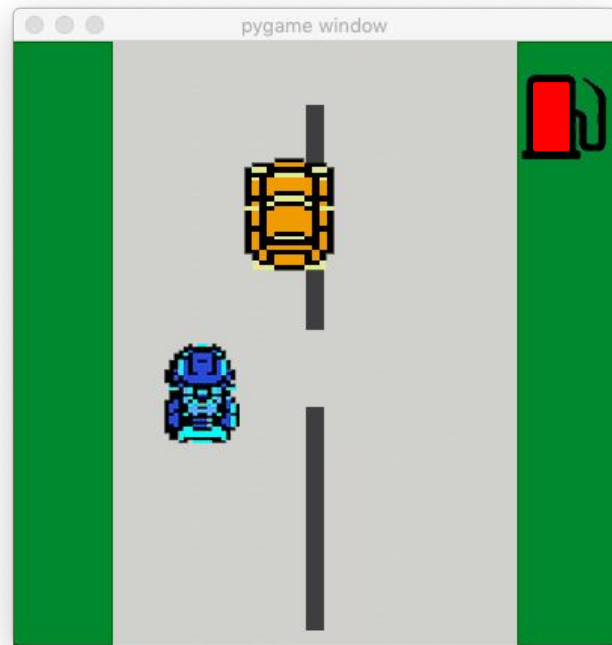
Заполненные каким-  
нибудь цветом



# Вопрос

---

Каким должен быть бензобак, если он будет заполнен наполовину топливом?

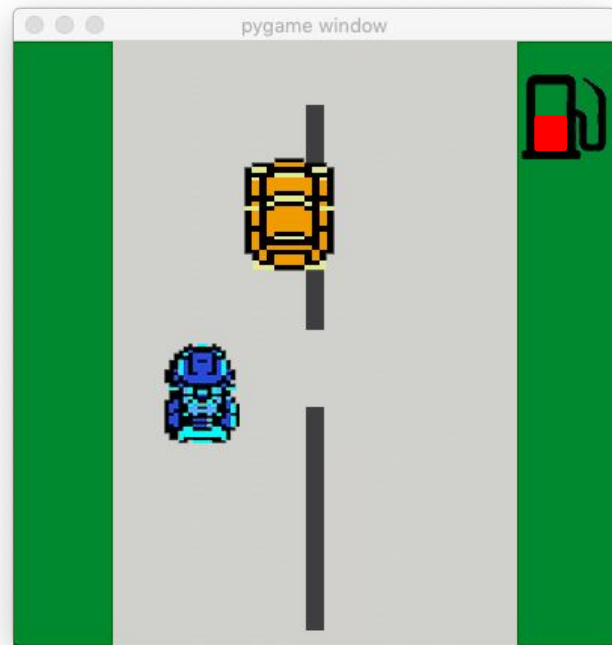


# Вопрос-ответ

---

Каким должен быть бензобак, если он будет заполнен наполовину топливом?

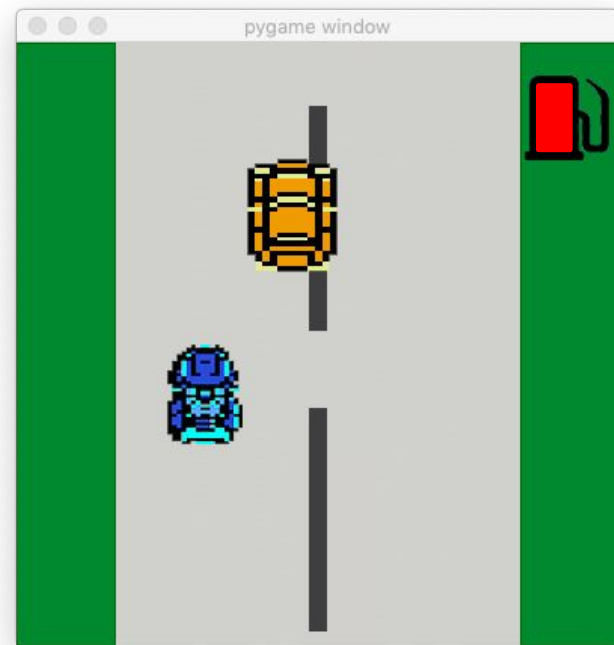
Краски должно стать  
наполовину меньше



# Вопрос

---

Какой фигурой мы обозначаем уровень топлива в бензобаке?

