```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
1   #!/usr/bin/python
2   # Toadie
3   # Code Angel
4
5   import sys
6   import os
7   import pygame
8   from pygame.locals import *
9
10  import toad
11  import world
12  import scoreboard
13
14  # Define the colours
15  BLACK = (0, 0, 0)
16  DARK_BLUE = (0, 0, 121)
17
18  # Define constants
19  SCREEN_WIDTH = 448
20  SCREEN_HEIGHT = 512
21
22  # Setup
23  os.environ['SDL_VIDEO_CENTERED'] = '1'
24  pygame.mixer.pre_init(44100, -16, 2, 512)
25  pygame.mixer.init()
26  pygame.init()
27  game_screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
28  pygame.display.set_caption('Toadie')
29  pygame.key.set_repeat(500, 200)
30
31  clock = pygame.time.Clock()
32
33  large_font = pygame.font.SysFont('Helvetica', 24)
34  score_font = pygame.font.SysFont('Helvetica Bold', 24)
35
36
37  def main():
38
39      # Load images
40      toad_lives_image = load_media('image', 'toad_lives')
41      home_image = load_media('image', 'home')
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
42
43      # initialise variables
44      screen_blocks_wide = int(SCREEN_WIDTH / world.BLOCK_SIZE)
45
46      # Initialise objects
47      toadie = toad.Toad()
48      score = 0
49      hi_score = 0
50      game_timer = world.Timer()
51
52      # Create a list of pavement blocks the width of the screen
53      pavement_blocks = []
54      for counter in range(screen_blocks_wide):
55          pavement_block = world.Pavement([counter, world.PAVEMENT_LANE_1])
56          pavement_blocks.append(pavement_block)
57          pavement_block = world.Pavement([counter, world.PAVEMENT_LANE_2])
58          pavement_blocks.append(pavement_block)
59
60      # Create a list of landing pads
61      landing_pads = []
62      for counter in range(5):
63          landing_pad = world.Pad(counter * 3)
64          landing_pads.append(landing_pad)
65
66      # List of cars, trucks and diggers
67      traffic = []
68
69      # List of logs and turtles
70      river = []
71
72      # Create 3 red cars
73      red_car_1 = world.RedCar(4)
74      red_car_2 = world.RedCar(9)
75      red_car_3 = world.RedCar(14)
76      traffic.extend((red_car_1, red_car_2, red_car_3))
77
78      # Create 3 diggers
79      digger_1 = world.Digger(6)
80      digger_2 = world.Digger(11)
81      digger_3 = world.Digger(16)
82      traffic.extend((digger_1, digger_2, digger_3))
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
83
84      # Create 3 purple cars
85      purple_car_1 = world.PurpleCar(2)
86      purple_car_2 = world.PurpleCar(7,)
87      purple_car_3 = world.PurpleCar(12)
88      traffic.extend((purple_car_1, purple_car_2, purple_car_3))
89
90      # Create 3 pink cars
91      pink_car_1 = world.PinkCar(3)
92      pink_car_2 = world.PinkCar(8)
93      pink_car_3 = world.PinkCar(13)
94      traffic.extend((pink_car_1, pink_car_2, pink_car_3))
95
96      # Create 2 trucks
97      truck_1 = world.Truck(3)
98      truck_2 = world.Truck(9)
99      traffic.extend((truck_1, truck_2))
100
101     # Create 4 turtle chains (3 turtles per chain)
102     turtle_a1 = world.Turtle(1, 3)
103     turtle_a2 = world.Turtle(5, 3)
104     turtle_a3 = world.Turtle(9, 3)
105     turtle_a4 = world.Turtle(13, 3)
106     river.extend((turtle_a1, turtle_a2, turtle_a3, turtle_a4))
107
108     # Create 4 turtle chains (2 turtles per chain
109     turtle_b1 = world.Turtle(2, 2)
110     turtle_b2 = world.Turtle(6, 2)
111     turtle_b3 = world.Turtle(10, 2)
112     turtle_b4 = world.Turtle(14, 2)
113     river.extend((turtle_b1, turtle_b2, turtle_b3, turtle_b4))
114
115     # Create 3 of the smallest logs
116     log_a1 = world.Log(1, 1)
117     log_a2 = world.Log(6, 1)
118     log_a3 = world.Log(11, 1)
119     river.extend((log_a1, log_a2, log_a3))
120
121     # Create 2 of the middle sized logs
122     log_b1 = world.Log(5, 3)
123     log_b2 = world.Log(13, 3)
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
    river.extend((log_b1, log_b2))

    # Create 3 of the longest logs
    log_c1 = world.Log(3, 2)
    log_c2 = world.Log(9, 2)
    log_c3 = world.Log(15, 2)
    river.extend((log_c1, log_c2, log_c3))

    # Main game loop
    while True:
        for event in pygame.event.get():
            key_pressed = pygame.key.get_pressed()

            if key_pressed[pygame.K_LEFT]:
                toadie.move('L')
