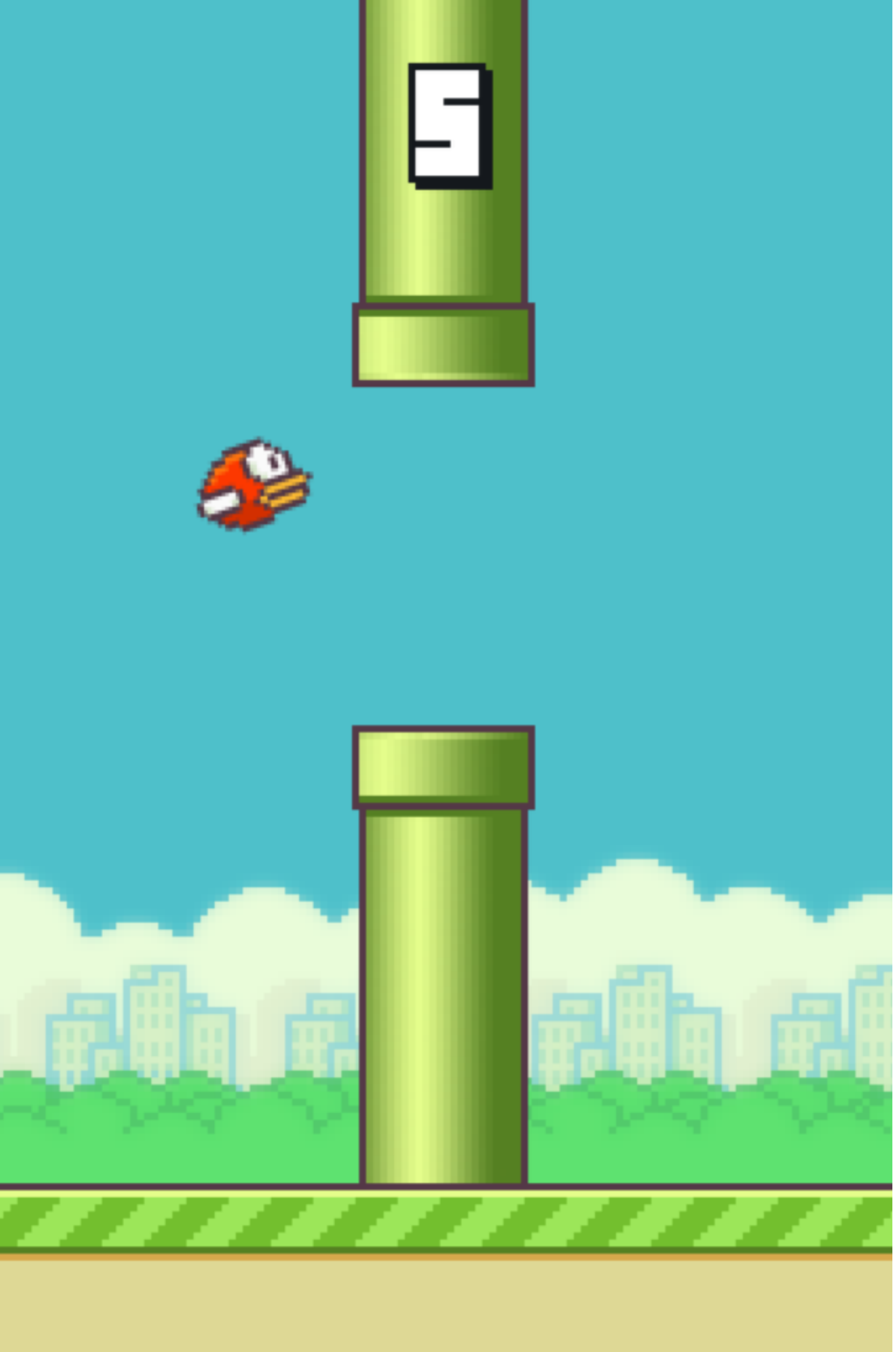
FLAPPY-LEARN

genetic algorithm learns to fly



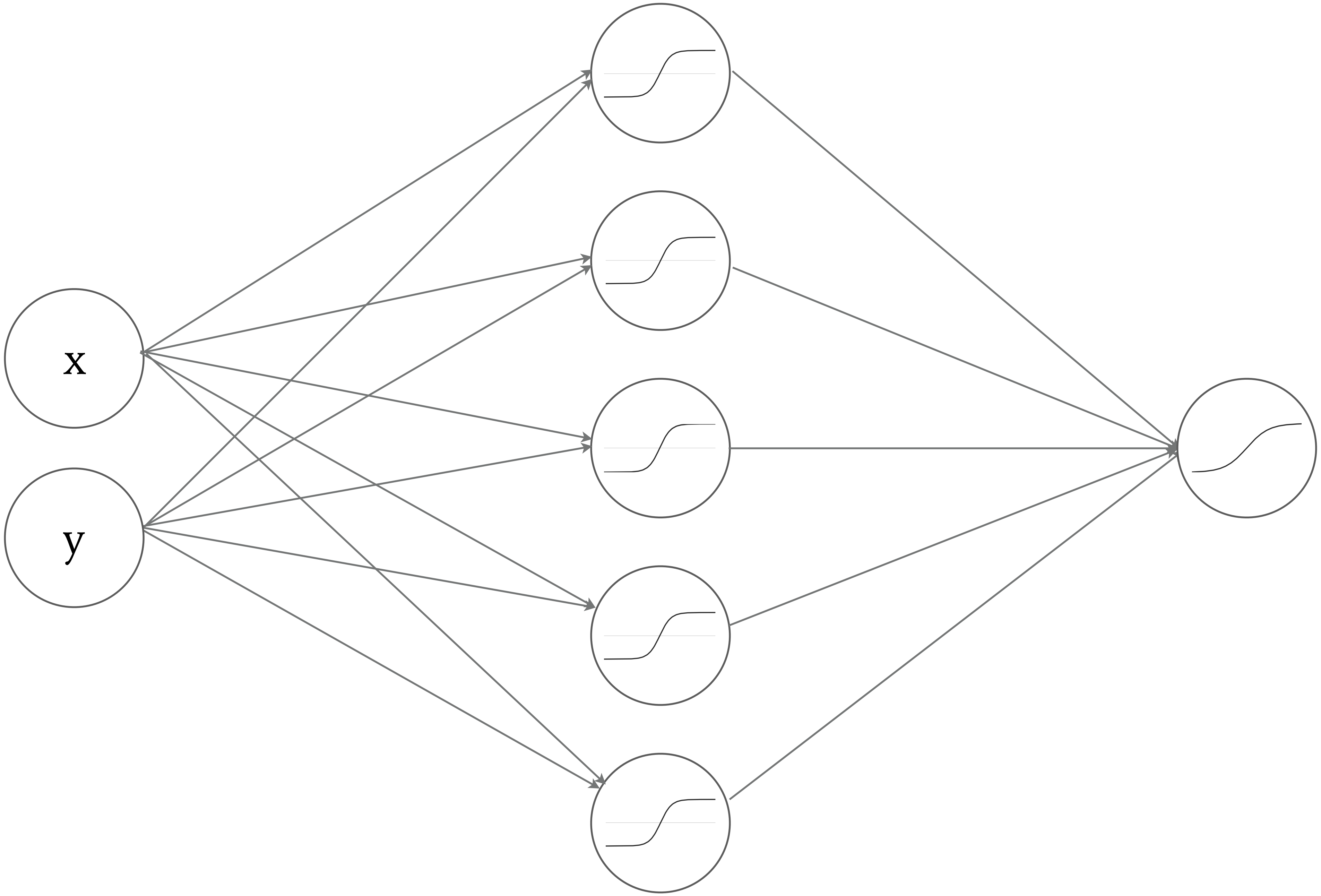
FLAPPY BIRD

- released in 2013
- notoriously difficult
- addictive *(caused major distractions among 1st year FMI UB students)*

- **objective**: survive as long as possible
- **controls**: tap to "jump"
- **rules**: die upon hitting a pipe or leaving the screen