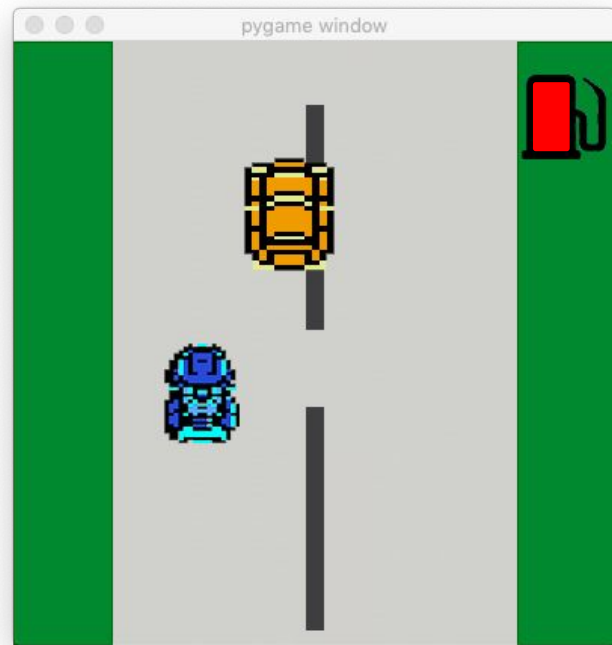


# Вопрос-ответ

---

Какой фигурой мы обозначаем уровень топлива в бензобаке?

Прямоугольником



# Создание фигур в pygame

---

Функция `draw.rect()` модуля `pygame` создает прямоугольник с заданными аргументами

```
pygame.draw.rect(игровое_окно, (Цвет RGB), (x,y,width,height))
```

**Игровое окно** – имя окна, в котором необходимо нарисовать прямоугольник

**(Цвет RGB)** – кортеж значения цвета RGB палитры

**(x, y, width, height)** – кортеж значений прямоугольника

**x, y** – координаты верхнего левого угла прямоугольника

**width** – ширина прямоугольника

**height** – высота прямоугольника



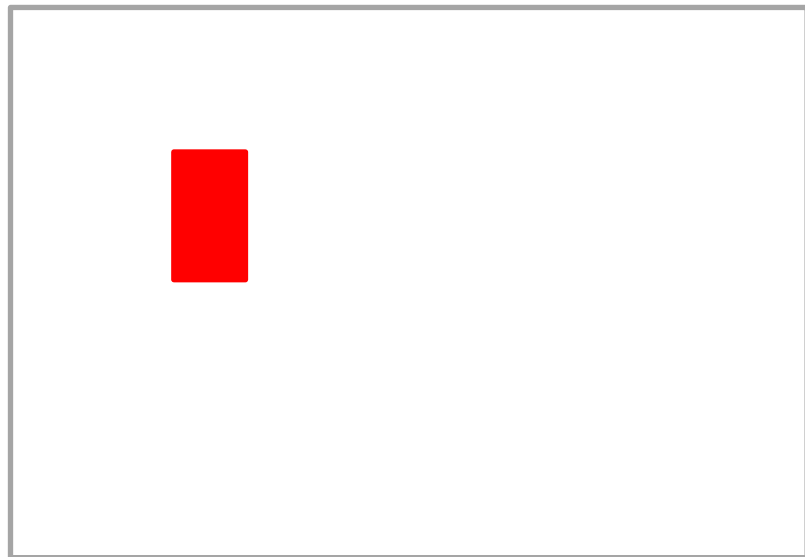


# Создание фигур в pygame

---

```
pygame.draw.rect(sc, (255,0,0), (100,100,20,50))
```

В игровом окне **sc**  
нарисует **красный**  
прямоугольник с началом  
в координатах **(100,100)**,  
шириной **20** и высотой **50**



