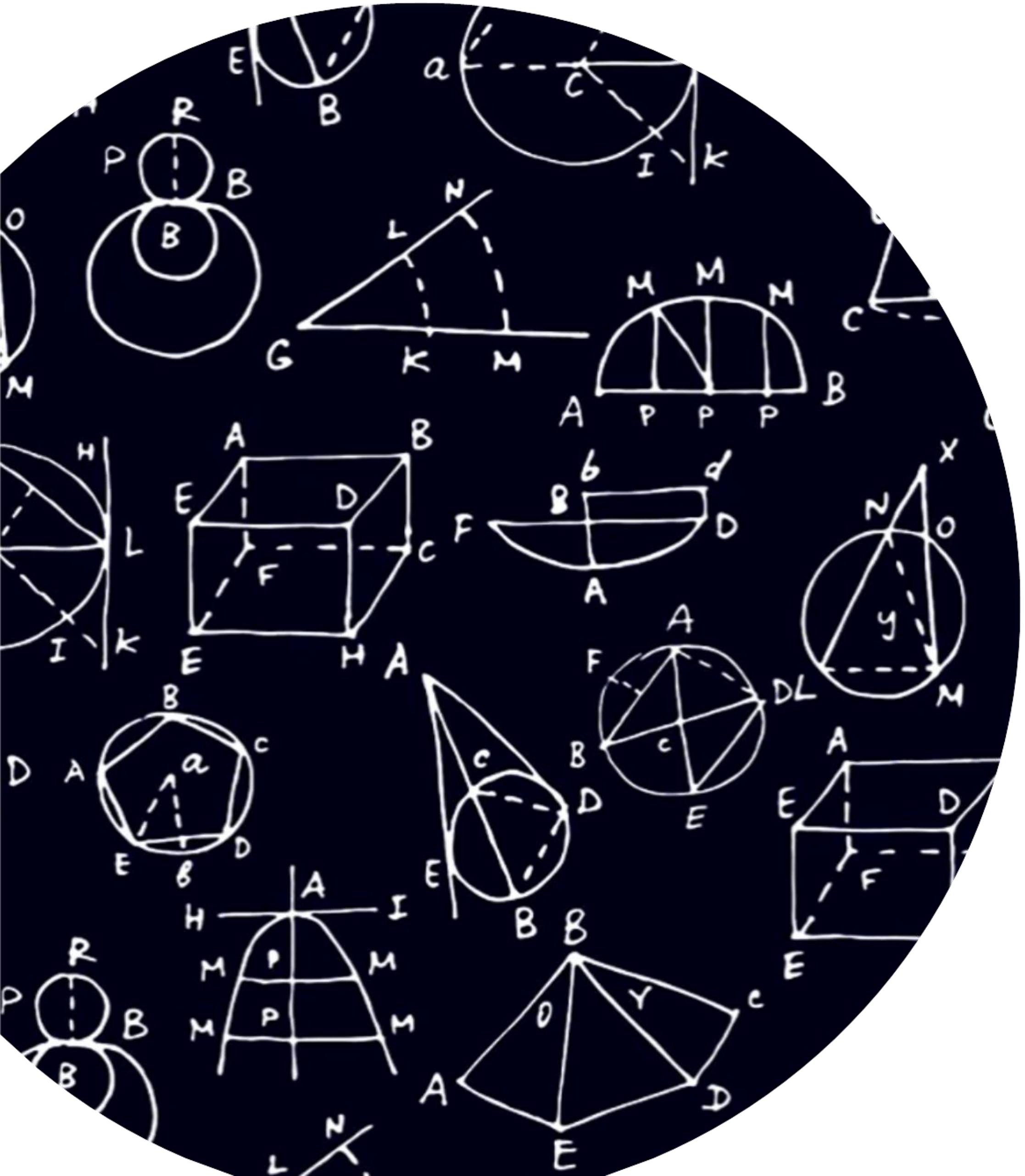


Genetic Programming

Presenter:

Shaksham Kapoor

https://www.linkedin.com/in/shakshamkapoor/

📖 Agenda

- Why use GA? 🤔
- Genetic Algorithms – Introduction ✍️
- Problem Statement ✍️
- Darwin Natural Selection 🤠
- GA – Steps 👣
- Code Walkthrough 🖥️

Why use GA? 🤔

- Shakespearean Monkey or “The Infinite Monkey Theorem”

Why use GA?

