

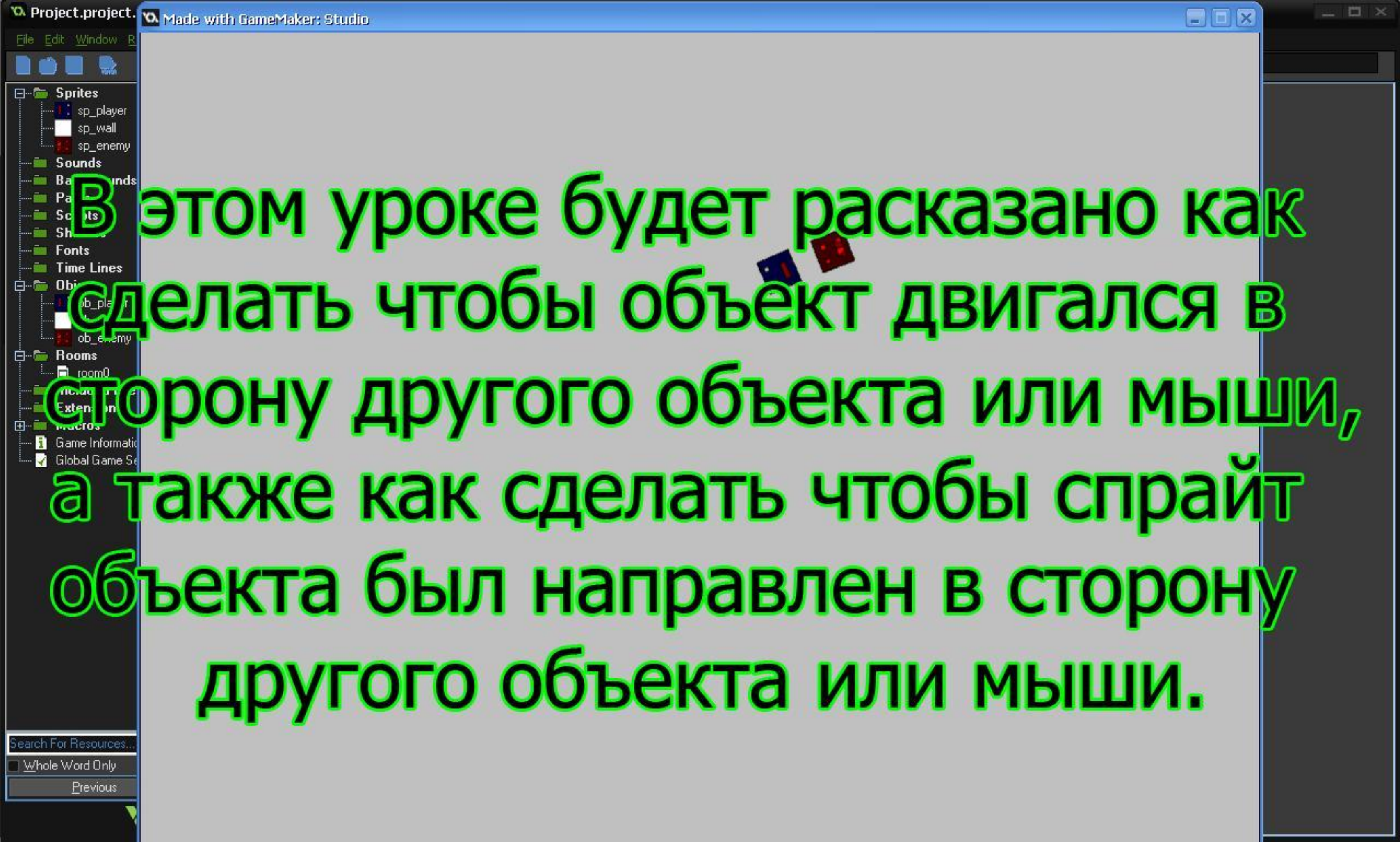
- Sprites
  - sp\_player
  - sp\_wall
  - sp\_enemy
- Sounds
- Backgrounds
- Paths
- Scripts
- Shaders
- Fonts
- Time Lines
- Objects
  - ob\_player
  - ob\_wall
  - ob\_enemy
- Rooms
  - room0
- Included Resources
- Extensions
- Macros
- Game Instance
- Global Settings

