```
|0 |1 |2 |3 |4 |5 |6 |7 |8
1 #!/usr/bin/python
   # Alien Invasion
   # Code Angel
4
5
    import sys
    import os
    import pygame
    from pygame.locals import *
9
    import random
10
11
    # Define the colours
    LIGHT YELLOW = (255, 255, 204)
12
13
    WHITE = (255, 255, 255)
14
15
    # Define constants
16
    SCREEN WIDTH = 640
17
    SCREEN HEIGHT = 480
    SCOREBOARD MARGIN = 4
18
19
20
    MISSILE PLATFORM = 31
21
    MISSILE SPEED = 10
22
    GAME MISSILES = 20
23
24
    UFO UPPER Y = 20
25
    UFO LOWER Y = 240
    UFO HIT TIME = 20
27
    UFO OFF TIME = 60
28
    UFO SCORE = 50
29
30
    RANDOM VERTICAL CHANGE = 20
31
    RANDOM HORIZONTAL CHANGE = 100
    UFO DIRECTIONS = ['left', 'right', 'up', 'down']
32
33
34
    RANDOM RAY = 200
35
    RANDOM RAY TIME MAX = 120
36
    RANDOM RAY TIME MIN = 30
37
38
    BASE SPEED = 6
39
40
    # Setup
41
    os.environ['SDL VIDEO CENTERED'] = '1'
    |0 |1 |2 |3 |4 |5 |6 |7 |8
```

Book of Code 3 – Alien Invasion 1 © 2017 Code Angel

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
    pygame.mixer.pre init(44100, -16, 2, 512)
    pygame.mixer.init()
    pygame.init()
44
    game screen = pygame.display.set mode((SCREEN WIDTH, SCREEN HEIGHT))
45
46
    pygame.display.set caption('Alien Invasion')
47
    pygame.key.set repeat(10, 20)
48
    clock = pygame.time.Clock()
    font = pygame.font.SysFont('Helvetica', 16)
49
50
51
    # Load images
52
    background image = pygame.image.load('background.png').convert()
53
    base image = pygame.image.load('base.png').convert alpha()
54
    missile image = pygame.image.load('missile.png').convert alpha()
55
    missile fired image = pygame.image.load('missile fired.png').convert alpha()
56
57
    ufo 1 image = pygame.image.load('ufo 1.png').convert alpha()
    ufo 2 image = pygame.image.load('ufo 2.png').convert alpha()
59
    ufo 1 exploded image = pygame.image.load('ufo 1 exploded.png').convert alpha()
60
    ufo 2 exploded image = pygame.image.load('ufo 2 exploded.png').convert alpha()
    ufo ray image 1 = pygame.image.load('ufo ray 1.png').convert alpha()
61
62
    ufo ray image 2 = pygame.image.load('ufo ray 2.png').convert alpha()
63
64
    # Load sounds
65
    spaceship hit sound = pygame.mixer.Sound('spaceship hit.ogg')
    launch sound = pygame.mixer.Sound('launch.ogg')
67
68
69
    def main():
70
71
         # Initialise variables
72
        base x = 300
73
        base y = 430
74
        base width = base image.get rect().width
75
76
        ufo width = ufo 1 image.get rect().width
77
        ufo height = ufo 1 image.get rect().height
78
79
         ray width = ufo ray image 1.get rect().width
80
81
        ufo 1 x = SCREEN WIDTH - ufo width
82
        ufo 1 y = random.randint(UFO UPPER Y, UFO LOWER Y)
     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
 83
 84
          # UFO 1 dicitionary
 85
          ufo 1 = {'x loc': ufo 1 x, 'y loc': ufo 1 y, 'direction': 'left', 'hit': False, 'hit time': 0, 'off time': 0,
 86
                   'ray time': 0, 'speed': 5}
 87
 88
          ufo 2 y = random.randint(UFO UPPER_Y, UFO_LOWER_Y)
 89
 90
          # UFO 2 dictionary
 91
          ufo 2 = {'x loc': 0, 'y loc': ufo 2 y, 'direction': 'right', 'hit': False, 'hit time': 0, 'off time': 0,
 92
                   'ray time': 0, 'speed': 3}
 93
 94
          missile x = 0
 95
          missile y = 0
 96
          missile firing = False
 97
 98
          missile width = missile image.get rect().width
 99
          missile height = missile image.get rect().height
100
101
          score = 0
102
         hi score = 0
103
          missiles = GAME MISSILES
104
          game over = False
105
106
          # Main game loop
107
          while True:
108
109
              for event in pygame.event.get():
110
                  key pressed = pygame.key.get pressed()
111
112
                  # Left key pressed, move base left
113
                  if key pressed[pygame.K LEFT]:
114
                     base x -= BASE SPEED
115
                      if base x < 0:
116
                          base x = 0
117
118
                  # Right key pressed, move base right
119