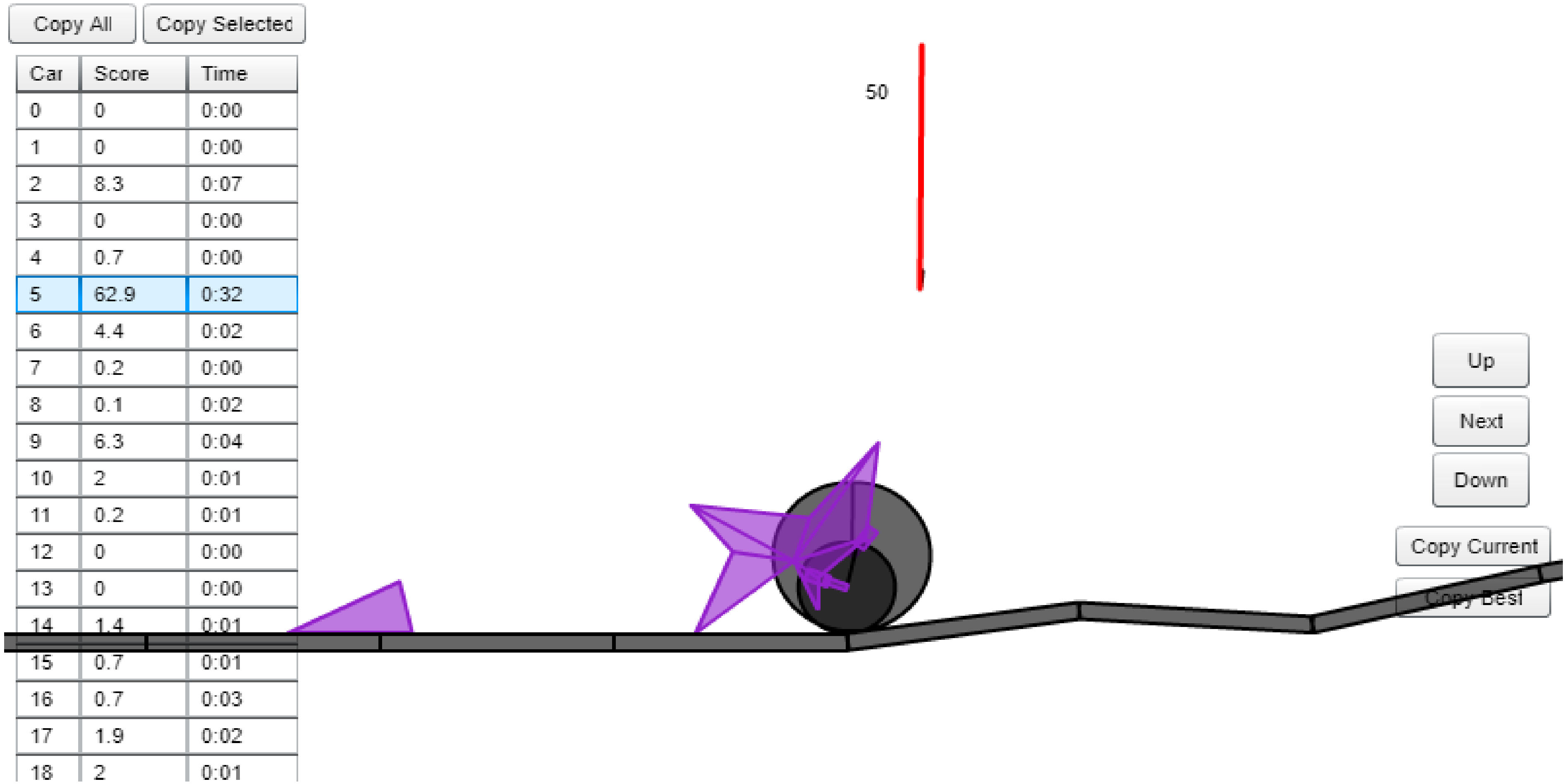
Brute Force Search

"to be or not to be that is the question"

```
likelihood of typing a "t" randomly:     1/27
likelihood of typing "to" randomly:      1/27 * 1/27

likelihood of typing entire phrase:      (1/27) ^ 39 or

1 in
66,555,937,033,867,822,607,895,549,241,096,482,953,017,615,834,735,226,163

A computer simulation with 1 million phrases per second would take:

~ 9,719,096,182,010,563,073,125,591,133,903,305,625,605,017 years.

Age of the universe: 13,750,000,000 years (estimated).
```

Interesting Stuff 😍

- BoxCar2D: http://boxcar2d.com/



Genetic Algorithm - Intro

- Genetic algorithms are a specific approach to optimization problems that can estimate known solutions and simulate evolutionary behavior in complex systems.

- This can further be broken down into the following three parts:
  - Traditional GA
  - Interactive Selection
  - Ecosystem Simulation

Problem Statement ✍️

- Find the phrase “to be or not to be” in a sea of possibilities?
- **Using Brute Force**: Wait for Gazillion years !!!!
- **Using GA:**

```
Best phrase:
To be or not to be.

total generations: 369
average fitness: 0.7897368421052614
total population: 200
mutation rate: 1%
```

Darwinian Natural Selection

- Heredity.
- Variation.
- Selection.

GA - Steps

SETUP:

Step 1: ***Initialize***. Create a population of N elements, each with randomly generated DNA.