APPROACH

THE BIRD'S BRAIN



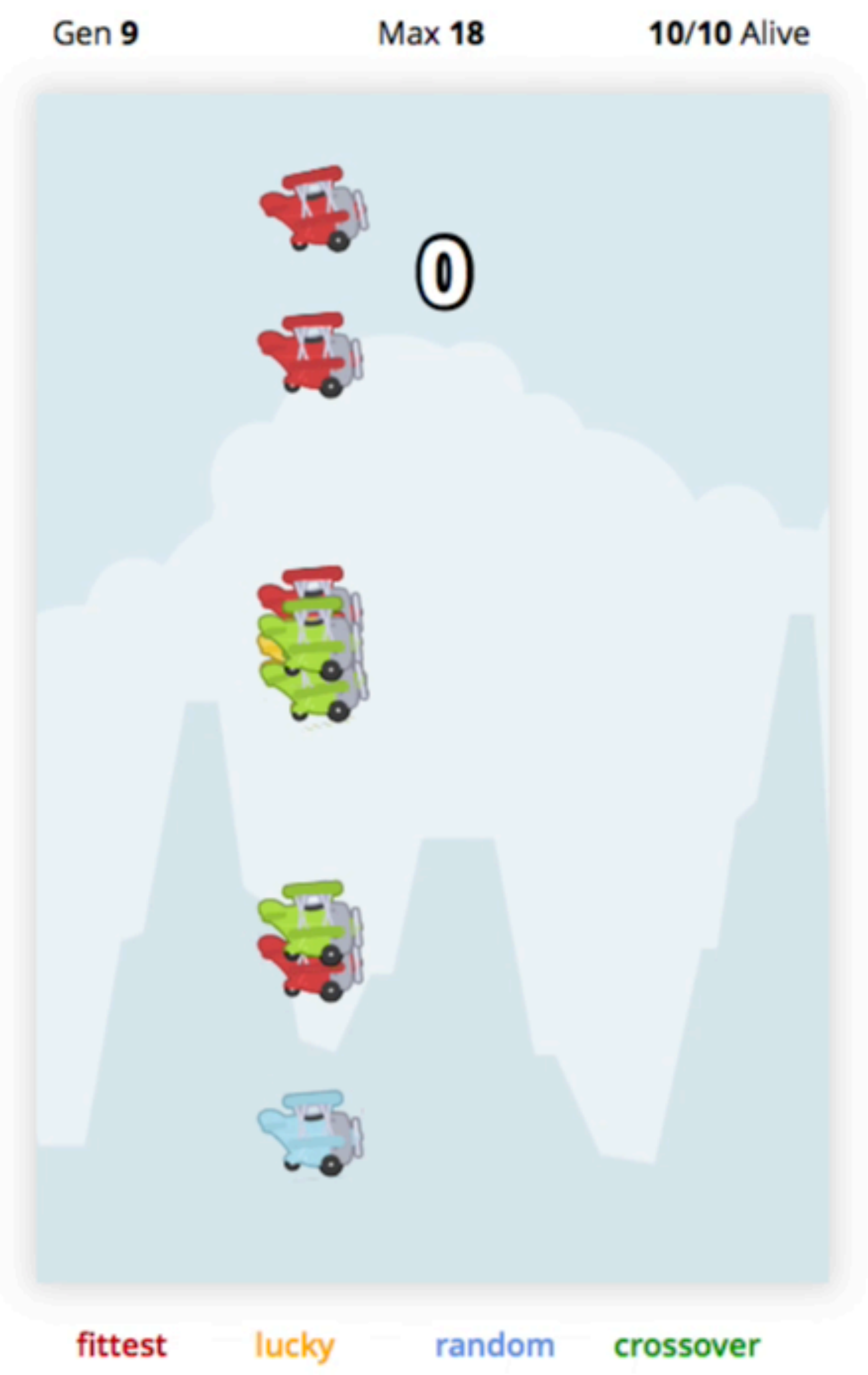
THE BIRD'S BRAIN

- simple neural network
- inputs: horizontal & vertical distance to next gap
  - distance to screen edge "contained" in *y* component
  - normalized *[–1, +1]*
- hidden layer: 5 neurons
  - *tanh* non-linearity (we don't use backprop)
- output: % willingness to jump
  - *sigmoid* activation (it's a brain, right?)
  - fires if over *0.5*

THE WORLD

- barrier position: anywhere from top to bottom edge
- gaps get increasingly smaller
  - until birds can no longer fit ⇒ force next generation
  - decay exponentially
- a bird's brain is simulated multiple times per second

A Generation Taking Off



GENETIC OPERATORS

GENETIC SETTING

- **genome**: a bird's brain
- **population**: 10 genomes (one generation)
- **fitness**: the bird's age