                  elif key pressed[pygame.K RIGHT]:
120
                     base x += BASE SPEED
121
                      if base x > SCREEN WIDTH - base width:
122
                          base x = SCREEN WIDTH - base width
123
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
124
                  # Space pressed, fire missile
125
                  elif key pressed[pygame.K SPACE] and missile firing is False and game over is False:
                      missile firing = True
126
127
                      missile x = base x + MISSILE PLATFORM
128
                      missile y = base y - missile height
129
                      missiles -= 1
130
                      launch sound.play()
131
                      if missiles == 0:
132
                          game over = True
133
134
                  # Return pressed at end of game, start new game
135
                  elif key pressed[pygame.K_RETURN] and game_over is True:
136
                      game over = False
137
                      score = 0
138
                      missiles = GAME MISSILES
139
140
                  # User quits
141
                  if event.type == QUIT:
142
                      pygame.quit()
143
                      sys.exit()
144
145
              # Update missile location
146
              if missile firing is True:
147
                  missile y -= MISSILE SPEED
148
                  if missile y < 0:</pre>
149
                      missile firing = False
150
151
              # Update UFO locations
152
              move ufo(ufo 1, ufo width)
153
              move ufo(ufo 2, ufo width)
154
155
              # Update UFO rays
156
              update ray(ufo 1)
157
              update ray(ufo 2)
158
159
              # Check if missile hits a UFO
160
              missile rect = pygame.Rect(missile x, missile y, missile width, missile height)
161
162
              if ufo 1.get('hit') is False and missile firing is True:
163
                  ufo hit = check ufo hit(ufo 1, missile rect, ufo width, ufo height)
164
                  if ufo hit == 'missile destroyed':
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
165
                      missile firing = False
166
                      pygame.mixer.stop()
167
                  elif ufo hit == 'direct hit':
168
169
                      missile firing = False
170
                      score += UFO SCORE * 2
                      ufo 1['hit time'] = UFO HIT TIME
171
                      ufo 1['hit'] = True
172
173
174
                      pygame.mixer.stop()
175
                      spaceship hit sound.play()
176
177
              if ufo 2.get('hit') is False and missile firing is True:
178
                  ufo hit = check ufo hit(ufo 2, missile rect, ufo width, ufo height)
179
                  if ufo hit == 'missile destroyed':
180
                      missile firing = False
181
                      pygame.mixer.stop()
182
183
                  elif ufo hit == 'direct hit':
184
                      missile firing = False
185
                      score += UFO SCORE
                      ufo 2['hit time'] = UFO HIT TIME
186
187
                      ufo 2['hit'] = True
188
189
                      pygame.mixer.stop()
190
                      spaceship hit sound.play()
191
192
              # Update hit UFOs
193
              update hit ufo(ufo 1, SCREEN WIDTH - ufo width, 'left')
194
              update hit ufo(ufo 2, 0, 'right')
195
196
              # Draw background
197
              game screen.blit(background image, [0, 0])
198
199
              # Draw base
200
              game screen.blit(base image, [base x, base y])
201
202
              # Draw missile
203
              if missile firing is True:
204
                  game screen.blit(missile fired image, [missile x, missile y])
205
              else:
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
206
                  game screen.blit(missile image, [base x + MISSILE PLATFORM, base y - missile height])
207
208
              # Draw UFOs
209
              if ufo 1.get('hit time') > 0:
210
                  game screen.blit(ufo 1 exploded image, [ufo 1.get('x_loc'), ufo 1.get('y_loc')])