LOOP:

Step 2: ***Selection***. Evaluate the fitness of each element of the population and build a mating pool.

Step 3: ***Reproduction***. Repeat N times:

a) Pick two parents with probability according to relative fitness.
b) Crossover—create a “child” by combining the DNA of these two parents.
c) Mutation—mutate the child’s DNA based on a given probability.
d) Add the new child to a new population.

Step 4. Replace the old population with the new population and return to Step 2.

When do we stop?

1

At a predetermined generation number – Not very effective!!!

2

When we reach a predetermined goal. – Again, not effective!!

3

When our solution (local maximum) hasn’t changed in X generations – Effective!!!

Real Life Applications

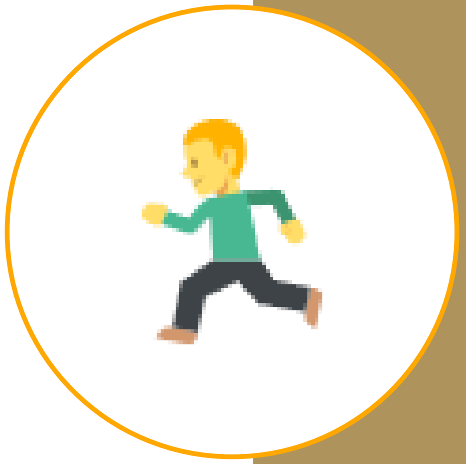
- Automotive Design
- Robotics
- Scheduling routes and travel (TSP!)

🖥️ Code Walkthrough

- **Objective:** Maximize the output of an equation.
- **Programming Language:** Python

Where to go next?

- The Nature of Code - https://natureofcode.com/
- Good Introduction - https://blog.floydhub.com/introduction-to-genetic-algorithms/
- Fun Source - http://geneticprogramming.com/
- Java - https://processing.org/
- JavaScript - https://p5js.org/
- Python - https://gplearn.readthedocs.io/en/stable/
- And do check out Wikipedia!!



Any Questions?