            elif key_pressed[pygame.K_RIGHT]:
                toadie.move('R')
            elif key_pressed[pygame.K_UP]:
                toadie.move('U')
            elif key_pressed[pygame.K_DOWN]:
                toadie.move('D')

            # RETURN key pressed when lives are 0, or all 5 frogs are home so start a new game
            elif key_pressed[pygame.K_RETURN] and (toadie.lives == 0 or toadie.home_count == 5):
                toadie = toad.Toad()
                score = 0
                game_timer = world.Timer()

                if new_hi_score > hi_score:
                    hi_score = new_hi_score

                del landing_pads[:]
                for counter in range(5):
                    landing_pad = world.Pad(counter * 3)
                    landing_pads.append(landing_pad)

            if event.type == QUIT:
                pygame.quit()
                sys.exit()

        # Draw the water
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
165         game_screen.fill(DARK_BLUE)
166
167         # Draw a black rectangle for the road and lower part of the screen
168         pygame.draw.rect(game_screen, BLACK, (0, SCREEN_HEIGHT / 2, SCREEN_WIDTH, SCREEN_HEIGHT))
169
170         # Display the home bases at the top of the screen
171         game_screen.blit(home_image, [0, 0])
172
173         # Draw pavement blocks
174         for pavement in pavement_blocks:
175             pavement.draw(game_screen)
176
177         # Draw landing pads
178         for pad in landing_pads:
179             pad.draw(game_screen)
180
181         # Move and display the river objects (turtles, logs)
182         for moving_river_object in river:
183             moving_river_object.move(game_screen)
184
185         # Move and display the road objects (cars, diggers, trucks)
186         for moving_road_object in traffic:
187             moving_road_object.move(game_screen)
188
189         # Draw toadie
190         toadie.draw(game_screen)
191
192         # Check if toadie has collided with traffic or road object
193         toadie.check_collision(traffic)
194         toadie.check_water(river)
195
196         # Check if toadie is home
197         toadie.check_home(landing_pads, game_timer)
198
199         # Update the timer and display the scores
200         game_timer.update_time()
201         scoreboard.display_scores(game_screen, toadie.lives, score, game_timer.time_remaining, toad_lives_image)
202
203         # Check if timer has run down - if so toadie will die
204         if game_timer.out_of_time() is True:
205             toadie.die()
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
206
207         # If toadie is dead, the toad skeleton is displayed for a short time
208         if toadie.alive is False and toadie.lives > 0:
209             toadie.check_death_pause(game_timer)
210
211         # If lives are 0 or all 5 toads are home then it is game over
212         if toadie.lives == 0 or toadie.home_count == 5:
213             new_hi_score = check_hi_score(score, hi_score)
214             scoreboard.game_over(game_screen, score, hi_score, new_hi_score)
215
216         # Update the score
217         score = score + toadie.collect_points()
218
219         pygame.display.update()
220         clock.tick(30)
221
222
223 # Check if the new high score is greater than the current high score
224 def check_hi_score(score, hi_score):
225     new_hi_score = 0
226     if score > hi_score:
227         new_hi_score = score
228
229     return new_hi_score
230
231
232 # Get an image or audio from folder
233 def load_media(media_type, filename):
234     media = None
235     full_path = os.path.dirname(os.path.realpath(__file__))
236
237     if media_type == 'image':
238         images_path = os.path.join(full_path, 'images')
239         full_filename = os.path.join(images_path, filename + '.png')
240         media = pygame.image.load(full_filename).convert_alpha()
241     elif media_type == 'audio':
242         audio_path = os.path.join(full_path, 'audio')
243         full_filename = os.path.join(audio_path, filename + '.ogg')
244         media = pygame.mixer.Sound(full_filename)
245
246     return media
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```

```
    |0  |1  |2  |3  |4  |5  |6  |7  |8
247
248
249 if __name__ == '__main__':
250     main()
251
    |0  |1  |2  |3  |4  |5  |6  |7  |8
```