

Are we Living in a Computer Simulation?

Python Programming 2025

Matthew-Stephen Wileman

Bianca-Gabriela Leoveanu



Introduction

🧠 Is reality simulated?

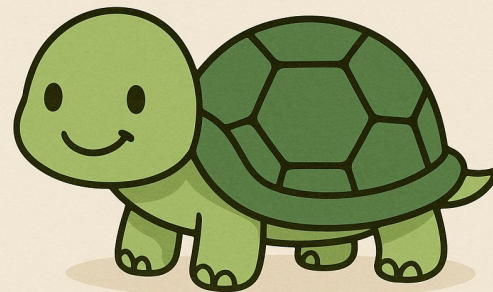
- We explored the idea that our world might be a computer simulation.

🐢 How?

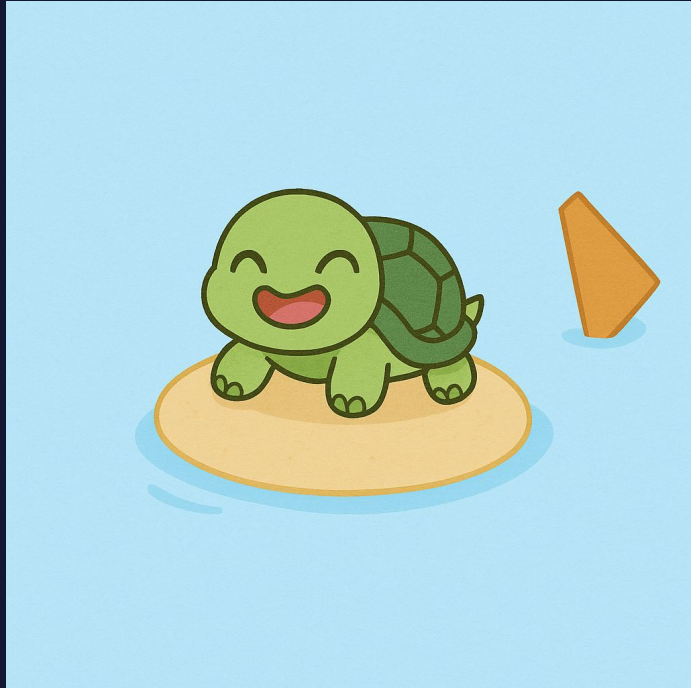
- Using Python's turtle module, we created a simple world and tested for grid-like artifacts.

🔍 Goal

- Find visual or timing clues that could suggest a simulated reality.



Aim



Goal

- Build a Python-based simulation of a virtual world called *Yertle's Pond*.

Focus

- Compare movement costs along vs. across a pixel grid to detect computational patterns.

Why?

- To explore if such motion artifacts could be signs of an underlying simulation.

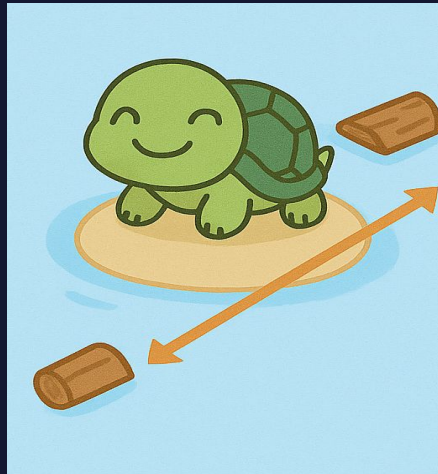
Scope

🌊 Design a virtual pond with elements like a turtle, water, and a log using Python turtle graphics

🐢 Program Yertle to move in both straight and diagonal paths

🕒 Measure timing differences between horizontal and angled movement across a grid

🧠 Analyze results as potential clues of an underlying simulated structure.



