Chicken Invaders Code Explanation

Tools:

I used the **Turtle** Library to make this game. **turtle** is a pre-installed Python library that enables users to create pictures and shapes by providing them with a virtual canvas. The onscreen pen that you use for drawing is called the **turtle** and this is what gives the library its name.

STEPS:

Step 1: Importing the turtle module that is used to make the game and setting up some variables. (Line 1-16)

```
1  import turtle
2  import math
3  import random
4
5  # Variables
6  start_screen = "src/start_screen1.gif"
7  game_screen = "src/background.gif"
8  game_over_screen = "src/gameoverscreen.gif"
9  shape_chicken = "src/chickff.gif"
10 shape_player = "src/player.gif"
11 shape_missile = "src/missile.gif"
12 shape_enemy = "src/chickff.gif"
13 width, height = 960, 540
14 score, number_of_enemies = 0, 20
15 enemies, enemyspeed = [turtle.Turtle() for _ in range(number_of_enemies)], 0.045
16 # Note : 'enemyspeed' is an experimental value, have to change for different machines.
```

Step 2: Game window and player setup (Line 19-40)

```
19  # Window Setup
20  wn = turtle.Screen()
21  wn.setup(width+60, height+60)
22  wn.bgcolor("black")      You, 4 days ago • init …
23  wn.title("Chiken Invaders")
24  wn.tracer(0)
25  wn.bgpic(start_screen)
26  player_name = wn.textinput("player_name", " Player Name :")
27  wn.bgpic(game_screen)
28  wn.register_shape(shape_chicken)
29  wn.register_shape(shape_player)
30  wn.register_shape(shape_missile)
31
32
33  # Player Spaceship Setup
34  player = turtle.Turtle()
35  player.color("blue")
36  player.shape(shape_player)
37  player.penup()
38  player.speed(0)
39  player.setposition(0, 30-(height/2))
40  player.speed = 0
```

Step 3: Creating the bullet for player, and setting up enemies in random position on screen (Line 42-61)

```
42  # Bullet Setup
43  bullet = turtle.Turtle()
44  bullet.shape(shape_missile)
45  bullet.penup()
46  bullet.speed(0)
47  bullet.setheading(90)
48  bullet.shapesize(0.8, 0.8)
49  bullet.hideturtle()
50  bulletspeed = 15
51  bulletstate = "ready"
52
53
54  # Enemy Setup
55  for enemy in enemies:
56      enemy.shape(shape_enemy)
57      enemy.penup()
58      enemy.speed(0)
59      x = random.randint(int(-width/2 +60), int(width/2 -60))
60      y = random.randint( 80, int(height/2 - 60))
61      enemy.setposition(x, y)
```

Step 4: Defining functions to update the score and move the player to right and left. (Line 66-88)

```
66  # Scoring System
67  score_pen = turtle.Turtle()
68  def update_score(_score, _x = -120, _y = 250, _fontsz = 14):
69      score_pen.speed(0)
70      score_pen.pensize(5)
71      score_pen.color("white")
72      score_pen.penup()
73      score_pen.setposition(_x, _y)
74      scorestring = f"Player : {player_name}  Score : {_score}"
75      score_pen.write(scorestring, False, align="left", font=("Arial", _fontsz, "bold"))
76      score_pen.hideturtle()
77  update_score(0)
78
79
80  # Functions to move player
81  def move_left():
82      player.speed = -25
83      x = player.xcor() + player.speed
84      player.setx(x)
85  def move_right():
86      player.speed = 25
87      x = player.xcor() + player.speed
88      player.setx(x)
```

Step 5: Defining a function to fire bullets, and another function to check collision between any two objects (Bullet vs Enemy or Enemy vs Player). Finally setting up keyboard bindings for moving players and firing bullets. (Line 91-111)

```
91  def fire_bullet():
92      global bulletstate
93      if bulletstate == "ready":
94          bulletstate = "fire"
95          bullet.setposition(player.xcor(), player.ycor() + 20)
96          bullet.showturtle()
97
98
99  def isCollision(t1, t2):
100     distance = math.sqrt(math.pow(t1.xcor()-t2.xcor(),2)+math.pow(t1.ycor()-t2.ycor(),2))
101     if distance < 30:
102         return True
103     else:
104         return False
105
106
107 # Keyboard Bindings
108 wn.listen()
109 wn.onkeypress(move_left, "Left")
110 wn.onkeypress(move_right, "Right")
111 wn.onkeypress(fire_bullet, "space")
```

Step 6: Main Game Loop. First, we update the bullet if it is firing or reset if it goes out of screen. Then, we update the enemy's position. If any enemy hits a wall, move every enemy down 50 pixels, and increase their speed a bit. (Line 114-144)

```
114  # Main Game loop
115  is_alive = True
116  while is_alive:      You, 4 days ago • final without sound
117
118      wn.update()
119
120      if bulletstate == "fire":
121          y = bullet.ycor()
122          y += bulletspeed
123          bullet.sety(y)
124
125      if bullet.ycor() > height-40:
126          bullet.hideturtle()
127          bulletstate = "ready"
128
129
130      for enemy in enemies:
131          x = enemy.xcor()
132          x += enemyspeed
133          enemy.setx(x)
134
135          if enemy.xcor() > ((width/2)-25):
136              for e in enemies:
137                  e.sety(e.ycor() - 50)
138              enemyspeed *= -1.015
139
140          if enemy.xcor() < -((width/2)-25):
141              for e in enemies:
142                  e.sety(e.ycor() - 50)
143              enemyspeed *= -1.015
144
145          if isCollision(bullet, enemy):
```

Step 7: For each enemy, check for collision with enemy and player. In the first case, we relocate the enemy and increase the score. In the case of the latter, we end the game and show a game over screen with scores. (Line 114-170)

```
143             enemyspeed *= -1.015
144
145         if isCollision(bullet, enemy):
146
147             bullet.hideturtle()
148             bulletstate = "ready"
149             bullet.setposition(0, -400)
150
151             x = random.randint(int(-width/2 +60), int(width/2 -60))
152             y = random.randint( 80 , int(height/2 -60))
153             enemy.setposition(x, y)
154
155             score += 10
156             score_pen.clear()
157             update_score(score)
158
159         if isCollision(player, enemy):
160             is_alive = False
161             wn.clear()
162             break
163
164
165 # Game Over Screen
166 while True:
167     wn.bgpic(game_over_screen)
168     update_score(score, -165, -50, 20)
169     wn.update()
170
```