```
"""Pacman, classic arcade game.

Exercises

1. Change the board.
2. Change the number of ghosts.
3. Change where pacman starts.
4. Make the ghosts faster/slower.
5. Make the ghosts smarter.

"""

from random import choice
from turtle import *
from freegames import floor, vector

state = {'score': 0}
path = Turtle(visible=False)
writer = Turtle(visible=False)
aim = vector(5, 0)
pacman = vector(-40, -80)
ghosts = [
    [vector(-180, 160), vector(5, 0)],
    [vector(-180, -160), vector(0, 5)],
    [vector(100, 160), vector(0, -5)],
    [vector(100, -160), vector(-5, 0)],
]
tiles = [
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0,
    0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
    0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
]

def square(x, y):
    "Draw square using path at (x, y)."
    path.up()
    path.goto(x, y)
    path.down()
    path.begin_fill()

    for count in range(4):
        path.forward(20)
```

```
        path.left(90)

    path.end_fill()

def offset(point):
    "Return offset of point in tiles."
    x = (floor(point.x, 20) + 200) / 20
    y = (180 - floor(point.y, 20)) / 20
    index = int(x + y * 20)
    return index

def valid(point):
    "Return True if point is valid in tiles."
    index = offset(point)

    if tiles[index] == 0:
        return False

    index = offset(point + 19)

    if tiles[index] == 0:
        return False

    return point.x % 20 == 0 or point.y % 20 == 0

def world():
    "Draw world using path."
    bgcolor('blue')
    path.color('black')

    for index in range(len(tiles)):
        tile = tiles[index]

        if tile > 0:
            x = (index % 20) * 20 - 200
            y = 180 - (index // 20) * 20
            square(x, y)

            if tile == 1:
                path.up()
                path.goto(x + 10, y + 10)
                path.dot(2, 'white')

def move():
    "Move pacman and all ghosts."
    writer.undo()
    writer.write(state['score'])

    clear()

    if valid(pacman + aim):
        pacman.move(aim)

    index = offset(pacman)

    if tiles[index] == 1:
        tiles[index] = 2
        state['score'] += 1
        x = (index % 20) * 20 - 200
```

```
            y = 180 - (index // 20) * 20
            square(x, y)

        up()
        goto(pacman.x + 10, pacman.y + 10)
        dot(20, 'yellow')

        for point, course in ghosts:
            if valid(point + course):
                point.move(course)
            else:
                options = [
                    vector(5, 0),
                    vector(-5, 0),
                    vector(0, 5),
                    vector(0, -5),
                ]
                plan = choice(options)
                course.x = plan.x
                course.y = plan.y

            up()
            goto(point.x + 10, point.y + 10)
            dot(20, 'red')

        update()

        for point, course in ghosts:
            if abs(pacman - point) < 20:
                return

        ontimer(move, 100)

    def change(x, y):
        "Change pacman aim if valid."
        if valid(pacman + vector(x, y)):
            aim.x = x
            aim.y = y

    setup(420, 420, 370, 0)
    hideturtle()
    tracer(False)
    writer.goto(160, 160)
    writer.color('white')
    writer.write(state['score'])
    listen()
    onkey(lambda: change(5, 0), 'Right')
    onkey(lambda: change(-5, 0), 'Left')
    onkey(lambda: change(0, 5), 'Up')
    onkey(lambda: change(0, -5), 'Down')
    world()
    move()
    done()
```