

CSc11300 Final Project

Question Idea Explain

Gong Qi Chen

07/28/2018

Try It Yourself:

12-1 implement in Alien Invasion folder

I updated the bg_color variable from Setting class to (0, 190, 220), which is a light blue color. This will apply to update_screen() method and change the color of the background.

```
0 # 12-1 implementation Setting background color
1 self.bg_color = (0,190,220)
```

12-2 implement in Alien Invasion folder

I used photoshop to create a Star Fox PNG file with the transparent background color. Then I created a class call Character use the class Ship as super class and paint it in the center of the screen.

```
4 class Character(Ship):
5     def __init__(self, ai_settings, screen):
6         # Initialize the Character using super() to call the Ship constructor
7         super().__init__(ai_settings, screen)
8
9         # Load the character image and get its rect
10        self.image = pygame.image.load('images/Star_Fox.png')
11        self.rect = self.image.get_rect()
12        self.screen_rect = screen.get_rect()
13
14        # Set the ship at the bottom center of the screen at every start
15        self.rect.center = self.screen_rect.center
```

12-3 implement in Alien Invasion folder

I set the rect.centerx and rect.centery to the screen centerx and century in class ship to make the rocket in the center of the screen. I created event.key pygame.K_UP and pygame.K_DOWN for both key press and key release The key pressing will set the movement flag to true and make the ship move up or down continuously. The key releasing will set the movement flag to false and the stop the movement. I also created conditional statements in ship's update method to make ship stay in bound.

```
elif event.key == pygame.K_UP:
    # 12-3 implementation
    # Set the movement flags moving_up to false when the up key is release
    ship.moving_up = False
elif event.key == pygame.K_DOWN:
    # 12-3 implementation
    # Set the movement flags moving_down to false when the down key is release
    ship.moving_down = False
```

```
2 # 12-3 implementation
3 # Movement flag
4 self.moving_right = False
5 self.moving_left = False
6 self.moving_up = False
7 self.moving_down = False
8
9 # 12-3 implementation
10 def update(self):
11     # Update the ship's center base on movement flag and bound its moving range within the screen
12     if self.moving_right and self.rect.right < self.screen_rect.right:
13         self.xcenter += self.ai_settings.ship_speed_factor
14     if self.moving_left and self.rect.left > 0:
15         self.xcenter -= self.ai_settings.ship_speed_factor
16     if self.moving_up and self.rect.top > 0:
17         self.ycenter -= self.ai_settings.ship_speed_factor
18     if self.moving_down and self.rect.bottom < self.screen_rect.bottom:
19         self.ycenter += self.ai_settings.ship_speed_factor
```

12-4 implement in Keys folder

I created a new folder call Keys to implement this question, since it requires to create a new screen. I created an empty screen and check_events() method. The check_event() method print the event.key attribute whenever a pygame.KEYDOWN event is happening. The method was printing an integer number whenever I press a key.

```
6 # 12-4 implementation
7 def run_test():
8     # Initialize pygame, settings, and screen object.
9     pygame.init()
10    ai_settings = Settings()
11    screen = pygame.display.set_mode((ai_settings.screen_width, ai_settings.screen_height))
12    pygame.display.set_caption("Key Testing")
13
14    # Start the main loop for the test
15    while True:
16        gf.check_events()
17
18    run_test()
```

```
5 # 12-4 implementation
6 def check_events():
7     # Respond to keypresses and mouse events
8     for event in pygame.event.get():
9         if event.type == pygame.QUIT:
10             # Exit the program when close the window
11             sys.exit()
12         elif event.type == pygame.KEYDOWN:
13             # print the event.key attribute whenever a pygame.KEYDOWN event
14             print(event.key)
```

