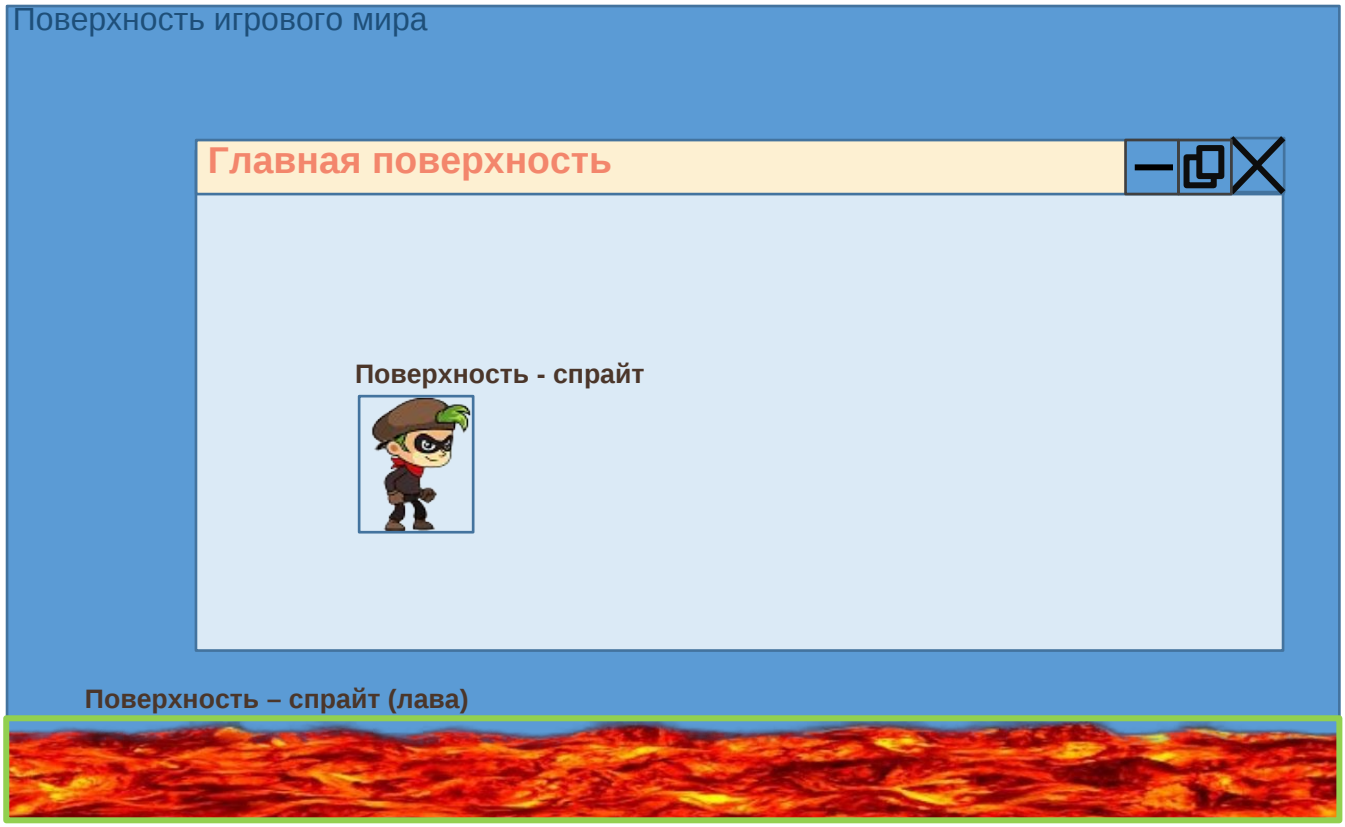
IMAGE. SURFACE. СОБЫТИЯ НАЖАТИЙ КЛАВИШ.

ЦЕЛЬ: **познакомиться с объектом Surface и методами модуля image, а также с событиями нажатий клавиш.**

ПЛАНИРОВАНИЕ

1. этап.

Сегодня мы познакомимся с таким модулем из библиотеки pygame, как **image**. Этот модуль содержит различные методы (функции) для работы с изображениями. Основным методом в этом модуле является метод load, аргументом которого является путь к файлу с изображением. Метод **load()** возвращает объект Surface (поверхность), которая представляет собой изображение из файла формата jpg или png. Вообще, в pygame основным объектом является прямоугольная поверхность (Surface), состоящая из пикселей, которые имеют разный цвет (в том числе и прозрачный) и представляющие собой какое либо изображение (спрайт). И по сути вся игра строиться из таких поверхностей, их можно представить как слои разного размера, которые рисуются поверх друг друга.



На прошлом уроке мы создали главную поверхность, а сейчас создадим поверхность с изображением спрайта игрока. Чтобы ее создать, необходимо вызвать метод load() у модуля image и в качестве аргумента передать ему путь к файлу с изображением. И он вернет вам поверхность с картинкой из файла.

Пример:

Player = pygame.image.load('Images/player.png')

Если файл изображения расположен не в той же папке, где ваш проект, то необходимо указать полный путь. Если в дочерней папке, то путь можно указать относительный.