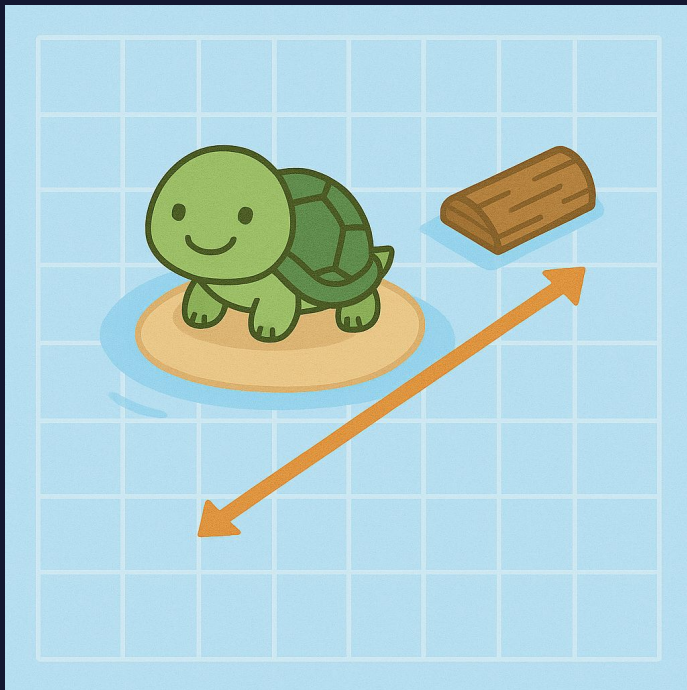
Methodology

🐢 How We Tested the Hypothesis

- We created a simplified pond world using Python's turtle graphics to explore if motion artifacts could reveal an underlying simulation.

🧱 Steps We Followed

- Built a grid-based pond using turtle graphics
- Placed Yertle on a log and programmed both horizontal and diagonal movements
- Timed each motion using Python's timing tools
- Compared results to check for computational bias in different directions



Theoretical Part

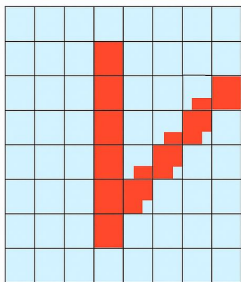
Simulation Argument



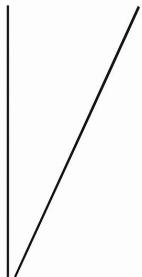
Core Idea

Nick Bostrom's Simulation Argument suggests we could be digital beings inside a computer simulation.

Directional Artifacts on a Pixel Grid



Pixelated movement



Smooth lines



Key Concepts

- **Grids & Pixels:** Simulations run on discrete grids → world is made of pixels.
- **Anisotropy:** Movement differs by direction → diagonal = horizontal
- **Granularity:** Minimal possible unit → affects how smooth things appear.



Our Simulation

- Turtle motion is smoother on horizontal lines than on diagonal ones.
- This reflects grid-based constraints like **anisotropy** and **granularity**.



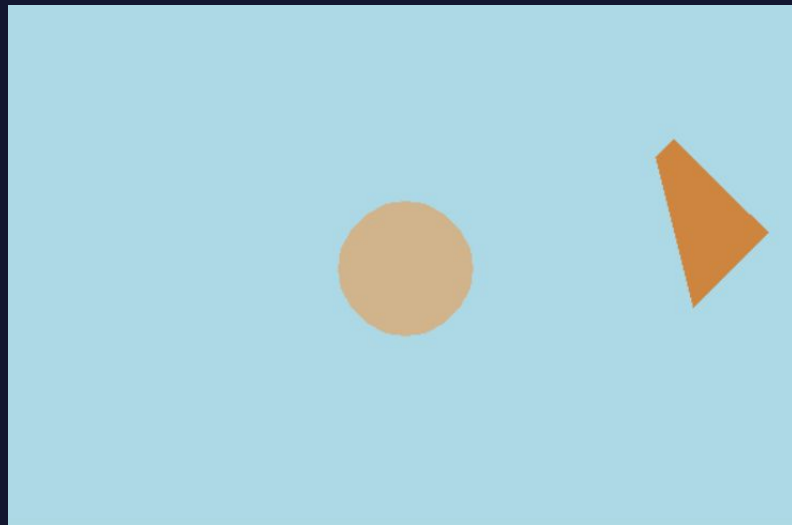
Philosophical Insight

If our world is a simulation, such directional artifacts might also be detectable in reality.

