

Module Start

► EXPAND SOURCE CODE

Functions

```
def load_image(name, colorkey=None)
```

► EXPAND SOURCE CODE

Classes

```
class Clouds (cl, flag, *gr)
```

simple base class for visible game objects

pygame.sprite.Sprite(*groups): return Sprite

The base class for visible game objects. Derived classes will want to override the Sprite.update() method and assign Sprite.image and Sprite.rect attributes. The initializer can accept any number of Group instances that the Sprite will become a member of.

When subclassing the Sprite class, be sure to call the base initializer before adding the Sprite to Groups.

► EXPAND SOURCE CODE

Ancestors

pygame.sprite.Sprite

Class variables

```
var chert_image
```

```
var cloud_image
```

```
class Exit (*gr)
```

кнопка "выйти" при проигрыше

► EXPAND SOURCE CODE

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

Sprite.update(*args*, *kwargs):

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by Group.update() with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the Group class.

► [EXPAND SOURCE CODE](#)

```
class Finish (x, y, *gr)
```

дерево и табличка в конце уровней

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
class Heart (x, y, *gr)
```

сердца, или количество жизней, которые есть у Марио

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
class Info (*gr)
```

кнопка "информация" на главном экране(слева)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image1
```

```
var image2
```

```
var imn
```

Methods

```
def click(self, x, y)
```

нажатие

► EXPAND SOURCE CODE

```
def is_on(self, x, y)
```

реакция кнопки на наведение курсора на нее

► EXPAND SOURCE CODE

```
def update(self)
```

изменение при нажатии или наведении курсора на кнопку

► EXPAND SOURCE CODE

```
class Match (i, ln, *gr)
```

спрайты, которые пользователь видит, когда выбирает уровень сложности

► EXPAND SOURCE CODE

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def clear(self)
```

► EXPAND SOURCE CODE

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def del_dot(self)
```

► [EXPAND SOURCE CODE](#)

```
def get_name(self)
```

возвращает номер спрайта от 1 до 3(функция используется для расставления всех страйтов)

► [EXPAND SOURCE CODE](#)

```
def set_dot(self)
```

► [EXPAND SOURCE CODE](#)

```
def set_gr(self, gr)
```

добавление спрайта в группу

► [EXPAND SOURCE CODE](#)

```
class Next (*gr)
```

кнопка "продолжить играть" после победы

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

Sprite.update(*args*, *kwargs):

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by Group.update() with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the Group class.

► [EXPAND SOURCE CODE](#)

```
class Quit (*gr)
```

кнопка выхода из игры при победе

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

Sprite.update(*args*, *kwargs):

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by Group.update() with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the Group class.

► [EXPAND SOURCE CODE](#)

```
class Reload (*gr)
```

спрайт "играть заново" при проигрыше

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

Sprite.update(*args*, *kwargs):

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by Group.update() with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the Group class.

► [EXPAND SOURCE CODE](#)

```
class Settings (*gr)
```

кнопка, открывающая окно об авторах проекта

► [EXPAND SOURCE CODE](#)

Ancestors

pygame.sprite.Sprite

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

Sprite.update(*args*, *kwargs):

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by `Group.update()` with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the `Group` class.

► [EXPAND SOURCE CODE](#)

```
class Start (*gr)
```

кнопка "старт", т.е. начало игры

► [EXPAND SOURCE CODE](#)

Ancestors

`pygame.sprite.Sprite`

Class variables

```
var image
```

```
var image1
```

```
var imn
```

Methods

```
def click(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def is_on(self, x, y)
```

► [EXPAND SOURCE CODE](#)

```
def update(self)
```

method to control sprite behavior

`Sprite.update(args, *kwargs):`

The default implementation of this method does nothing; it's just a convenient "hook" that you can override. This method is called by `Group.update()` with whatever arguments you give it.

There is no need to use this method if not using the convenience method by the same name in the `Group` class.

► [EXPAND SOURCE CODE](#)

Index

Functions

[load_image](#)

Classes

Clouds

chert_image
cloud_image

Exit

click
image
image1
imn
is_on
update

Finish

image

Heart

image

Info

click
image1
image2
imn
is_on
update

Match

clear
click
del_dot
get_name
image
image1
imn
set_dot
set_gr

Next

click
image
image1
imn
is_on
update

Quit

click
image
image1
imn
is_on
update

Reload

click

image
image1
imn
is_on
update

Settings

click
image
image1
imn
is_on
update

Start

click
image
image1
imn
is_on
update