211
              elif ufo 1.get('hit') is False:
212
                  game screen.blit(ufo 1 image, [ufo 1.get('x loc'), ufo 1.get('y loc')])
213
214
              if ufo 2.get('hit time') > 0:
215
                  game screen.blit(ufo 2 exploded image, [ufo 2.get('x loc'), ufo 2.get('y loc')])
216
              elif ufo 2.get('hit') is False:
217
                  game screen.blit(ufo 2 image, [ufo 2.get('x_loc'), ufo_2.get('y_loc')])
218
219
              # Draw UFO defence rays
220
              if ufo 1.get('ray time') > 0:
221
                  ray x = ufo 1.qet('x loc') + (ufo width - ray width) / 2
222
                  ray y = ufo 1.get('y loc') + ufo height
223
                  if ufo 1.get('ray time') % 4 == 0 or ufo 1.get('ray time') % 5 == 0:
                      game screen.blit(ufo ray image 2, [ray x, ray y])
224
225
                  else:
226
                      game screen.blit(ufo ray image 1, [ray x, ray y])
227
228
              if ufo 2.get('ray time') > 0:
229
                  ray x = ufo 2.get('x loc') + (ufo width - ray width) / 2
230
                  ray y = ufo 2.get('y loc') + ufo height
231
                  if ufo 2.qet('ray time') % 4 == 0 or ufo 2.qet('ray time') % 5 == 0:
232
                      game screen.blit(ufo ray image 2, [ray x, ray y])
233
                  else:
234
                      game screen.blit(ufo ray image 1, [ray x, ray y])
235
236
              # Game over
237
              if game over is True and missile firing is False:
238
                  if score > hi score:
239
                      hi score = score
240
241
                  display game over()
242
243
              # Display score board
244
              score text = 'Score: ' + str(score)
245
              display scoreboard data(score text, 'left')
246
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
247
              missile text = 'Missiles: ' + str(missiles)
248
              display scoreboard data(missile text, 'centre')
249
250
              hi score text = 'Hi: ' + str(hi score)
251
              display scoreboard data(hi score text, 'right')
252
253
              pygame.display.update()
254
              clock.tick(30)
255
256
257
     # Move the UFO
258
      def move ufo(ufo, ufo width):
259
          if ufo.get('hit') is False:
260
              if ufo.get('direction') == 'left':
261
                  ufo['x loc'] -= ufo.get('speed')
262
              elif ufo.get('direction') == 'right':
263
                  ufo['x loc'] += ufo.get('speed')
264
              elif ufo.get('direction') == 'up':
                  ufo['y loc'] -= ufo.get('speed')
265
266
              elif ufo.get('direction') == 'down':
267
                  ufo['y loc'] += ufo.get('speed')
268
269
              # If the UFO goes off the screen left, reset x coordinate and change direction
270
              if ufo.get('x loc') < 0:</pre>
271
                  ufo['x loc'] = 0
272
                  ufo['direction'] = 'right'
273
274
              # If the UFO goes off the screen right, reset x coordinate and change direction
275
              elif ufo.get('x loc') > SCREEN WIDTH - ufo width:
276
                  ufo['x loc'] = SCREEN WIDTH - ufo width
                  ufo['direction'] = 'left'
277
278
279
              # If the UFO goes too high, reset y coordinate and change direction
280
              elif ufo.get('y loc') < UFO UPPER Y:</pre>
                  ufo['y loc'] = UFO UPPER Y
281
282
                  ufo['direction'] = 'down'
283
284
              # If the UFO goes too low, reset y coordinate and change direction
              elif ufo.get('y_loc') > UFO_LOWER_Y:
285
286
                  ufo['y loc'] = UFO LOWER Y
287
                  ufo['direction'] = 'up'
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
288
289
              # If none of the above, then random chance of changing direction
290
              else:
291
                  if ufo.get('direction') == 'up' or ufo.get('direction') == 'down':
292
                      ufo direction chance = random.randint(0, RANDOM VERTICAL CHANGE)
293