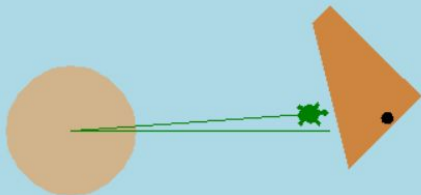
The Pond Simulation

🌐 Setting Up the Environment:

- 🎨 What We Simulated
 - We modeled a small pond using Python's turtle graphics.
- 🌴 Environment:
 - Light blue background (600x400 pixels)
 - Mud Island: Oval-shaped tan patch made from a turtle circle
 - Floating Log & Knothole: Added for detail and variety
 - Yertle the Turtle: Our explorer moving across this world
- 🧠 Why This Matters
 - The visual simulation helps us metaphorically explore signs of artificiality — like those we might find in a simulated universe.



The Pond Simulation



Drawing the Log & Knothole

- A rectangular log is created with a turtle object
- It's rotated diagonally for a natural floating look
- A small knothole is added using a 5-pixel radius circle for detail



Animating Yertle the Turtle

- Yertle swims from the island to the log and back
- Then swims off again at an angled path, representing memory drift
- This motion reflects unpredictability — a subtle clue in simulated systems

Measuring the Cost of Crossing the Lattice

Experiment Goal

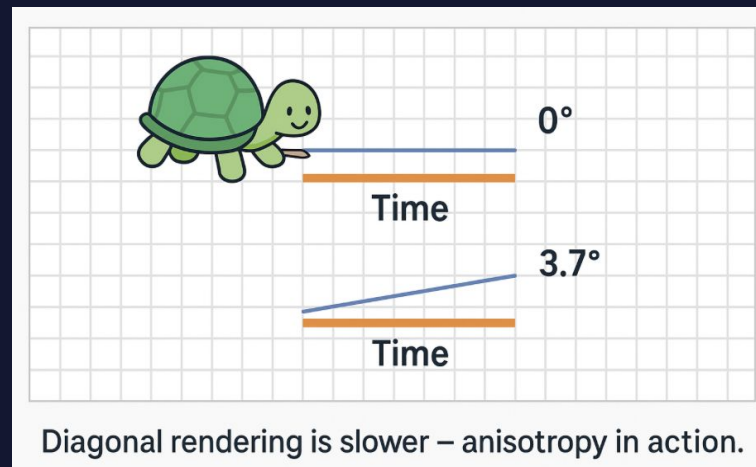
Compare drawing time between horizontal and diagonal lines to observe **anisotropy** in simulated grids.

Simulation Setup

- **Tool:** Python's `turtle` module
- **Lines:** Same length → Horizontal (0°) vs. Slight diagonal (3.7°)
- **Runs:** 20 times each for better accuracy
- **Speed:** Max (`speed=0`) to remove animation delays

Finding

- **Diagonal lines take longer**, confirming anisotropic behavior
- Caused by **more complex math** (trigonometry vs simple addition)






Performance Results

Drawing Duration

Runs	Speed	Direction	Avg. Time (s)	Ratio = $\frac{\text{Average time (angled)}}{\text{Average time (horizontal)}}$
20	0	0°	0.01585	-
20	0	3.7°	0.01882	1.19×
500	0	0°	0.02369	-
500	0	3.7°	0.05293	2.23×
1000	0	0°	0.02905	-
1000	0	3.7°	0.08258	2.84×
500	6	0°	0.53798	-
500	6	3.7°	0.82748	1.53×

