

Материалы к занятиям

```
import arcade

SCREEN_WIDTH = 1024
SCREEN_HEIGHT = 768
SCREEN_TITLE = "Lighting Demo"
MOVEMENT_SPEED = 5

class MyGame(arcade.Window):

    def __init__(self, width, height, title):
        super().__init__(width, height, title, resizable=True)

        self.background_sprite_list = None
        self.player_list = None
        self.wall_list = None
        self.player_sprite = None

        self.physics_engine = None

    def setup(self):
        self.background_sprite_list = arcade.SpriteList()
        self.player_list = arcade.SpriteList()
        self.wall_list = arcade.SpriteList()

        self.player_sprite =
arcade.Sprite(":resources:images/animated_characters/female_person/femalePerson_idle.png",
0.4)

        self.player_sprite.center_x = 64
        self.player_sprite.center_y = 270
        self.player_list.append(self.player_sprite)

        for x in range(-128, 2000, 128):
            for y in range(-128, 1000, 128):
                sprite =
arcade.Sprite(":resources:images/tiles/brickTextureWhite.png")
                sprite.position = x, y
                self.background_sprite_list.append(sprite)

        self.physics_engine =
arcade.PhysicsEngineSimple(self.player_sprite, self.wall_list)

    def on_draw(self):
        self.clear()
        self.background_sprite_list.draw()
        self.player_list.draw()

    def on_key_press(self, key, _):
        if key == arcade.key.UP:
            self.player_sprite.change_y = MOVEMENT_SPEED
```

```

        elif key == arcade.key.DOWN:
            self.player_sprite.change_y = -MOVEMENT_SPEED
        elif key == arcade.key.LEFT:
            self.player_sprite.change_x = -MOVEMENT_SPEED
        elif key == arcade.key.RIGHT:
            self.player_sprite.change_x = MOVEMENT_SPEED

    def on_key_release(self, key, _):
        if key == arcade.key.UP or key == arcade.key.DOWN:
            self.player_sprite.change_y = 0
        elif key == arcade.key.LEFT or key == arcade.key.RIGHT:
            self.player_sprite.change_x = 0

    def on_update(self, delta_time):
        self.physics_engine.update()

if __name__ == "__main__":
    window = MyGame(SCREEN_WIDTH, SCREEN_HEIGHT, SCREEN_TITLE)
    window.setup()
    arcade.run()

```

Теперь давайте добавим скролинг

```
VIEWPORT_MARGIN = 200
```

`__init__`

```

self.view_left = 0
self.view_bottom = 0

```

`setup`

```

self.view_left = 0
self.view_bottom = 0

```

```

def on_draw(self):
    self.clear()
    self.background_sprite_list.draw()
    self.player_list.draw()
    arcade.draw_text("Press SPACE to turn character light
on/off.",
                    10 + self.view_left, 10 + self.view_bottom,
                    arcade.color.WHITE, 20)

```

```
def on_update(self, delta_time):
    self.physics_engine.update()
    self.scroll_screen()

def on_resize(self, width, height):
    self.scroll_screen()
```

и сам метод для расчета скроллинга

```
def scroll_screen(self):

    left_boundary = self.view_left + VIEWPORT_MARGIN
    if self.player_sprite.left < left_boundary:
        self.view_left -= left_boundary - self.player_sprite.left

    right_boundary = self.view_left + self.width - VIEWPORT_MARGIN
    if self.player_sprite.right > right_boundary:
        self.view_left += self.player_sprite.right -
right_boundary

    top_boundary = self.view_bottom + self.height -
VIEWPORT_MARGIN
    if self.player_sprite.top > top_boundary:
        self.view_bottom += self.player_sprite.top - top_boundary

    bottom_boundary = self.view_bottom + VIEWPORT_MARGIN
    if self.player_sprite.bottom < bottom_boundary:
        self.view_bottom -= bottom_boundary -
self.player_sprite.bottom

    self.view_left = int(self.view_left)
    self.view_bottom = int(self.view_bottom)

    arcade.set_viewport(self.view_left,
                        self.width + self.view_left,
                        self.view_bottom,
                        self.height + self.view_bottom)
```

Теперь давайте выключим наш свет. А вернее отрисовку персонажа и фона

```
def on_draw(self):
    self.clear()
    arcade.draw_text("Press SPACE to turn character light
on/off.",
                    10 + self.view_left, 10 + self.view_bottom,
                    arcade.color.WHITE, 20)
```

Давайте сделаем тусклое освещение, а также дадим возможность пользователю включать фонарь по пробелу.

Создадим константу с цветом в формате RGB

```
AMBIENT_COLOR = (10, 10, 10)
```

В `__init__` создадим переменные для общего освещения и для освещения персонажа

```
self.light_layer = None  
self.player_light = None
```

В `setup` создадим экземпляр класса для света и установим фоновый цвет

```
self.light_layer = LightLayer(SCREEN_WIDTH, SCREEN_HEIGHT)  
self.light_layer.set_background_color(arcade.color.BLACK)
```

Изменим метод `on_draw` для отрисовки нашего освещения

```
def on_draw(self):  
    self.clear()  
    with self.light_layer:  
        self.background_sprite_list.draw()  
        self.player_list.draw()  
        self.light_layer.draw(ambient_color=AMBIENT_COLOR)  
  
    arcade.draw_text("Press SPACE to turn character light  
on/off.",  
                    10 + self.view_left, 10 + self.view_bottom,  
                    arcade.color.WHITE, 20)
```

Теперь наш персонаж бежит в потемках. В следующем занятии мы добавим фонарик, а также отдельные источники света

Давайте добавим нашему персонажу фонарик. Тем более почти все у нас уже готово. В `setup` добавим настройки света пользователя, такие как режим, радиус, цвет

```
radius = 150
```

```
mode = 'soft'
color = arcade.csscolor.WHITE
self.player_light = Light(0, 0, radius, color, mode)
```

Добавим в метод `on_key_press` нажатие на пробел включающее и выключающее свет

```
def on_key_press(self, key, _):

    if key == arcade.key.UP:
        self.player_sprite.change_y = MOVEMENT_SPEED
    elif key == arcade.key.DOWN:
        self.player_sprite.change_y = -MOVEMENT_SPEED
    elif key == arcade.key.LEFT:
        self.player_sprite.change_x = -MOVEMENT_SPEED
    elif key == arcade.key.RIGHT:
        self.player_sprite.change_x = MOVEMENT_SPEED
    elif key == arcade.key.SPACE:
        if self.player_light in self.light_layer:
            self.light_layer.remove(self.player_light)
        else:
            self.light_layer.add(self.player_light)
```

Запускаем и свет загорается в левом нижнем углу, так как мы установили координаты освещения в 0

```
self.player_light = Light(0, 0, radius, color, mode)
```

Давайте в `on_update` укажем, чтобы освещение перемещалось за нашим персонажем

```
self.player_light.position = self.player_sprite.position
```

Проверяем. Все работает отлично. Можете поэкспериментировать с настройками света для нашего игрока, а также с координатами освещения.

Давайте теперь раскидаем по уровню различные световые круги с разными настройками
Создадим маленький круг с мягким светом в координатах $x = 100$ и $y = 200$ в `setup`

```
x = 100
y = 200
radius = 100
mode = 'soft'
color = arcade.csscolor.WHITE
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Большой круг

```
x = 300
y = 150
radius = 200
color = arcade.csscolor.WHITE
mode = 'soft'
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Красный круг

```
x = 50
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.RED
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Синий круг

```
x = 450
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.BLUE
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Зеленый круг

```
x = 250
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.GREEN
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Теперь создадим три пересекающихся круга тех же цветов

```
x = 650
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.RED
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)

x = 750
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.GREEN
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)

x = 850
y = 450
radius = 100
mode = 'soft'
color = arcade.csscolor.BLUE
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

И добавим 3 пересекающихся круга с жестким светом

```
x = 650
y = 150
radius = 100
mode = 'hard'
color = arcade.csscolor.RED
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)

x = 750
y = 150
radius = 100
mode = 'hard'
color = arcade.csscolor.GREEN
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)

x = 850
y = 150
```

```
radius = 100
mode = 'hard'
color = arcade.csscolor.BLUE
light = Light(x, y, radius, color, mode)
self.light_layer.add(light)
```

Как мы видим работать со светом в arcade не так и сложно!