12-5 implement in Alien Invasion folder

I created a new PNG file of the ship that is heading right direction and set the position to center left side of the screen (this is conflicting with 12-3 task, ask to position ship in the center). I reset the bullet size to have a longer width and short height in bullet class. Then I modified the update method from Bullet class that the bullet will increment in horizontal direction instead of vertical.

```
9 # 12-5 implementation
10 # Load the ship image and get its rect
11 self.image = pygame.image.load('images/ship_sideway.png')
12 self.rect = self.image.get_rect()
13 self.screen_rect = screen.get_rect()
14
15 # Set the ship at the center of the screen at every start
16 self.rect.left = self.screen_rect.left
17 self.rect.centery = self.screen_rect.centery
```

```
17 # 12-5 implementation
18 # Bullet settings
19 self.bullet_speed_factor = 3.5
20 self.bullet_width = 15
21 self.bullet_height = 3
22 self.bullet_color = 60, 60, 60
23 self.bullets_allowed = 10
```

13-1 implement in Alien Invasion folder

I used the Alien class as a super class to create a Star class, since they are very similar in structure. Then I used the methods from game_functions modules to generate the stars grid, however I purposely make the star image background larger to make the star dispersed. get_number_items_y() and get_number_column() were used to calculate how star in the grid. create_item() were to create the group of stars. Then create_stars() called all of the method above to create the grid.

```

146 def get_number_items_y(ai_settings, item_height):
147     # Determine the number of items that fit in a column
148     available_space_y = ai_settings.screen_height - 2 * item_height
149     number_item_y = int(available_space_y / (2 * item_height))
150     return number_item_y
151
152 def get_number_column(ai_settings, ship_width, item_width):
153     # Determine the number of rows of items that fit on the screen
154     available_space_x = (ai_settings.screen_width - (6 * item_width) - ship_width)
155     number_column = int(available_space_x / (2 * item_width))
156     return number_column
157
158 def create_item(item, items, item_number, column_number, rand_num):
159     # Create an item and place it in the column
160     # Use a random number to randomize the item position
161     item_height = item.rect.height
162     item.y = item_height + 2 * item_height * item_number + rand_num
163     item.rect.y = item.y
164     item.rect.x = 6 * item.rect.width + 2 * item.rect.width * column_number + rand_num
165     items.add(item)

```

```

4 # 13-1 implementation
5 class Star(Alien):
6     # A class to represent single star on the screen
7     def __init__(self, ai_settings, screen):
8         # Initialize the star using super() to call the Alien constructor
9         super().__init__(ai_settings, screen)
10
11         # Load the star image and set its rect attribute.
12         self.image = pygame.image.load('images/star.png')
13         self.rect = self.image.get_rect()

```

13-2 implement in Alien Invasion folder

I generated a random number from -60 to 60, whenever I create a star. Then in create_item() method I used that random number to adjust the x, y coordinates of the star. In this way, the stars will be positioned randomly.

```

50 # 13-1 & 13-2 implementation
51 def create_stars(ai_settings, screen, ship, stars):
52     # Create a grid of stars
53     # Create a star and find the number of stars in a column
54     star = Star(ai_settings, screen)
55     number_stars_y = get_number_items_y(ai_settings, star.rect.height)
56     number_column = get_number_column(ai_settings, ship.rect.width, star.rect.width)
57
58     # Create a grid of stars
59     for column_number in range(number_column):
60         # Create an star and place it in the column
61         for star_number in range(number_stars_y):
62             star = Star(ai_settings, screen)
63             # Create a random number to randomize the position of the stars
64             random_number = randint(-60, 60)
65             create_item(star, stars, star_number, column_number, random_number)