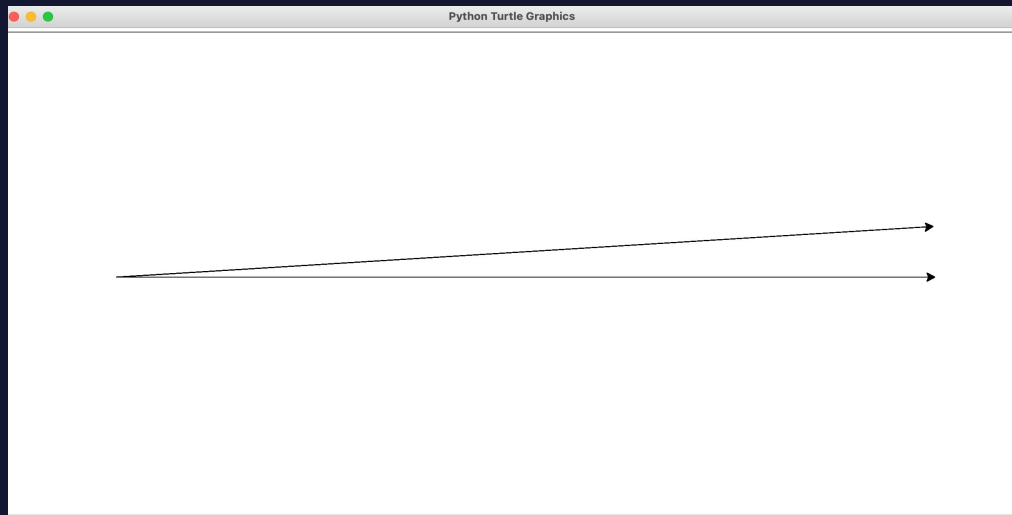
Table 1: Results Table

-  Diagonal lines took up to 2.84x longer to draw than horizontal ones
-  Extra time caused by trigonometric calculations
-  Results consistent across multiple runs and speeds

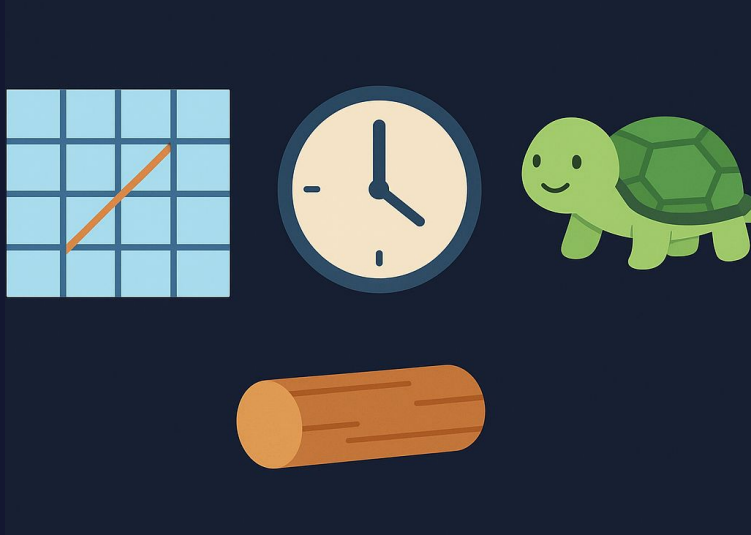
Final Output

Turtle Movement

- 🐢 Turtle moved along 0° and 3.7° paths
- 👁️ Angled line shows stair-stepping artifacts
- 🧠 Confirms anisotropy in grid-based simulations



Conclusion



What We Learned

- A simple turtle-based simulation can help explore big ideas about computational realities.

Key Insights

- Grid-based environments reveal movement constraints
- Anisotropy and timing differences can hint at underlying structure
- Such artifacts mirror techniques physicists use to test for simulation clues

Takeaway

- Even basic simulations can offer meaningful metaphors for detecting signs of artificiality in our own universe.

Github repository:

<https://github.com/wilemanstephen/Python-Programming.git>

