

# PYTHON PROJECT

Topic :- PACMAN GAME

**SUBMITTED TO :**

MRS. AVANI JAIN

**SUBMITTED BY:**

TANMAY AGRAWAL

YASHWARDHAN CHOUKSEY

MANAS SHARMA

## **PACMAN GAME :**

*Pac-Man* is an action maze chase video game; the player controls the eponymous character through an enclosed maze. The objective of the game is to eat all of the dots placed in the maze while avoiding four colored ghosts — Blinky (red), Pinky (pink), Inky (cyan), and Clyde (orange) — that pursue Pac-Man. When Pac-Man eats all of the dots, the player advances to the next level. Levels are indicated by fruit icons at the bottom of the screen. In between levels are short cutscenes featuring Pac-Man and Blinky in humorous, comical situations.

If Pac-Man is caught by a ghost, he will lose a life; the game ends when all lives are lost. Each of the four ghosts has their own unique artificial intelligence (A.I.), or "personality": Blinky gives direct chase to Pac-Man; Pinky and Inky try to position themselves in front of Pac-Man, usually by cornering him; and Clyde will switch between chasing Pac-Man and fleeing from him.<sup>[7]</sup>

Placed at the four corners of the maze are large flashing "energizers" or "power pellets." Eating these will cause the ghosts to turn blue with a dizzied expression and to reverse direction. Pac-Man can eat blue ghosts for bonus points; when a ghost is eaten, their eyes make their way back to the center box in the maze, where the ghost "regenerates" and resumes their normal activity. Eating multiple blue ghosts in succession increases their point value. After a certain amount of time, blue-colored ghosts will flash white before turning back into their normal form. Eating a certain number of dots in a level will cause a bonus item — usually in the form of a fruit — to appear underneath the center box; the item can be eaten for bonus points. To the sides of the maze are two "warp tunnels", which allow Pac-Man and the ghosts to travel to the opposite side of the screen. Ghosts become slower when entering and exiting these tunnels.

The game increases in difficulty as the player progresses: the ghosts become faster, and the energizers' effect decreases in duration, eventually disappearing entirely. Due to an integer overflow, the 256th level loads improperly, rendering it impossible to complete.

## CODE :

```
main.py x project x +
1  """-----Pacman-----"""
2
3  from random import choice
4  from turtle import *
5
6
7  from freegames import floor, vector
8
9
10 state = {'score': 0}
11 path = Turtle()
12 writer = Turtle()
13
14 aim = vector(5, 0)
15 pacman = vector(-40, -80)
16
17 ▼ ghosts = [
18     [vector(-180, 160), vector(5, 0)],
19     [vector(-180, -160), vector(0, 5)],
20     [vector(100, 160), vector(0, -5)],
21     [vector(100, -160), vector(-5, 0)],
22 ]
23 # fmt: off
```

```
main.py x project x +
1  """-----Pacman-----"""
2
3  from random import choice
4  from turtle import *
5
6
7  from freegames import floor, vector
8
9
10 state = {'score': 0}
11 path = Turtle()
12 writer = Turtle()
13
14 aim = vector(5, 0)
15 pacman = vector(-40, -80)
16
17 ▼ ghosts = [
18     [vector(-180, 160), vector(5, 0)],
19     [vector(-180, -160), vector(0, 5)],
20     [vector(100, 160), vector(0, -5)],
21     [vector(100, -160), vector(-5, 0)],
22 ]
23 # fmt: off
```

```

48
49 ▼ def square(x, y):
50     path.up()
51     path.goto(x, y)
52     path.down()
53     path.begin_fill()
54
55 ▼     for count in range(4):
56         path.forward(20)
57         path.left(90)
58
59     path.end_fill()
60
61
62 ▼ def offset(point):
63     "Return offset of point in tiles."
64     x = (floor(point.x, 20) + 200) / 20
65     y = (180 - floor(point.y, 20)) / 20
66     index = int(x + y * 20)
67     return index
68
69
70 ▼ def valid(point):
71     "Return True if point is valid in tiles."
72     index = offset(point)
73

```

```

74 ▼     if tiles[index] == 0:
75         return False
76
77     index = offset(point + 19)
78
79 ▼     if tiles[index] == 0:
80         return False
81
82     return point.x % 20 == 0 or point.y % 20 == 0
83
84
85 ▼ def world():
86     "Draw world using path."
87     bgcolor('black')
88     path.color('blue')
89
90 ▼     for index in range(len(tiles)):
91         tile = tiles[index]
92
93 ▼         if tile > 0:
94             x = (index % 20) * 20 - 200
95             y = 180 - (index // 20) * 20
96             square(x, y)
97
98 ▼         if tile == 1:
99             path.up()
100             path.goto(x + 10, y + 10)

```

```

100         path.goto(x + 10, y + 10)
101         path.dot(4, 'white')
102
103
104     update()
105
106 def move():
107     "Move pacman and all ghosts."
108     writer.undo()
109     writer.write(state['score'])
110
111     clear()
112
113     if valid(pacman + aim):
114         pacman.move(aim)
115
116     index = offset(pacman)
117
118     if tiles[index] == 1:
119         tiles[index] = 2
120         state['score'] += 1
121         x = (index % 20) * 20 - 200
122         y = 180 - (index // 20) * 20
123         square(x, y)
124

```

```

124
125     up()
126     goto(pacman.x + 10, pacman.y + 10)
127     dot(20, 'yellow')
128
129     for point, course in ghosts:
130         if valid(point + course):
131             point.move(course)
132         else:
133             options = [
134                 vector(5, 0),
135                 vector(-5, 0),
136                 vector(0, 5),
137                 vector(0, -5),
138             ]
139             plan = choice(options)
140             course.x = plan.x
141             course.y = plan.y
142
143             up()
144             goto(point.x + 10, point.y + 10)
145             dot(15, 'red')
146
147     update()
148
149     for point, course in ghosts:
150         if abs(pacman - point) < 20:
151

```

```

150     if abs(pacman - point) < 20:
151         return
152
153     ontimer(move, 100)
154
155
156 def change(x, y):
157     "Change pacman aim if valid."
158     if valid(pacman + vector(x, y)):
159         aim.x = x
160         aim.y = y
161
162
163 setup(420, 420, 370, 0)
164 tracer(0,0)
165 writer.hideturtle()
166 writer.goto(160, 160)
167 writer.color('white')
168 writer.write(state['score'])
169 listen()
170 hideturtle()
171 onkey(lambda: change(5, 0), 'Right')
172 onkey(lambda: change(-5, 0), 'Left')
173 onkey(lambda: change(0, 5), 'Up')
174 onkey(lambda: change(0, -5), 'Down')
175
176 world()
177 move()
178 done()

```

## OUTPUT:

