

```

1 # Load and initialize pygame
2 import pygame
3 pygame.init()
4
5 # Create a display
6 screenWidth = 640
7 screen = pygame.display.set_mode((screenWidth,screenWidth))
8 pygame.display.set_caption('Space Invaders')
9
10 # Time
11 clock = pygame.time.Clock()
12
13 # Load images
14 background = pygame.image.load('Background.png')
15 ship = pygame.image.load('Ship.png')
16 enemypic = pygame.image.load('enemyship.png')
17
18 # Play music
19 music = pygame.mixer.music.load('music.mp3')
20 pygame.mixer.music.play(-1)
21
22 # Class dimensions of the spaceship
23 class dimensions():
24     def __init__(self,x,y,width,height):
25
26         # Overall variables in class
27         self.x = x
28         self.y = y
29         self.width = width
30         self.height = height
31         self.speed = 2
32         self.hitbox = (self.x, self.y, self.width, self.height)
33         self.health = 1
34         self.visible = True
35
36 # Function which draws the ship
37 def draw(self, screen):
38     if self.visible:
39         self.hitbox = (self.x, self.y, self.width, self.height)
40         screen.blit(ship, (self.x, self.y))
41
42 # Function which processes when the ship gets hit
43 def hit(self):
44     self.visible = False
45     font1 = pygame.font.SysFont('comicans', 50)
46     text = font1.render('The Empire has won', 1, (255,0,0))
47     screen.blit(text, (320 - (text.get_width()/2), 200))
48     pygame.display.update()
49     i = 0
50     while i < 300:
51         pygame.time.delay(10)
52         i += 1
53         for event in pygame.event.get():
54             if event.type == pygame.QUIT:
55                 i = 301
56     pygame.quit()
57

```

```

58 # Class dimensions of the enemy ship
59 class enemy(object):
60     def __init__(self, x, y, width, height, end):
61
62         # Overall variables in the class
63         self.x = x
64         self.y = y
65         self.width = width
66         self.height = height
67         self.end = end
68         self.path = [self.x, self.end]
69         self.speed = 2
70         self.hitbox = (self.x, self.y, self.width, self.height)
71         self.health = 5
72         self.visible = True
73
74 # Function which draws the enemy
75 def draw(self, screen):
76     if self.visible:
77         self.move()
78         screen.blit(enemypic, (self.x, self.y))
79         self.hitbox = (self.x, self.y, self.width, self.height)
80         #pygame.draw.rect(screen, (255, 0, 0), self.hitbox, 2)
81         pygame.draw.rect(screen, (255,0,0), (self.hitbox[0], self.
            hitbox[1] - 20, 50, 10))
82         pygame.draw.rect(screen, (0,255,0), (self.hitbox[0], self.
            hitbox[1] - 20, 50 - (10 * (5-self.health)), 10))
83
84 # Function which treats the movement of the enemy
85 def move(self):
86     if self.speed > 0:
87         if self.x + self.speed < self.path[1]:
88             self.x += self.speed
89         else:
90             self.speed = self.speed * -1
91     else:
92         if self.x - self.speed > self.path[0]:
93             self.x += self.speed
94         else:
95             self.speed = self.speed *-1
96     self.y += 0.05
97
98 # Function which treats enemy getting hit, losing hp each hit,
99 # and in the end dying
100 def hit(self):
101     if self.health > 0:
102         self.health -= 1
103     else:
104         self.visible = False
105         font1 = pygame.font.SysFont('comicsans', 50)
106         text = font1.render('The Rebels have won', 1, (0,255,0))
107         screen.blit(text, (320 - (text.get_width()/2), 200))
108         pygame.display.update()
109         i = 0
110         while i < 300:
111             pygame.time.delay(10)
112             i += 1

