

```

|0 |1 |2 |3 |4 |5 |6 |7 |8
1  #!/usr/bin/python
2  # Lair of Doom
3  # Code Angel
4
5  import sys
6  import os
7  import pygame
8  from pygame.locals import *
9
10 import level
11 import player
12 import screen
13
14 # Setup
15 os.environ['SDL_VIDEO_CENTERED'] = '1'
16 pygame.mixer.pre_init(44100, -16, 2, 512)
17 pygame.mixer.init()
18 pygame.init()
19 game_screen = pygame.display.set_mode((screen.SCREEN_WIDTH, screen.SCREEN_HEIGHT))
20 pygame.display.set_caption('The Lair of Doom')
21 pygame.key.set_repeat(10, 20)
22 clock = pygame.time.Clock()
23 medium_font = pygame.font.SysFont('Helvetica Bold', 24)
24 large_font = pygame.font.SysFont('Helvetica Bold', 36)
25
26
27 def main():
28
29     # Load background image
30     background_image = load_media('image', 'background')
31
32     # Initialise objects
33     # Create a Player object. Let's call him Alfie
34     alfie = player.Player()
35
36     # Create a Level object and set up the level
37     game_level = level.Level()
38     game_level.set_up()
39
40     # Set alfie's start location
41     alfie.set_location(game_level.player_start_loc)
|0 |1 |2 |3 |4 |5 |6 |7 |8

```

```

42 |0 |1 |2 |3 |4 |5 |6 |7 |8
43 # Main game loop
44 while True:
45     for event in pygame.event.get():
46         key_pressed = pygame.key.get_pressed()
47
48         # Space pressed and player is not already jumping
49         if key_pressed[pygame.K_SPACE] and alfie.jumping is False:
50             alfie.start_jump()
51
52         # Left key pressed
53         elif key_pressed[pygame.K_LEFT]:
54             if alfie.jumping is False:
55                 alfie.move(-player.MOVEMENT_SPACE, 0)
56             else:
57                 alfie.jumping_left = True
58
59         # Right key pressed
60         elif key_pressed[pygame.K_RIGHT]:
61             if alfie.jumping is False:
62                 alfie.move(player.MOVEMENT_SPACE, 0)
63             else:
64                 alfie.jumping_right = True
65
66         # Game has ended and player has pressed return
67         elif key_pressed[pygame.K_RETURN]:
68             if alfie.game_over is True or alfie.game_completed is True:
69                 game_level = level.Level()
70                 game_level.set_up()
71                 alfie = player.Player()
72                 alfie.set_location(game_level.player_start_loc)
73                 alfie.diamonds_collected = 0
74
75             if event.type == QUIT:
76                 pygame.quit()
77                 sys.exit()
78
79         # Update player location if jumping / falling
80         if alfie.jumping is True:
81             alfie.jump_move()
82

```

```

83      |0  |1  |2  |3  |4  |5  |6  |7  |8
84      # Move doom monsters
85      for monster in level.doom_monsters:
86          monster.move()
87
88      # Check if player has collided with anything
89      alfie.check_doom_monsters()
90      alfie.check_water()
91      alfie.check_diamonds()
92      alfie.check_exit(game_level)
93
94      # Draw background
95      game_screen.blit(background_image, [0, 0])
96
97      # Draw ledges
98      for ledge in level.ledges:
99          for block_count in range(ledge.block_length):
100              game_screen.blit(ledge.image, [ledge.rect.x + block_count * level.BLOCK_SIZE, ledge.rect.y])
101
102      # Draw water
103      for wave in level.water:
104          for block_count in range(wave.block_length):
105              game_screen.blit(wave.image, [wave.rect.x + block_count * level.BLOCK_SIZE, wave.rect.y])
106
107      # Draw diamonds
108      for diamond in level.diamonds:
109          game_screen.blit(diamond.image, [diamond.rect.x, diamond.rect.y])
110
111      # Draw doom monsters
112      for monster in level.doom_monsters:
113          game_screen.blit(monster.image, [monster.rect.x, monster.rect.y])
114
115      # Draw exits
116      for door in level.exit_doors:
117          game_screen.blit(door.image, [door.rect.x, door.rect.y])
118
119      # Draw player
120      if alfie.game_over is False:
121          game_screen.blit(alfie.image, [alfie.rect.x, alfie.rect.y])
122
123      screen.display_scoreboard(game_screen, medium_font, alfie.diamonds_collected)

```

```

124 |0 |1 |2 |3 |4 |5 |6 |7 |8
125     # Display Game Over
126     if alfie.game_over is True:
127         screen.display_game_over(game_screen, medium_font, large_font, 'Game Over')
128     elif alfie.game_completed is True:
129         screen.display_game_over(game_screen, medium_font, large_font, 'Game Completed')
130
131     pygame.display.update()
132     clock.tick(60)
133
134     # Get an image or audio from folder
135     def load_media(media_type, filename):
136         media = None
137         full_path = os.path.dirname(os.path.realpath(__file__))
138
139         if media_type == 'image':
140             images_path = os.path.join(full_path, 'images')
141             full_filename = os.path.join(images_path, filename + '.png')
142             media = pygame.image.load(full_filename).convert_alpha()
143         elif media_type == 'audio':
144             audio_path = os.path.join(full_path, 'audio')
145             full_filename = os.path.join(audio_path, filename + '.ogg')
146             media = pygame.mixer.Sound(full_filename)
147
148         return media
149
150
151     if __name__ == '__main__':
152         main()
153

```

```

|0 |1 |2 |3 |4 |5 |6 |7 |8

```