                  else:
294
                      ufo direction chance = random.randint(0, RANDOM HORIZONTAL CHANGE)
295
296
                  if ufo direction chance == 1:
297
                      ufo['direction'] = random.choice(UFO DIRECTIONS)
298
299
300
      # Update the status of the UFO ray
301
      def update ray(ufo):
302
303
          # If there is not already a ray, then random chance of there being a ray
304
          if ufo.get('ray time') == 0 and ufo.get('hit') is False:
305
              random ray = random.randint(0, RANDOM RAY)
306
              if random ray == 1:
307
                  ufo['ray time'] = random.randint(RANDOM RAY TIME MIN, RANDOM RAY TIME MAX)
308
309
          # If there is a ray, decrease its time
310
          elif ufo.get('ray time') > 0:
311
              ufo['ray time'] -= 1
312
313
314
      # Has the UFO been hit my the missile
315
      def check ufo hit (ufo, missile rect, ufo width, ufo height):
316
317
          ufo rect = pygame.Rect(ufo.get('x loc'), ufo.get('y loc'), ufo width, ufo height)
318
319
          if missile rect.colliderect(ufo rect):
320
321
              # If the missile collides with the UFO and there is no defence ray, direct hit
322
              if ufo.get('ray time') == 0:
323
                  ufo hit = 'direct hit'
324
325
              # If the missile collides with the UFO and there is a defence ray, missile is destroyed
326
              else:
327
                  ufo hit = 'missile destroyed'
328
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
329
        # If the missile has not collided with the UFO, no hit
330
          else:
331
             ufo hit = 'no hit'
332
333
          return ufo hit
334
335
336
      # Update status of UFO if it has been hit
337
     def update hit ufo(ufo, new x loc, new direction):
338
339
         # UFO has been hit, redice the hit time
340
          if ufo.get('hit time') > 0:
341
             ufo['hit time'] -= 1
342
343
             # When hit time reaches zero, UFO should go off screen
344
             if ufo.get('hit time') == 0:
345
                 ufo['off time'] = UFO OFF TIME
346
347
         # UFO is off screen, reduce the off screen time
348
          elif ufo.get('off time') > 0:
349
             ufo['off time'] -= 1
350
351
              # When off screen time reaches 0, set new UFO location and direction
352
             if ufo.get('off time') == 0:
353
                 ufo['y loc'] = random.randint(UFO UPPER Y, UFO LOWER Y)
354
                 ufo['x loc'] = new x loc
355
                 ufo['direction'] = new direction
356
                 ufo['hit'] = False
357
358
359
     # Display the scoreboard data
360
     def display scoreboard data(scoreboard text, alignment):
361
          display text = font.render(scoreboard text, True, LIGHT YELLOW)
362
          text rect = display text.get rect()
363
364
          text loc = [0, 0]
365
366
          if alignment == 'left':
367
              text loc = [SCOREBOARD MARGIN, SCOREBOARD MARGIN]
368
369
          elif alignment == 'right':
      |0 |1 |2 |3 |4 |5 |6 |7 |8
```

```
|0 |1 |2 |3 |4 |5 |6 |7 |8
              text loc = [SCREEN WIDTH - text rect.width - SCOREBOARD MARGIN, SCOREBOARD MARGIN]
370
371
372
          elif alignment == 'centre':
373
              text loc = [(SCREEN WIDTH - text rect.width) / 2, SCOREBOARD MARGIN]
374
375
          game screen.blit(display text, text loc)
376
377
378
      # Display the game over message
379
      def display game over():
380
          text line 1 = font.render('GAME OVER', True, WHITE)
381
          text rect 1 = text line 1.get rect()
          text line 1 loc = [(SCREEN WIDTH - text rect 1.width) / 2, (SCREEN HEIGHT / 2) - 16]
382
383
384
          text line 2 = font.render('Hit RETURN for new game', True, WHITE)
385
          text rect 2 = text line 2.get rect()
386
          text line 2 loc = [(SCREEN WIDTH - text rect 2.width) / 2, (SCREEN HEIGHT / 2) + 16]
387
388
          game screen.blit(text line 1, text line 1 loc)
          game screen.blit(text line 2, text line 2 loc)
389
390
391
392
     if name == '__main__':
393
         main()
```

394