# Задание

---

Добавьте в программу создание прямоугольника, который бы полностью заполнил бензобак

```
pygame.draw.rect(sc, (255,0,0), (x,y,width,height))
```



# Задание. Решение

---

Добавьте в программу создание прямоугольника, который бы полностью заполнил бензобак

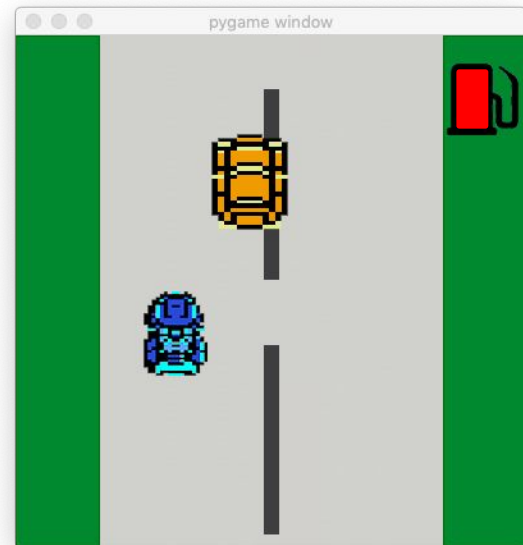
```
pygame.draw.rect(sc, (255,0,0), (x,y,width,height))
```

```
sc.blit(car2.image,car2.rect)
```

```
pygame.draw.rect(sc,(255,0,0), (346,28,22,42))
```

```
sc.blit(canister.image,canister.rect)
```

```
sc.blit(fuel.image,fuel.rect)
```

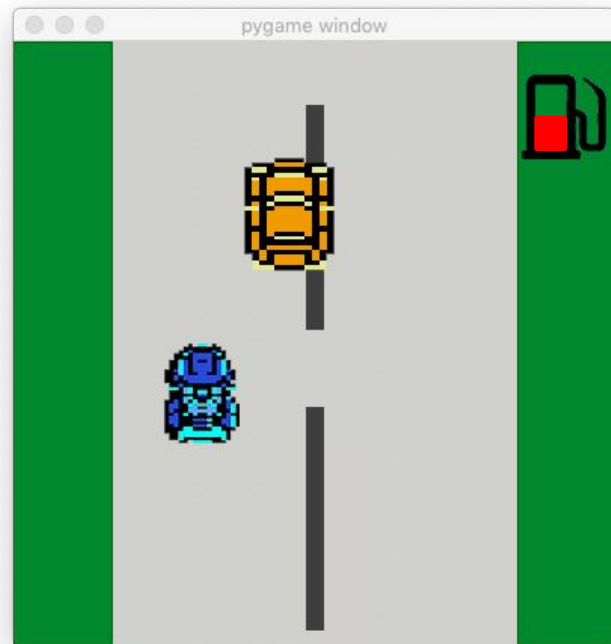


# Вопрос

---

Как изменятся аргументы метода `rect`, если бензобак будет заполнен наполовину?

```
pygame.draw.rect(sc,(255,0,0), (346,28,22,42))
```



# Вопрос-Ответ

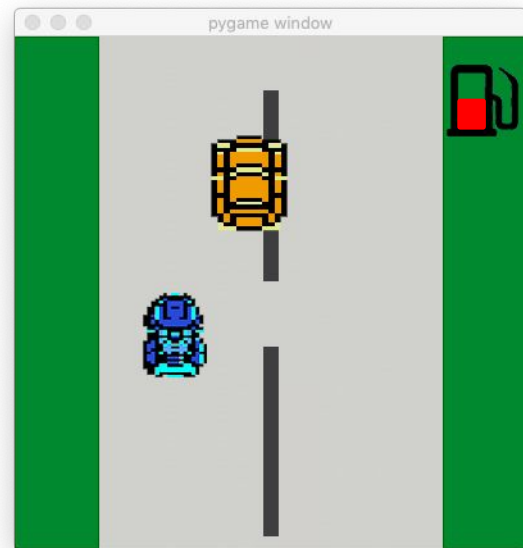
Как изменятся аргументы метода `rect`, если бензобак будет заполнен наполовину?

```
pygame.draw.rect(sc,(255,0,0), (346,28,22,42))
```



```
pygame.draw.rect(sc,(255,0,0), (346,56,22,21))
```