# Движение за объектом

SEMEN  
SEROV

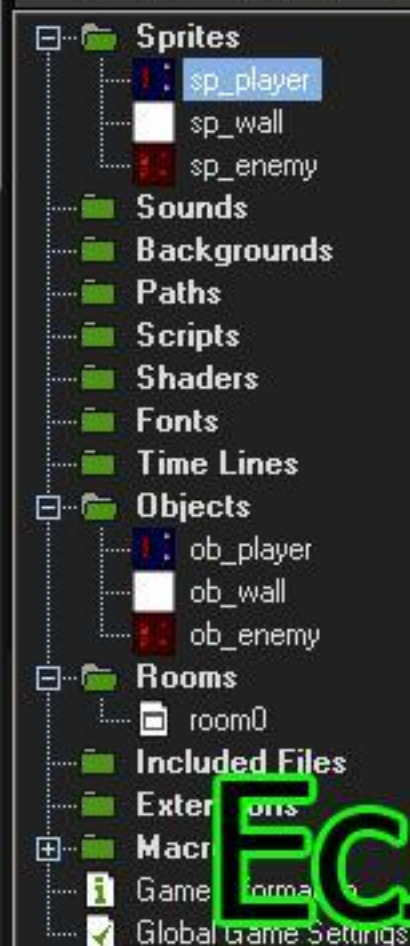
Search For Resources...  
 Whole Word Only  Filter Tree  
Previous Next





В этом уроке будет рассказано как сделать чтобы объект двигался в сторону другого объекта или мыши, а также как сделать чтобы спрайт объекта был направлен в сторону другого объекта или мыши.





### Sprite Properties: sp\_player

Name:

Width: 32 Height: 32  
Number of subimages: 1

Origin:

