SNAKE GAME

```
from turtle import *
from random import randrange
from freegames import square, vector

food = vector(0, 0)
snake = [vector(10, 0)]
aim = vector(0, -10)

def change(x, y):
    "change snakes direction"
    aim.x = x
    aim.y = y

def inside(head):
    "return true if head inside boundarie"
    return -300 < head.x < 300 and -300 < head.y < 300

def move():
    "move sanke forward one step"
    head = snake[-1].copy()
    head.move(aim)

    if not inside(head) or head in snake:
        square(head.x, head.y, 9, 'red')
        update()
        return

    snake.append(head)

    if head == food:
        print('Snake length:', len(snake))
```

```
        food.x = randrange(-15, 15) * 10
        food.y = randrange(-15, 15) * 10
    else:
        snake.pop(0)

    clear()

    for body in snake:
        square(body.x, body.y, 9, 'green')

    square(food.x, food.y, 9, 'red')
    update()
    ontimer(move, 100)

hideturtle()
tracer(False)
listen()
onkey(lambda: change(10, 0), 'Right')
onkey(lambda: change(-10, 0), 'Left')
onkey(lambda: change(0, 10), 'Up')
onkey(lambda: change(0, -10), 'Down')

move()
done()
```

CODE:

```
from turtle import *
from random import randrange
from freegames import square, vector

food = vector(0, 0)
snake = [vector(10, 0)]
aim = vector(0, -10)

def change(x, y):
    "change snakes direction"
    aim.x = x
    aim.y = y

def inside(head):
    "return true if head inside boundarie"
    return -300 < head.x < 300 and -300 < head.y < 300

def move():
    "move sanke forward one step"
    head = snake[-1].copy()
    head.move(aim)

    if not inside(head) or head in snake:
        square(head.x, head.y, 9, 'red')
        update()
        return

    snake.append(head)

    if head == food:
        print('Snake length:', len(snake))
        food.x = randrange(-15, 15) * 10
        food.y = randrange(-15, 15) * 10
    else:
        snake.pop(0)

    clear()

    for body in snake:
        square(body.x, body.y, 9, 'green')

    square(food.x, food.y, 9, 'red')
    update()
    ontimer(move, 100)

hideturtle()
tracer(False)
```

```
hideturtle()
tracer(False)
listen()
onkey(lambda: change(10, 0), 'Right')
onkey(lambda: change(-10, 0), 'Left')
onkey(lambda: change(0, 10), 'Up')
onkey(lambda: change(0, -10), 'Down')

move()
done()
```

OUTPUT: