

# Adding images



Almost every game needs images. You can use images for your character (an "actor") or add them as background items.

## Creating an Actor

For your games, PyGame Zero provides the Actor class that can be initialized with a PNG image:

```
1 bee = Actor('bee')
```

PyGame Zero will look for the `bee.png` (beware of the upper and lower cases!) in the `images` directory next to your Python program. To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:



To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    bee.draw()
```

# Adding images



Almost every game needs images. You can use images for your character (an "actor") or add them as background items.

## Creating an Actor

For your games, PyGame Zero provides the Actor class that can be initialized with a PNG image:

```
1 bee = Actor('bee')
```

PyGame Zero will look for the `bee.png` (beware of the upper and lower cases!) in the `images` directory next to your Python program. To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:



To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    bee.draw()
```

# Adding images



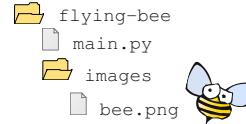
Almost every game needs images. You can use images for your character (an "actor") or add them as background items.

## Creating an Actor

For your games, PyGame Zero provides the Actor class that can be initialized with a PNG image:

```
1 bee = Actor('bee')
```

PyGame Zero will look for the `bee.png` (beware of the upper and lower cases!) in the `images` directory next to your Python program. To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:



To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    bee.draw()
```

# Adding images



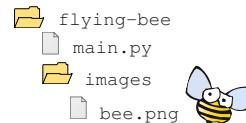
Almost every game needs images. You can use images for your character (an "actor") or add them as background items.

## Creating an Actor

For your games, PyGame Zero provides the Actor class that can be initialized with a PNG image:

```
1 bee = Actor('bee')
```

PyGame Zero will look for the `bee.png` (beware of the upper and lower cases!) in the `images` directory next to your Python program. To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:



To get PyGame Zero to show the bee, you call its `draw()` function in the PyGame's `draw()` function:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    bee.draw()
```

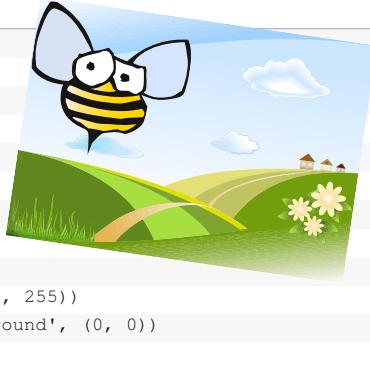
If you click on the "Play" button, you should see a bee in the top left corner of a light blue window.

If the image is too big or too small, you can resize it an image editor (like Gimp).

## Adding background images

While actors contain some handy features for animating, moving and checking for collisions, sometimes you might simply want to add a "simple" image to your game: you can use the `blit` ("bit block transfer") function for it:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    screen.blit('background', (0, 0))
12    bee.draw()
```



You need a file called `background.png` in your `images` directory. If you want the background to fill the window you will need to resize it to the window size: in the example above it's 640 x 480 pixels.

If you click on the "Play" button, you should see a bee in the top left corner of a light blue window.

If the image is too big or too small, you can resize it an image editor (like Gimp).

## Adding background images

While actors contain some handy features for animating, moving and checking for collisions, sometimes you might simply want to add a "simple" image to your game: you can use the `blit` ("bit block transfer") function for it:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    screen.blit('background', (0, 0))
12    bee.draw()
```



You need a file called `background.png` in your `images` directory. If you want the background to fill the window you will need to resize it to the window size: in the example above it's 640 x 480 pixels.

If you click on the "Play" button, you should see a bee in the top left corner of a light blue window.

If the image is too big or too small, you can resize it an image editor (like Gimp).

## Adding background images

While actors contain some handy features for animating, moving and checking for collisions, sometimes you might simply want to add a "simple" image to your game: you can use the `blit` ("bit block transfer") function for it:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    screen.blit('background', (0, 0))
12    bee.draw()
```



You need a file called `background.png` in your `images` directory. If you want the background to fill the window you will need to resize it to the window size: in the example above it's 640 x 480 pixels.

If you click on the "Play" button, you should see a bee in the top left corner of a light blue window.

If the image is too big or too small, you can resize it an image editor (like Gimp).

## Adding background images

While actors contain some handy features for animating, moving and checking for collisions, sometimes you might simply want to add a "simple" image to your game: you can use the `blit` ("bit block transfer") function for it:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 def update(dt):
7     pass
8
9 def draw():
10    screen.fill((0, 210, 255))
11    screen.blit('background', (0, 0))
12    bee.draw()
```



You need a file called `background.png` in your `images` directory. If you want the background to fill the window you will need to resize it to the window size: in the example above it's 640 x 480 pixels.