Изменится координата **у** и  
**высота** прямоугольника



# Вопрос

---

На сколько отличается значение координаты  $y$  от высоты прямоугольника?

```
pygame.draw.rect(sc,(255,0,0), (346,28,22,42))
```



# Вопрос-Ответ

---

На сколько отличается значение координаты y от высоты прямоугольника?

```
pygame.draw.rect(sc,(255,0,0), (346,28,22,42))
```

$$42-28=14$$

на 14



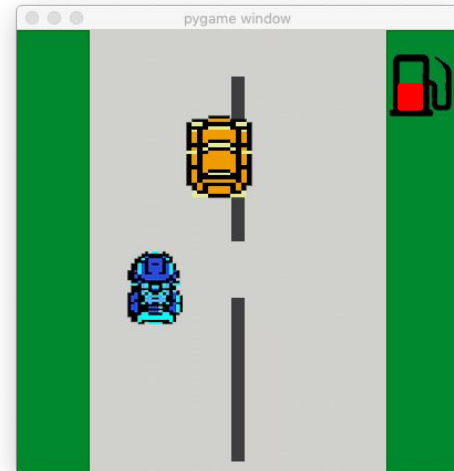
# Динамическое изменение объекта

Для динамического изменения объекта во время игры, необходимы переменные, с помощью которых будет регулироваться визуальный вид объекта.

**f = 42** – переменная высоты прямоугольника

**k = 14** – разница между значения координаты y и высотой

```
pygame.draw.rect(sc,(255,0,0), (346,14+k,22,f))
```





# Задание

---

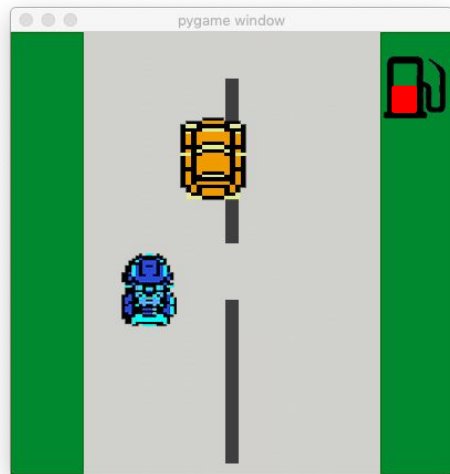
Добавьте в программу три глобальные переменные RED=(255, 0, 0), f=42 и d=14. И передачу этих значение в функцию rect.

```
RED=(255,0,0)
```

```
f=42
```

```
k=14
```

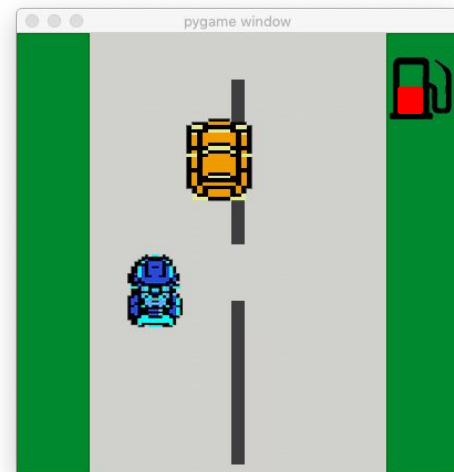
```
pygame.draw.rect(sc,RED, (346,14+k,22,f))
```



# Вопрос

---

Когда должен изменяться уровень топлива?