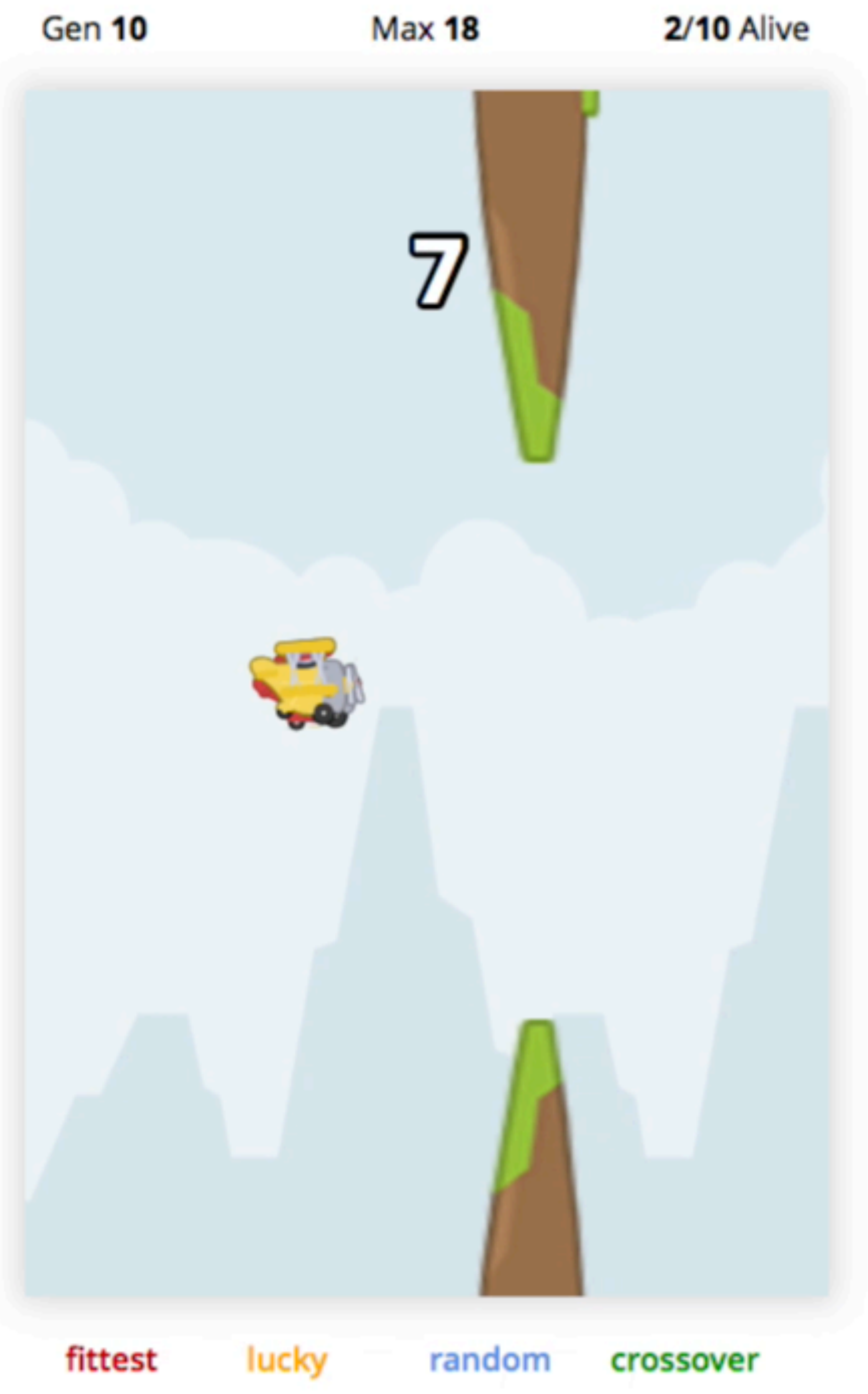
- first generation: random network
  - weights: ~ *U[–1, +1]*
  - biases: zero
- next generation: *select*, *crossover* and *mutation*
  - if none managed to pass a single barrier, wipe out (but keep best one)

SELECTION

- *who advances to the next generation*
- fitness-proportionate roulette wheel selection

- keep the 3 fittest (elitism)
- pick 2 lucky ones, at random
- add 1 random genome
  - combat staleness