```

13-3 implement in Alien Invasion folder

I created a new Raindrop class, using Alien as its super class, since their structure is very similar. Then I used the same methods that I used to generate stars in 13-1 to generate grid of raindrop. I created an update() method in Raindrop class to make the raindrop drop down. I also created an update_raindrop() method to include the update() method and remove the raindrop from the group once it is out of the screen. Then, I called the update_raindrop inside the main loop.

```

4 # 13-3 implementation
5 class Raindrop(Alien):
6     # A class to represent single rain_drop on the screen
7     def __init__(self, ai_settings, screen):
8         # Initialize the rain_drop using super() to call the Alien constructor
9         super().__init__(ai_settings, screen)
10
11         # Load the rain_drop image and set its rect attribute.
12         self.image = pygame.image.load('images/raindrop.png')
13         self.rect = self.image.get_rect()
14
15     def update(self):
16         # Move the raindrop down
17         self.y += self.ai_settings.raindrop_speed_factor
18         self.rect.y = self.y
19
20 # 13-3 implementation
21 def create_raindrops(ai_settings, screen, ship, raindrops):
22     # Create a grid of raindrops
23     # Create a raindrop and find the number of raindrop in a column
24     raindrop = Raindrop(ai_settings, screen)
25     number_raindrops_y = get_number_items_y(ai_settings, raindrop.rect.height)
26     number_column = get_number_column(ai_settings, ship.rect.width, raindrop.rect.width)
27
28     # Create a grid of raindrops
29     for column_number in range(number_column):
30         # Create a raindrop and place it in the column
31         for raindrop_number in range(number_raindrops_y):
32             raindrop = Raindrop(ai_settings, screen)
33             # Create a random number to randomize the position of the raindrops
34             random_number = randint(-30, 30)
35             create_item(raindrop, raindrops, raindrop_number, column_number, random_number)

```

13-4 implement in Alien Invasion folder

I created a `stead_rain()` method in `game_functions` that will create a new row of rain once the number of raindrop is below the `raindrop_allow` amount. The method structure is like the regular raindrop grid creator method. I included this method inside the `update_raindrop()` method so the deleting and creating of the old and new raindrop will take place at the same time.

```

184 # 13-4 implementation
185 def stead_rain(ai_settings, screen, ship, raindrops):
186     # Create a raindrop and find the number of raindrop in a row
187     raindrop = Raindrop(ai_settings, screen)
188     number_column = get_number_column(ai_settings, ship.rect.width, raindrop.rect.width)
189
190     # Create a row of raindrops
191     for column_number in range(number_column):
192         raindrop = Raindrop(ai_settings, screen)
193         # Create a random number to randomize the position of the raindrops
194         random_number = randint(-60, 60)
195         create_item(raindrop, raindrops, 0, column_number, random_number)

```

13-5 implement in Catch folder

I created a new project folder, `Catch` folder, to implement this question, since it is basically a different game from `Alien Invasion`. The basic idea and structure of this game is like the one we have in `Alien Invasion`. We replaced the ship with the character and aliens with ball. We created `update_ball()` and `check_ball_bottom()` methods in `game_functions` to make sure the ball disappear whenever it collided with the catcher or the bottom.

```

def update_ball(ai_settings, stats, screen, catcher, balls):
    balls.update(ai_settings)

    # Look for ball-catcher collisions.
    if pygame.sprite.spritecollideany(catcher, balls):
        balls.empty()
        create_ball(ai_settings, screen, balls)

    # Look for ball hitting the bottom of the screen.
    check_ball_bottom(ai_settings, stats, screen, balls)

```

```

55 def check_ball_bottom(ai_settings, stats, screen, balls):
56     # Check if ball has reached the bottom of the screen
57     screen_rect = screen.get_rect()
58     for ball in balls:
59         if ball.rect.bottom >= screen_rect.bottom:
60             # The ball is miss call ball_miss method
61             ball_miss(ai_settings, stats, screen, balls)

```

13-6 implement in Catch folder

Same as 13-5, I implement this question in Catch folder. I created ball_miss() in game_functions method to keep track of the number of the balls that hit the bottom, and set the game_active to false when that number is below zero to stop the game.

```

3 def ball_miss(ai_settings, stats, screen, balls):
4     # Respond to ball miss
5     if stats.catcher_left > 0:
6         # Decrement catcher_left
7         stats.catcher_left -= 1
8     else:
9         stats.game_active = False
10
11     # remove old ball and create a new ball
12     balls.empty()
13     create_ball(ai_settings, screen, balls)