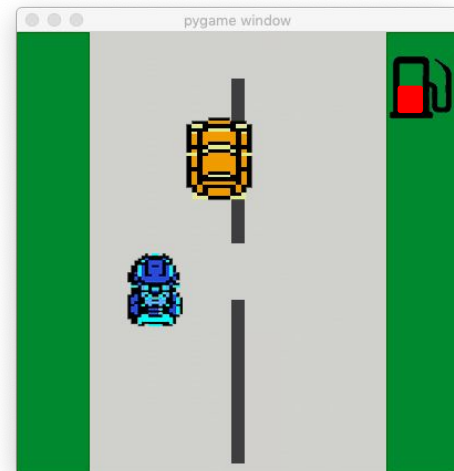


# Вопрос-Ответ

---

Когда должен изменяться уровень топлива?

**Во время всего игрового процесса.  
Пока `game_over = 1`**



# Задание

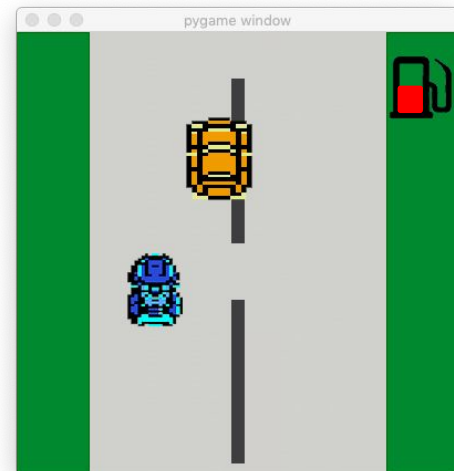
---

После проверки обновления игрового окна, добавьте изменение значений для переменных  $f$  и  $k$ .

```
pygame.display.update()
```

```
f=f-0.03
```

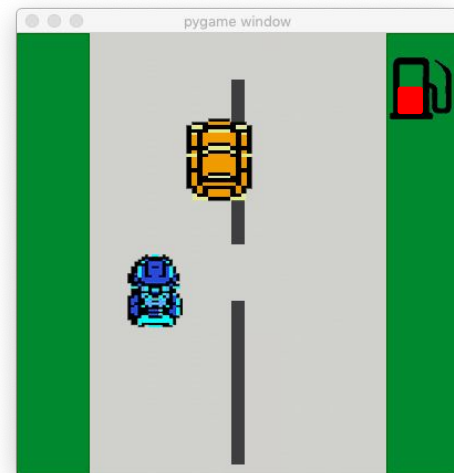
```
k=k+0.03
```



# Вопрос

---

Как должен измениться прямоугольник в бензобаке, когда гоночная машина коснется канистры?



# Вопрос-Ответ

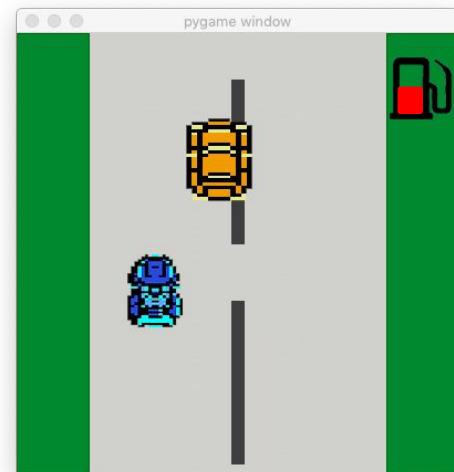
---

Как должен измениться прямоугольник в бензобаке, когда гоночная машина коснется канистры?