В нашем примере изображение хранится в папке **Images,** которая находится в папке нашего проекта. В ней будем хранить все изображения для нашей игры.

Объект, представляющий собой поверхность, можно создать и с помощью встроенного класса Surface, и в последующих уроках мы им воспользуемся.

2. Этап

Теперь у нас создан объект поверхность, мы ему задали имя и можем с ним работать, давать ему команды, совершать с ним действия. По аналогии с переменными наш объект хранится в оперативной памяти как изображение и имеет имя, которое мы ему задали – Player.

Но если запустить программу, то будет черный экран – это потому, что нам нужно отобразить наш объект Player на объекте – поверхность, которую мы создали на прошлом уроке, мы ей задали имя – **w.** Это можно сделать с помощью метода (функции) объекта-поверхность – **w.** Этот метод называется: **blit** и имеет следующий формат использования:

w.blit (<Имя объекта Image>, (x, y)), где x,y – координаты точки, в которой мы хотим отобразить наш объект – Player. В точку (x, y) – будет установлен левый верхний угол объекта Player.

Итак, добавим в нашу программу команду **blit:**

Player = pygame.image.load('Images/Player.png')

w.blit(Player, (0, 0))

Давайте запустим программу…. Все равно экран черный, ничего не изменилось. Дело в том, что все, что мы делаем на объекте-поверхности **w,** происходит в оперативной памяти с этим объектом, как будто бы он виртуальный. И, чтобы вывести изображение этой поверхности со всеми нарисованными на нем объектами на экран монитора необходимо выполнить метод – **update()** модуля – **display:**

pygame.display.update()

Теперь соберем все вместе, должно получиться вот так:

```
import pygame

w = pygame.display.set_mode ((1279, 700))

Player = pygame.image.load('Images/Player.png')

game = True

while game:

    for ev in pygame.event.get ():

        if ev.type == pygame.QUIT:

            game = False

     w.blit(Player, (0, 0))

     pygame.display.update()

pygame.quit ()
```

После запуска мы увидим наше изображение на экране, в левом верхнем углу. Это значит, что начало координат в **pygame** находится в левом верхнем углу.

! Проблемная задача:

Предложите ученикам попробовать изменять координаты, чтобы изображение спрайта отображалось в других местах окна. Это необходимо, чтобы ученики ориентировались по координатам.

3. Этап

На прошлом занятии мы познакомились с событием – нажатие крестика закрытия окна программы: **QUIT**

Сегодня мы изучим еще несколько событий – события нажатий клавиш на клавиатуре.

В библиотеке **pygame** есть такой модуль – **key,** и у него есть метод – **get_pressed(),** который возвращает список с нажатыми в данный момент клавишами. И мы можем в нашем цикле игры, который мы создали на прошлом уроке, установить проверку: например, если нажата клавиша стрелка вправо, то изменить координату – **x,** в которой рисуется объект Image. Таким образом двигать его и, получается, им управлять.

Клавиша – стрелка вправо в pygame называется: pygame.K_RIGHT

Давайте создадим переменные x, y – они будут отвечать за координаты, в которых мы будем отображать наш объект – Player.

Также создадим список – keys и будем ему присваивать метод: pygame.key.get_pressed(), который возвращает список нажатых клавиш в данный момент.

keys = pygame.key.get_pressed()

И реализуем в игровом цикле проверку, если в списке нажатых клавиш присутствует клавиша K_RIGHT, то координату x увеличим на 5 px.

Вот как это будет выглядеть:

```
import pygame
w = pygame.display.set_mode ((1279, 700))
Player = pygame.image.load('Images/Player.png')
x = 100
y = 100
game = True
while game:
        for ev in pygame.event.get ():
                if ev.type == pygame.QUIT:
                        game = False
        keys = pygame.key.get_pressed()
        if keys[pygame.K_RIGHT]:
                x += 5
        w.blit(Player, (x, y))
        pygame.display.update()


pygame.quit()
```

В результате после запуска программы мы увидим, что при нажатии клавиши – стрелка вправо, наш объект будет резко двигаться вправо, оставляя за собой шлейф из своих копий.

! Проблемная задача. Задать вопрос ученикам, почему так происходит и как это исправить.

Ответ: происходит так потому, что на нашей поверхности все время рисуется изображение в новом месте, а в старом не стирается.

Чтобы от этого избавиться, необходимо в игровом цикле, каждый раз перед рисованием нового кадра игры, затирать старый. В классе Surface есть такой метод - **fill(color),** который закрашивает все пикселы экрана в заданный цвет – **color.** Цвет – color, передается в формате rgb.

w.fill((0, 0, 0)) – закрашиваем весь экран игры в черный цвет

! Проблемная задача. Исправить программу, так чтобы объект перемещался не оставляя за собой след.

! Проблемная задача. Добавить обработку событий нажатия клавиш: стрелка влево, вверх и вниз.

! Проблемная задача. Установить другой цвет заливки экрана.

4. Рефлексия

- Сегодня мы познакомились с модулями:

image

key

- Узнали, как получать список нажатых клавиш
- Запрограммировали движение персонажа