```

```

112         for event in pygame.event.get():
113             if event.type == pygame.QUIT:
114                 i = 301
115                 pygame.quit()
116
117 # Class dimensions of the laser
118 class laser(object):
119     def __init__(self,x,y,radius,color,speed):
120
121         # Overall variables of the class
122         self.x = x
123         self.y = y
124         self.radius = radius
125         self.color = color
126         self.speed = 3
127
128 # Function which draws the laser
129 def draw(self,screen):
130     pygame.draw.circle(screen, self.color, (self.x, self.y), self.
        radius)
131
132 # Function where every image is printed
133 def printScreen():
134     screen.blit(background, (0, 0))
135     if shipPos.visible == False:
136         text = font.render('The Empire has won', 1, (255,0,0))
137         screen.blit(text, (320 - (text.get_width()/2), 200))
138     if shipPos.visible == True:
139         shipPos.draw(screen)
140     if stormtrooper.visible == True:
141         stormtrooper.draw(screen)
142     for missile in missiles:
143         missile.draw(screen)
144     for missile in enemymissiles:
145         missile.draw(screen)
146     pygame.display.update()
147
148 # Variables defined for loop
149 font = pygame.font.SysFont('comicsans', 30)
150 shipPos = dimensions(300,400,128,128)
151 missiles = []
152 enemymissiles = []
153 stormtrooper = enemy(100,100,64,64,450)
154 start_ticks = pygame.time.get_ticks()
155 run = True
156
157 # Running the loop
158 while run:
159
160     # Runs the screen at 75 fps
161     clock.tick(75)
162
163     # missile from ship hits the enemy
164     for missile in missiles:
165         if missile.y - missile.radius < stormtrooper.hitbox[1] +
            stormtrooper.hitbox[3] and missile.y + missile.radius >
            stormtrooper.hitbox[1]:

```

```

166         if missile.x + missile.radius > stormtrooper.hitbox[0] and
            missile.x - missile.radius < stormtrooper.hitbox[0] +
            stormtrooper.hitbox[2]:
167             stormtrooper.hit()
168             missiles.pop(missiles.index(missile))
169
170     # missile speed from the ship
171     for missile in missiles:
172         if missile.y > 0:
173             missile.y -= missile.speed
174         else:
175             missiles.pop(missiles.index(missile))
176
177     # Enemy crashes into the ship
178     if stormtrooper.y - stormtrooper.height < shipPos.hitbox[1] +
        shipPos.hitbox[3] and stormtrooper.y + stormtrooper.height >
        shipPos.hitbox[1]:
179         if stormtrooper.x + stormtrooper.width > shipPos.hitbox[0] and
            stormtrooper.x - stormtrooper.width < shipPos.hitbox[0] +
            shipPos.hitbox[2]:
180             shipPos.hit()
181
182     # missile from enemy hits the ship
183     for missile in enemymissiles:
184         if missile.y - missile.radius < shipPos.hitbox[1] + shipPos.
            hitbox[3] and missile.y + missile.radius > shipPos.hitbox[1]:
185             if missile.x + missile.radius > shipPos.hitbox[0] and missile
                .x - missile.radius < shipPos.hitbox[0] + shipPos.hitbox[2]:
186                 shipPos.hit()
187                 enemymissiles.pop(enemymissiles.index(missile))
188
189     # missile speed from the enemy
190     for missile in enemymissiles:
191         if missile.y < 600:
192             missile.y += missile.speed
193         else:
194             enemymissiles.pop(enemymissiles.index(missile))
195
196     # Input buttons for movement for the ship and firing
197     key = pygame.key.get_pressed()
198     if key[pygame.K_SPACE]:
199         if len(missiles) < 1:
200             missiles.append(laser(round(shipPos.x + shipPos.width //2),
                round(shipPos.y + shipPos.height //2), 6, (255,0,0), shipPos.
                speed))
201     if key[pygame.K_LEFT] and shipPos.x > shipPos.speed:
202         shipPos.x -= shipPos.speed
203     if key[pygame.K_RIGHT] and shipPos.x < screenWidth - shipPos.
        width - shipPos.speed:
204         shipPos.x += shipPos.speed
205
206     # Enemy shooting once every second
207     if len(enemymissiles) < 1:
208         seconds = (pygame.time.get_ticks()-start_ticks)/1000
209         if seconds > 1:
210             enemymissiles.append(laser(round(stormtrooper.x +
                stormtrooper.width //2), round(stormtrooper.y + stormtrooper.

```

```
211         height //2), 6, (0,255,0), stormtrooper.speed))
212     # Continuesly runs the loop so it doesnt crash
213     for event in pygame.event.get():
214         if event.type == pygame.QUIT:
215             run = False
216
217     # Prints the screen in the end of the loop
218     printScreen()
219
220 # Exits the program
221 pygame.quit()
```