

```
1  import CU_media_0_15 as CU_media
2
3  def ave_brightness(pixel):
4      """Calculate the brightness of the pixel."""
5      red = CU_media.get_red(pixel)
6      green = CU_media.get_green(pixel)
7      blue = CU_media.get_blue(pixel)
8      return (red+green+blue)/3
9
10 class ImageFilter3D(object):
11     """Objects of this class can be used to create anaglyph 3D images."""
12     def __init__(self):
13         """Construct an empty ImageFilter3D object."""
14         pass
15
16     def load_left_image(self):
17         """Loads the left image in as an instance variable."""
18         self.left_image = CU_media.load_picture(CU_media.choose_file())
19         self.left_height = CU_media.get_height(self.left_image)
20         self.left_width = CU_media.get_width(self.left_image)
21
22     def load_right_image(self):
23         """Loads the right image in as an instance variable."""
24         self.right_image = CU_media.load_picture(CU_media.choose_file())
25         self.right_height = CU_media.get_height(self.right_image)
26         self.right_width = CU_media.get_width(self.right_image)
27
28     def create_anaglyph_image(self):
29         """Ensure left and right images have been loaded. Before calling this
30         method. The left image brightness per pixel is saved to the red channel.
31         The right image brightness per pixel is saved to the blue channel."""
32         #ensure the left and right images have the same dimensions!
33         self.image_3D = CU_media.make_empty_picture(\
34             self.left_width,\
35             self.left_height)
36
37         for x_coord in range(self.left_width):
38             for y_coord in range(self.left_height):
39                 result_pixel = CU_media.get_pixel(self.image_3D,\
40                                                    x_coord,y_coord)
41
42                 red_val = ave_brightness(\
43                     CU_media.get_pixel(self.left_image,\
44                                         x_coord,\
45                                         y_coord))
46
47                 blue_val = ave_brightness(\
48                     CU_media.get_pixel(self.right_image,\
49                                         x_coord,\
50                                         y_coord))
51
52                 CU_media.set_red(result_pixel,red_val)
```

```
52         CU_media.set_blue(result_pixel,blue_val)
53
54     def show_3D_image(self):
55         CU_media.show(self.image_3D)
```