```

14-1 implement in Alien Invasion folder

I created a new method call start_game() in game_functions modules, and made it reset the game statistics, the ship position, aliens, and bullets, if the game_active is false. Then I created condition statement in check_keydown_events() to call start_game() if P is pressed.

```

266 # 14-1 implementation
267 def start_game(ai_settings, screen, stats, ship, aliens, bullets):
268     # Start a new game with the method is call
269     # Check if the game is off
270     if not stats.game_active:
271         # Hide the mouse cursor.
272         pygame.mouse.set_visible(False)
273
274         # Reset the game statistics.
275         stats.reset_stats()
276         stats.game_active = True
277
278         # Empty the list of aliens and bullets.
279         aliens.empty()
280         bullets.empty()
281
282         # Create a new fleet and center the ship.
283         create_fleet(ai_settings, screen, ship, aliens)
284         ship.center_ship()
285
44     elif event.key == pygame.K_p:
45         # Start the game with P is press
46         start_game(ai_settings, screen, stats, ship, aliens, bullets)

```

14-2 implement in Target Practice folder

I created a new project folder, Target Practice folder, to implement this question, since it is basically a different game from Alien Invasion. The basic idea and structure of this game is like the one we have in Alien Invasion. We replaced the aliens with the target bar and made the direction horizontal instead of vertical. We created a Button class so we can create a button to display on the screen. Then We created check_play_button() and start_game() in game_functions to make the button only appear as the game is not active and reset the game if the button is click. We also made a method target_miss() to set game_active to false if we miss three shots.

```

3 class Button():
4     def __init__(self, screen, msg):
5         # Initialize button attributes.
6         self.screen = screen
7         self.screen_rect = screen.get_rect()
8         # Set the dimensions and properties of the button.
9         self.width, self.height = 200, 50
10        self.button_color = (0, 255, 0)
11        self.text_color = (255, 255, 255)
12        self.font = pygame.font.SysFont(None, 48)
13        # Build the button's rect object and center it.
14        self.rect = pygame.Rect(0, 0, self.width, self.height)
15        self.rect.center = self.screen_rect.center
16        # The button message needs to be prepped only once.
17        self.prep_msg(msg)
18
19    def prep_msg(self, msg):
20        # Turn msg into a rendered image and center text on the button
21        self.msg_image = self.font.render(msg, True, self.text_color, self.button_color)
22        self.msg_image_rect = self.msg_image.get_rect()
23        self.msg_image_rect.center = self.rect.center
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

14-3 implement in Target Practice folder

Same as 13-5, I implement this question in Target Practice folder. We created `check_bullet_target_collisions()` in `game_functions` method to monitor our hit at the target bar, and increase the speed once we hit it three times. This method called the `increase_speed()` method we created for `Settings` class to increase the speed of the game.

```

78 def check_bullet_target_collisions(ai_settings, target, bullets):
79     # Check for any bullets that hit target
80     # If so, get rid of the bullet and the target
81     bullet = pygame.sprite.spritecollideany(target, bullets)
82     if bullet != None:
83         ai_settings.target_hit += 1
84         bullets.remove(bullet)
85
86     if ai_settings.target_hit == 3:
87         # Destroy existing bullets, speed up game
88         bullets.empty()
89         ai_settings.increase_speed()
90         ai_settings.target_hit = 0
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

14-4 implement in Alien Invasion folder

I created method `read_high_score()` in `GameStats` class and used it to read score from `high_score.txt`. Then it assigned the amount to `high_score` attribute. I also created `write_high_score()` method in `game_functions` modules that write the `high_score` to `high_score.txt`.