**Стать снова максимальным**

$$f=42$$

$$k=14$$

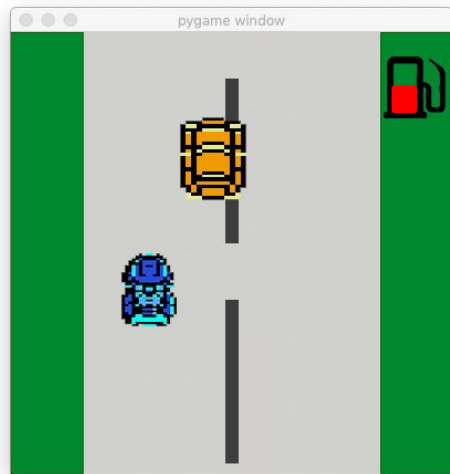


# Задание

---

Измените тело условного оператора проверки взаимодействия спрайта гоночной машины и канистры, чтобы после их взаимодействия уровень топлива стал максимальным.

$f=42$   
 $k=14$



# Задание. Решение

---

Измените тело условного оператора проверки взаимодействия спрайта гоночной машины и канистры, чтобы после их взаимодействия уровень топлива стал максимальным.

```
if pygame.sprite.spritecollideany(car1, canister_group):
```

```
    canister.rect.y=0-randint(200,1000)
```

```
    k=14
```

```
    f=42
```

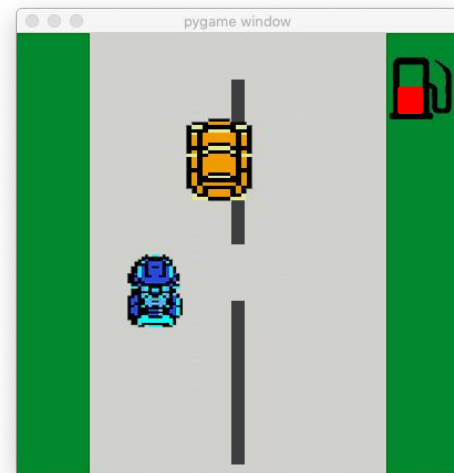




# Вопрос

---

В какой момент игры топливо снова должно стать максимальным?



# Вопрос-Ответ

---

В какой момент игры топливо снова должно стать максимальным?

**После перезапуска игры**

```
if i.type == pygame.QUIT:
```

```
    game=False
```

```
elif keys[pygame.K_9]:
```

```
    game_over=1
```

```
    car2.rect.y = 0
```



# Задание

---

Измените программу, чтобы после перезапуска уровень топлива снова становился максимальным.

```
if i.type == pygame.QUIT:
```

```
    game=False
```

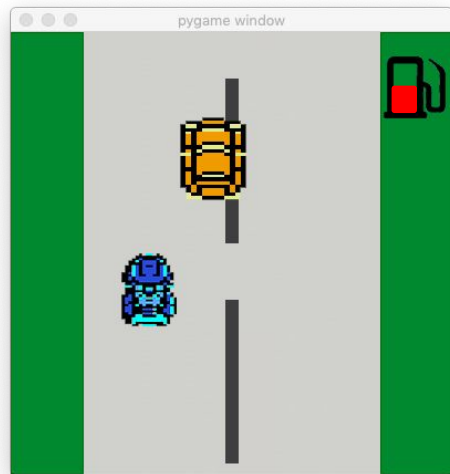
```
elif keys[pygame.K_9]:
```

```
    game_over=1
```

```
    car2.rect.y = 0
```

```
    k=14
```

```
    f=42
```



# Задание

---

Добавьте проверку значения переменной  $f$ . Если уровень топлива станет меньше 0, игра закончена.



# Задание. Решение

---

Добавьте проверку значения переменной  $f$ . Если уровень топлива станет меньше 0, игра закончена.

```
if f<0:  
    game_over=0
```



# Вопросы

---

1. Что такое условный оператор?
2. Как разместить спрайт в игровом окне?
3. Какая функция отвечает за создание графических примитивов в pygame?
4. Что такое динамический объект в программе?
5. Какие аргументы использует метод rect?

# Задание на дом. Уровень 1

---

Добавьте в программу взаимодействие спрайта канистры и машины противника. Если машина противника касается канистры, она пропадает, но уровень топлива не изменяется.

# Задание на дом. Уровень 2

---

Измените программу, если уровень топлива больше половины, прямоугольник - желтый, меньше половины - красный.

