I'm Tiago,

- Software Development Consultancy.
- Python Training for Professionals.
- Product Management and Lead Dev @Promptar.

https://tmont.es/ @setnomt
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> turtle.forward(100)
>>> turtle.up()
>>> turtle.forward(100)
```python
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> turtle.forward(100)
>>> turtle.up()
>>> turtle.forward(100)
>>> turtle.forward(-100)
>>> turtle.down()
>>> turtle.goto(0, 0)
```
$ clear
$ which python3.7
/usr/local/bin/python3.7
$ which python3.7 > game.py
$
#!/usr/local/bin/python3.7

import turtle

turtle.mainloop()
$ clear
$ which python3.7
$ which python3.7
$ chmod 750 game.py
$ vi game.py
#!/usr/local/bin/python3.7

import turtle

turtle.Screen().setup(startx=600, width=640, height=640)
turtle.bgcolor('orange')

turtle.mainloop()
$ clear
$ which python3.7
/usr/local/bin/python3.7
$ which python3.7 > game.py
$ chmod 750 game.py
$ vi game.py

Press ENTER or type command to continue.
turtle.bgcolor('orange')

GRID_SIZE = 80
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)

turtle.mainloop()
$ clear

$ which python3.7
/usr/local/bin/python3.7

$ which python3.7 > game.py

$ chmod 750 game.py

$ vi game.py

Press ENTER or type command to continue

Press ENTER or type command to continue
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)

turtle.speed(0)
turtle.hideturtle()
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)

--- INSERT ---
```python
def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)

turtle.speed(0)
turtle.hideturtle()
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
```
shell returned 1

Press ENTER or type command to continue.
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')
turtle.mainloop()
bash-3.2$ ls *gif
beast.gif     player.gif
bash-3.2$
Author statement of CC0 (public domain)

https://opengameart.org/content/some-characters

Assets downloaded from opengamegraphics.com
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')
turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
turtle.onkey(lambda: move_player(-1, 0), 'Left')
turtle.onkey(lambda: move_player(1, 0), 'Right')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
"game.py" 43L, 976C written
https://opengameart.org/content/sprites

Assets downloaded from opengameart.org

bash-3.2$ exit

Press ENTER or type command to continue.

Press ENTER or type command to continue.
Assets downloaded from opengameart.org

bash-3.2$ exit
Press ENTER or type command to continue
Press ENTER or type command to continue
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')
player.up()

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
"game.py" 44L, 988C written
Assets downloaded from opengamegraphics

bash-3.2$ exit

Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

def actor(image):
    t = turtle.Turtle()
    turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
def actor(image):
    t = turtle.Turtle()
    turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)

-- INSERT --
bash-3.2$ exit
Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
Press ENTER or type command to continue.
bash-3.2$ exit
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
t = turtle.Turtle()
turtle.register_shape(image)
t.shape(image)
t.up()
return t

beast = actor('beast.gif')
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        move_beast()
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        move_beast()
    else:
        if random.random() < MOVE_BEAST_ODDS:
            move_beast()

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)

"game.py" 66L, 1471C written
#!/usr/local/bin/python3.7

import random
import turtle

turtle.Screen().setup(startx=600, width=640, height=640)
turtle.bgcolor('orange')

GRID_SIZE = 80
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

MOVE_BEAST_ODDS = 0.2

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)
bash-3.2$ exit
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
Press ENTER or type command to continue
def actor(image):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')
karma = actor('circle')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

def attempt_capture():

def actor(image, *, color='black'):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        turtle.register_shape(image)
    t.shape(image)
    t.color(color)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

-- INSERT --
player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
attempt_capture()

def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        update_karma(KARMA_CAPTURE)
        move_beast()
    else:
        update_karma(KARMA_MOVE)
        if random.random() < MOVE_BEAST_ODDS:
            move_beast()

def update_karma(dk):
    karma.forward(dk)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
#!/usr/local/bin/python3.7

import random
import turtle

turtle.Screen().setup(startx=600, width=640, height=640)
turtle.bgcolor('orange')

GRID_SIZE = 80
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

MOVE_BEAST_ODDS = 0.2
KARMA_MOVE = -20
KARMA_CAPTURE = 100

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.goto(x2, y2)
"game.py" 79L, 1782C written
if random.random() < MOVE_BEAST_ODDS:
    move_beast()

def update_karma(dk):
    karma.forward(dk)
    karma_x, _ = karma.position()
    if karma_x >= GRID_MAX:
        end_game('Victory!')
    elif karma_x <= -GRID_MAX:
        end_game('Defeat...')

def end_game(message):
    player.write(
        message,
        align='center',
        font=('Helvetica', 64, 'bold'),
    )

def move_beast():

Press ENTER or type command to continue.