Then I called it before all the exit operate to write the high_score to the file.

```

11 # 14-4 implementation
12 # High score is read form the file and should never be reset during the game
13 self.high_score = int(self.read_high_score())
14
15 def reset_stats(self):
16     # Initialize statistics that can change during the game
17     self.ships_left = self.ai_settings.ship_limit
18     self.score = 0
19     self.level = 1
20
21 # 14-4 implementation
22 def read_high_score(self):
23     # Read high_score from the file high_score.txt
24     f = open('high_score.txt', 'r')
25     high_score = f.read()
26     f.close()
27     return high_score

```

```

331 # 14-4 implementation
332 def write_high_score(stats):
333     # Write the high_score into high_score.txt
334     f = open('high_score.txt', 'w')
335     f.write(str(stats.high_score))
336     f.close()
337
338 if event.type == pygame.QUIT:
339     # 14-4 implementation
340     # Write the high score to file before exit
341     write_high_score(stats)
342     sys.exit()

```

```

43 elif event.key == pygame.K_q:
44     # 14-4 implementation
45     # Exit the screen with Q is press
46     # Write the high score to file before exit
47     write_high_score(stats)
48     sys.exit()

```

14-5 implement in Alien Invasion folder

I refactored the program codes by moving the four method calls from Scoreboard constructor to make a new method `prep_images()`. I moved part of `check_play_button()` to create `start_game()` method. I also split some part of `check_bullet_alien_collisions()` to create `start_new_level()`. These new methods made the code implementation easier and cleaner. For example, with `prep_image()` `start_game()` only need to call one method to reset the score board, instead of four.

```

71 # 14-5 implementation
72 def prep_images(self):
73     # Prepare the initial score board image
74     self.prep_score()
75     self.prep_high_score()
76     self.prep_level()
77     self.prep_ships()

```

```

296 # 14-1 & 14-5 implementation
297 def start_game(ai_settings, screen, stats, sb, ship, aliens, bullets):
298     # Start a new game with the method is call
299     # Check if the game is off
300     if not stats.game_active:
301         # Reset the game speed settings
302         ai_settings.initialize_dynamic_settings()
303
304         # Hide the mouse cursor.
305         pygame.mouse.set_visible(False)
306
307         # Reset the game statistics.
308         stats.reset_stats()
309         stats.game_active = True
310
311         # 14-5 implementation
312         # Reset the scoreboard images.
313         sb.prep_images()
314
315         # Empty the list of aliens and bullets.
316         aliens.empty()
317         bullets.empty()

```

```

117     # 14-5 implementation
118     # If the entire fleet is destroyed, start a new level.
119     if len(alien) == 0:
120         start_new_level(ai_settings, screen, stats, sb, ship, alien, bullets)
121
122     # 14-5 implementation
123     def start_new_level(ai_settings, screen, stats, sb, ship, alien, bullets):
124         # Destroy existing bullets, speed up game
125         bullets.empty()
126         ai_settings.increase_speed()
127
128         # Increase level
129         stats.level += 1
130         sb.prep_level()
131
132         # Create new fleet
133         create_fleet(ai_settings, screen, ship, alien)

```

14-6 implement in Alien Invasion folder

I added shooting and explosion sound effect by using pygame.mixer. I set the sound effect variable in Settings class, and called shooting_sound whenever a bullet is shot, called explosion_sound whenever the bullet hit the alien or the ship hit the alien.

```

36     # 14-6 implementation Set sound effects
37     self.shooting_sound = pygame.mixer.Sound('sound/Shooting.wav')
38     self.explosion_sound = pygame.mixer.Sound('sound/Explosion.wav')
39
111     if collisions:
112         # 14-6 implement sound effect for explosion when the aliens is hit
113         ai_settings.explosion_sound.play()
114         for alien in collisions.values():
115             stats.score += ai_settings.alien_points * len(alien)
116             sb.prep_score()
117             check_high_score(stats, sb)
118
267     def ship_hit(ai_settings, screen, stats, sb, ship, alien, bullets):
268         # 14-6 implement sound effect for explosion when the ship is hit
269         ai_settings.explosion_sound.play()

```