**Collision Checking**

- Precise collision checking
- Separate collision masks

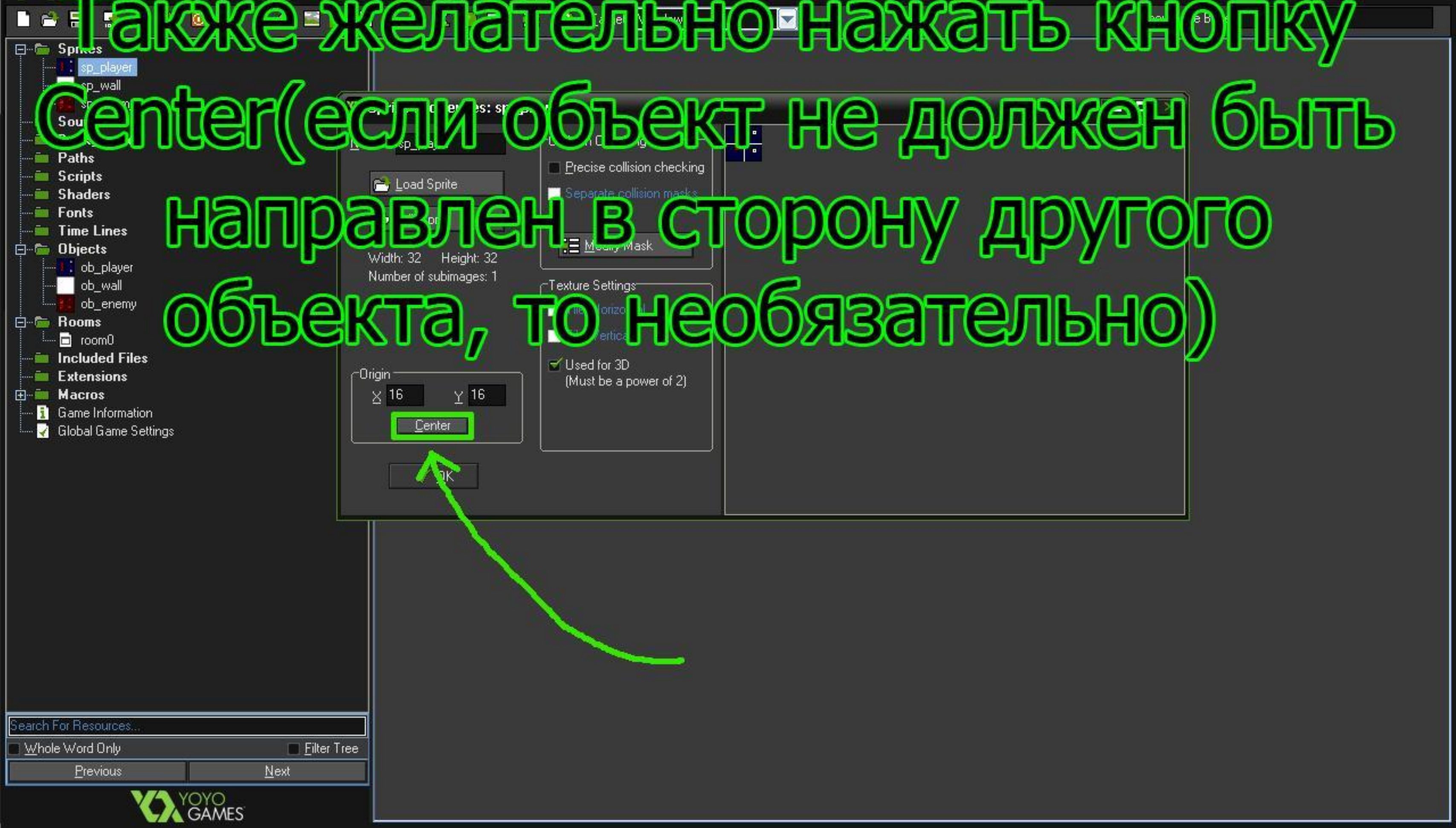
**Texture Settings**

- Tile: Horizontal
- Tile: Vertical
- Used for 3D

of 2)

Если вы хотите чтобы объект был направлен в сторону другого объекта или мыши, то спрайт должен быть направлен в правую сторону





Также желательно нажать кнопку Center (если объект не должен быть направлен в сторону другого объекта, то необязательно)

Load Sprite

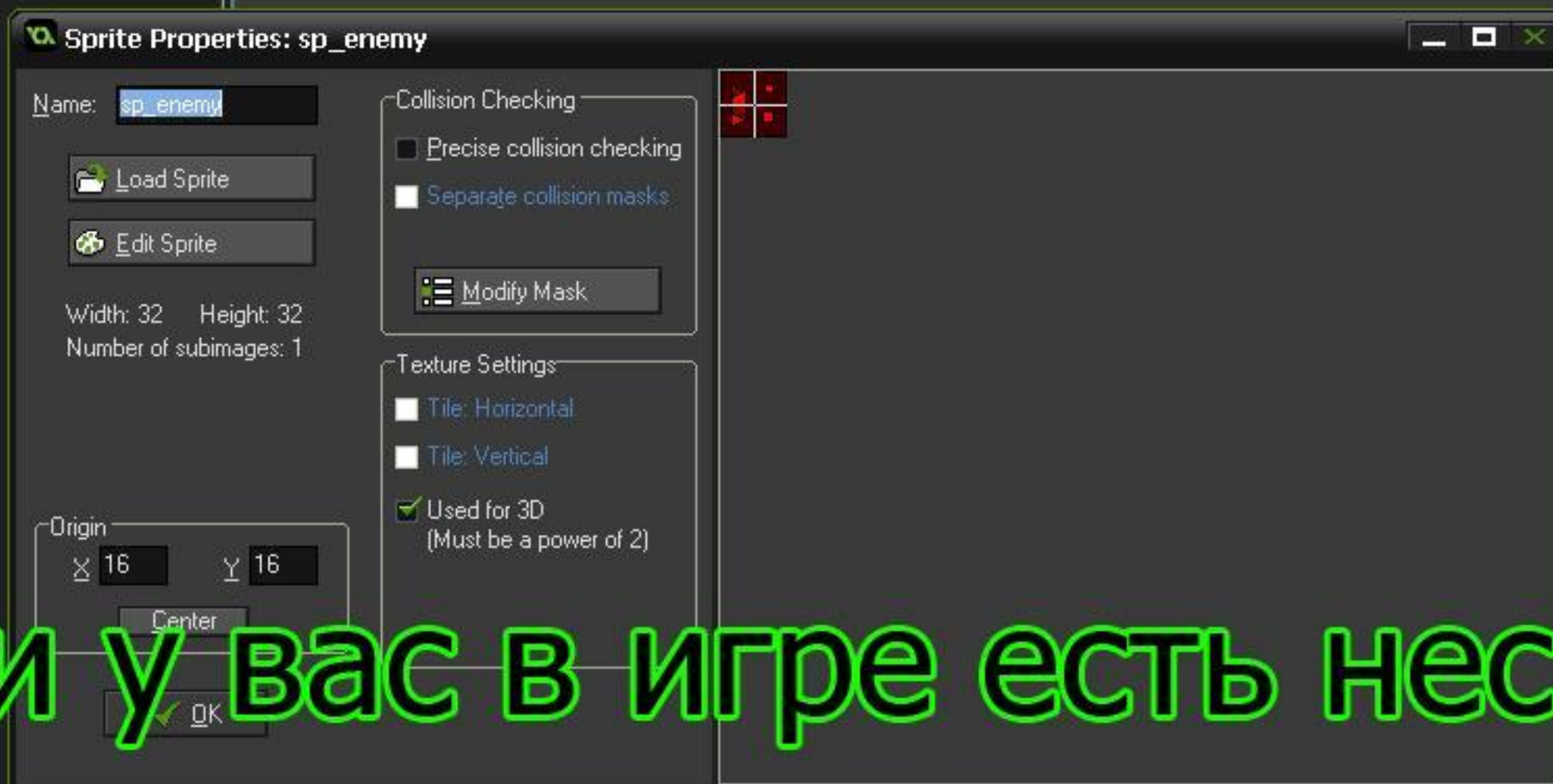
Width: 32 Height: 32  
Number of subimages: 1

Origin  
X: 16 Y: 16  
**Center**

Texture Settings  
 Used for 3D  
(Must be a power of 2)

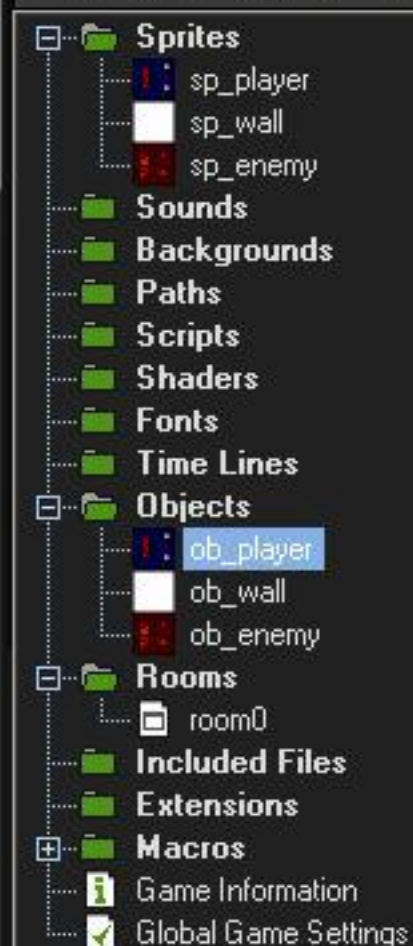
OK





Если у вас в игре есть несколько объектов «следящих» за другими объектами, то эти условия должны быть соблюдены и ими.





### Object Properties: ob\_player

Name: ob\_player

Sprite: sp\_player

Visible:  Solid:   
Persistent:  Uses Physics:

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:  
Step

Actions:  
Execute a piece of code

Move: [Move icons]  
Jump: [Jump icons]  
Paths: [Path icons]  
Steps: [Step icons]

Buttons: Add Event, Delete, Change, Show Information, OK

Запустим объект, который должен  
следить за мышкой



- Sprites
  - sp\_player
  - sp\_wall
  - sp\_enemy
- Sounds
- Backgrounds
- Paths
- Scripts
- Shaders
- Fonts
- Time Lines
- Objects
  - ob\_player
  - ob\_wall
  - ob\_enemy
- Rooms
  - room0
- Included Files
- Extensions
- Macros
- Game Information
- Global Game Settings

### Object Properties: ob\_player

Name: ob\_player

Sprite: sp\_player

Visible  Solid

Persistent  Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:

- Step

Actions:

- Execute a piece of code

Move: [Move icons]

Jump: [Jump icons]

Paths: [Path icons]

Steps: [Step icons]

Buttons: Add Event, Delete, Change, Show Information, OK

Создадим событие step



```
action  
1 direction=point_direction(x,y,mouse_x,mouse_y)  
2 image_angle=direction  
3
```

```
direction=point_direction(x,y,mouse_x,mouse_y)  
image_angle=direction
```

Чтобы спрайт объекта был  
направлен в сторону мышки:

```
direction=point_direction(x,y,mouse_x,mouse_y)  
image_angle=direction
```

Что значит:

направление=точке направления от x и y игрока,  
до x и y мышки;

угол изображения= направлению



```
action
1 if point_distance(x,y,mouse_x,mouse_y)>4
2 {
3 move_towards_point(mouse_x,mouse_y,4)
4 }
5 else {speed=0}
6
```

```
if point_distance(x,y,mouse_x,mouse_y)>4
{
move_towards_point(mouse_x,mouse_y,4)
}
else {speed=0}
```

# Чтобы объект следовал за мышкой:

```
if point_distance(x,y,mouse_x,mouse_y)>скорость
{
move_towards_point(mouse_x,mouse_y,скорость)
}
else {speed=0}
```



```
action
1 if point_distance(x,y,mouse_x,mouse_y)>4
2 {
3 move_towards_point(mouse_x,mouse_y,4)
4 }
5 else {speed=0}
6
```

```
if point_distance(x,y,mouse_x,mouse_y)>4
{
move_towards_point(mouse_x,mouse_y,4)
}
else {speed=0}
```