Victory! Victory! Victory! Victory!
```python
def end_game(message):
    player.write(
        message,
        align='center',
        font=('Helvetica', 64, 'bold'),
    )
    for key in ('Up', 'Down', 'Left', 'Right'):
        turtle.onkey(None, key)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
turtle.onkey(lambda: move_player(-1, 0), 'Left')
```

"game.py" 93L, 2148C written
```python
t = turtle.Turtle()
if image.endswith('.gif'):
    turtle.register_shape(image)
t.shape(image)
t.color(color)
t.up()
return t

beast = actor('beast.gif')
player = actor('player.gif', color='white')
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
attempt_capture()

def attempt_capture():
```
t = turtle.Turtle()
if image.endswith('.gif'):
    turtle.register_shape(image)
    t.shape(image)
    t.color(color)
    t.up()
return t

beast = actor('beast.gif')
player = actor('player.gif', color='white')
player.steps = 0
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    player.steps += 1
    attempt_capture()
"game.py" 95L, 2203C written
```python
def update_karma(dk):
    karma.forward(dk)
    karma_x, _ = karma.position()
    if karma_x >= GRID_MAX:
        end_game('Victory!')
    elif karma_x <= -GRID_MAX:
        end_game('Defeat...')

def end_game(message):
    player.write(
        message,
        align='center',
        font=('Helvetica', 64, 'bold'),
    )
    karma.write(
        f'Player steps: {player.steps},
        align='center',
        font=('Helvetica', 32, 'bold'),
    )
```

"game.py" 100L, 2323C written
3 steps

Defeat...
for key in ('Up', 'Down', 'Left', 'Right'):
    turtle.onkey(None, key)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
turtle.onkey(lambda: move_player(-1, 0), 'Left')
turtle.onkey(lambda: move_player(1, 0), 'Right')

move_beast()

turtle.mainloop()
#!/usr/local/bin/python3.7

import random
import turtle

turtle.Screen().setup(startx=600, width=640, height=640)
turtle.bgcolor('orange')

GRID_SIZE = 50
GRID_SPAN = 5
GRID_MAX = GRID_SIZE * GRID_SPAN

MOVE_BEAST_ODDS = 0.2
KARMA_MOVE = -20
KARMA_CAPTURE = 100

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
turtle.speed(0)
turtle.hideturtle()
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

def actor(image, *, color='black'):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        turtle.register_shape(image)
    t.shape(image)
    t.color(color)
    t.up()
    return t
line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

```python
def actor(image, *, color='black'):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        if image[-4:] == '.gif':
            turtle.register_shape(image)
```
line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

def actor(image, *, color='black'):
    t = turtle.Turtle()

if image.endswith('.jpeg'):
    if image[-4:] == '.jpeg':
        turtle.register_shape(image)
if image.endswith('.gif'):
    turtle.register_shape(image)
    t.shape(image)
    t.color(color)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif', color='white')
player.steps = 0
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    player.steps += 1
    attempt_capture()
move_beast()

def update_karma(dk):
    karma.forward(dk)

    karma_x, _ = karma.position()

    karma_x = karma.position()[0]

    if karma_x >= GRID_MAX:
        end_game('Victory!')
    elif karma_x <= -GRID_MAX:
        "game.py" 111L, 2361C written
move_beast()

def update_karma(dk):
    karma.forward(dk)

    karma_x, _ = karma.position()
    # rhs is iterable (think list-like)
    # two elements
    # we care about the first

    karma_x = karma.position()[0]
    # [42]?
    # ['key']?...
font=('Helvetica', 32, 'bold'),
)

for key in ('Up', 'Down', 'Left', 'Right'):
    turtle.onkey(None, key)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
turtle.onkey(lambda: move_player(-1, 0), 'Left')
turtle.onkey(lambda: move_player(1, 0), 'Right')

move_beast()
Thoughts
--------

- Python's Standard Library often surprises you.
- Direct, in-your-face Python is effective and fun.
- Having fun is one great way of learning new things.
- The kids did learn something.
Don't do this at work?

Extending the game:

- Distributed network-based play?
- Create an event loop for async turtles?
- Learn about, and fix race-conditions?
- Have an AI/ML-powered beast?

What could we learn?
Learning
Growing
Having Fun