Что значит:

если дистанция между координатами  
объекта и мышки больше скорости, то  
объект двигаться к точке с координатами  
мышки с заданной скоростью,  
иначе скорость равна нулю



```
action
1 if point_distance(x,y,mouse_x,mouse_y)>4
2 {
3 move_towards_point(mouse_x,mouse_y,4)
4 direction=point_direction(x,y,mouse_x,mouse_y)
5 image_angle=direction
6 }
7 else {speed=0}
8
```

```
if point_distance(x,y,mouse_x,mouse_y)>4
{
move_towards_point(mouse_x,mouse_y,4)
direction=point_direction(x,y,mouse_x,mouse_y)
image_angle=direction
}
else {speed=0}
```

Чтобы объект одновременно следовал за мышкой и был направлен в её сторону, достаточно объединить оба скрипта в один



- Sprites
- Sounds
- Backgrounds
- Paths
- Scripts
- Shaders
- Fonts
- Time Lines
- Objects
  - ob\_player
  - ob\_wall
  - ob\_enemy
- Rooms
- Included Files
- Extensions
- Macros
- Game Information
- Global Game Settings

**Object Properties: ob\_enemy**

Name: ob\_enemy

Sprite: sp\_enemy

Visible  Solid

Persistent  Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Show Information

OK

Events:

- Step

Actions:

- Execute a piece of code

Move

Jump

Paths

Steps

move main1 main2 control score extra draw

Add Event

Delete Change

Запустим объект, который должен  
следить за другим объектом



- Sprites
- Sounds
- Backgrounds
- Paths
- Scripts
- Shaders
- Fonts
- Time Lines
- Objects
  - ob\_player
  - ob\_wall
  - ob\_enemy
- Rooms
- Included Files
- Extensions
- Macros
- Game Information
- Global Game Settings

### Object Properties: ob\_enemy

Name: ob\_enemy

Sprite: sp\_enemy

Visible:  Solid:   
Persistent:  Uses Physics:

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events: Step

Actions: Execute a piece of code

Move: [Move icons]

Jump: [Jump icons]

Paths: [Path icons]

Steps: [Step icons]

Buttons: Add Event, Delete, Change, Show Information, OK

Создадим событие step



```
action  
1 direction=point_direction(x,y,ob_player.x,ob_player.y)  
2 image_angle=direction  
3
```

```
direction=point_direction(x,y,ob_player.x,ob_player.y)  
image_angle=direction
```

Чтобы спрайт объекта был  
направлен в сторону другого объекта:

```
direction=point_direction(x,y,объект.x,объект.y)  
image_angle=direction
```

Что значит:

направление равно точке направления от текущего  
объекта, до указанного;  
угол спрайта равен направлению



```
action
1 if distance_to_object(ob_player)>40
2 {
3 move_towards_point(ob_player.x,ob_player.y,4)
4 }
5 else {speed=0}
6
```

```
if distance_to_object(ob_player)>40
{
move_towards_point(ob_player.x,ob_player.y,4)
}
else {speed=0}
```

# Чтобы объект следовал за другим объектом:

```
if distance_to_object(объект) > расстояние на которое
должен приближаться объект
{
move_towards_point(объект.x,объект.y,скорость)
}
else {speed=0}
```



```
action
1 if distance_to_object(ob_player)>40
2 {
3 move_towards_point(ob_player.x,ob_player.y,4)
4 }
5 else {speed=0}
6
```

```
if distance_to_object(ob_player)>40
{
move_towards_point(ob_player.x,ob_player.y,4)
}
else {speed=0}
```

## Что значит:

Если дистанция между объектами больше расстояния на которое должен приближаться объект, то объект движется к другому объекту с заданной скоростью; иначе скорость равна 0



```
action
1 if distance_to_object(ob_player)>40
2 {
3 move_towards_point(ob_player.x,ob_player.y,4)
4 direction=point_direction(x,y,ob_player.x,ob_player.y)
5 image_angle=direction
6 }
7 else {speed=0}
8
```

```
if distance_to_object(ob_player)>40
{
move_towards_point(ob_player.x,ob_player.y,4)
direction=point_direction(x,y,ob_player.x,ob_player.y)
image_angle=direction
}
else {speed=0}
```

Чтобы объект одновременно следовал за другим объектом и был направлен в его сторону, достаточно объединить оба скрипта в один