Having Faith Pays Off



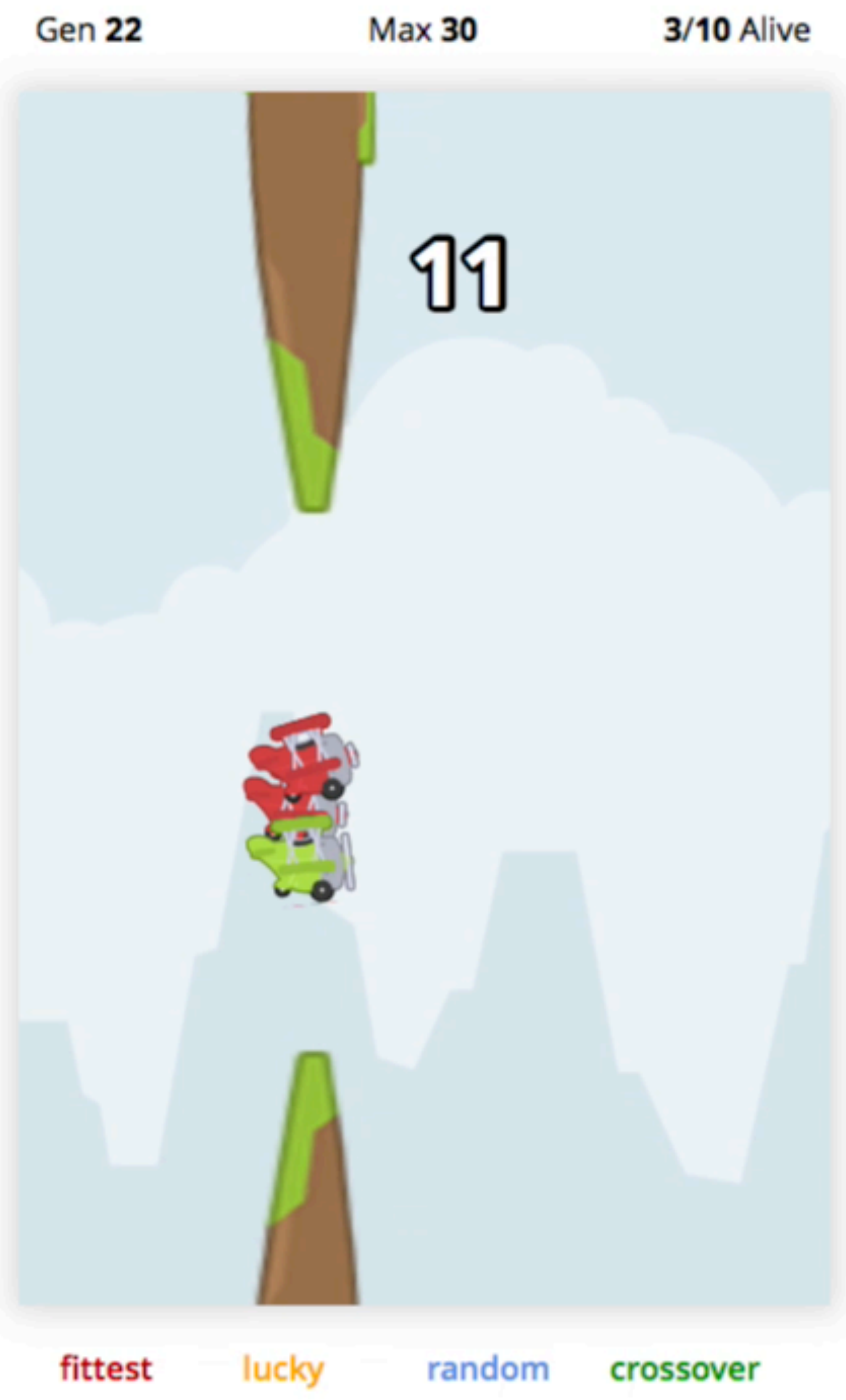
CROSSOVER

- *offspring of two parents (one dominant, one recessive)*
- uniform recombination, binary mask

- weights: inherited from dominant
- biases: 25% from recessive

- 3 couples:
  - fittest two
  - two random winners
  - two randomly selected

Overcomes its Parents



Gen **22** Max **30** **3/10** Alive

fittest lucky random crossover

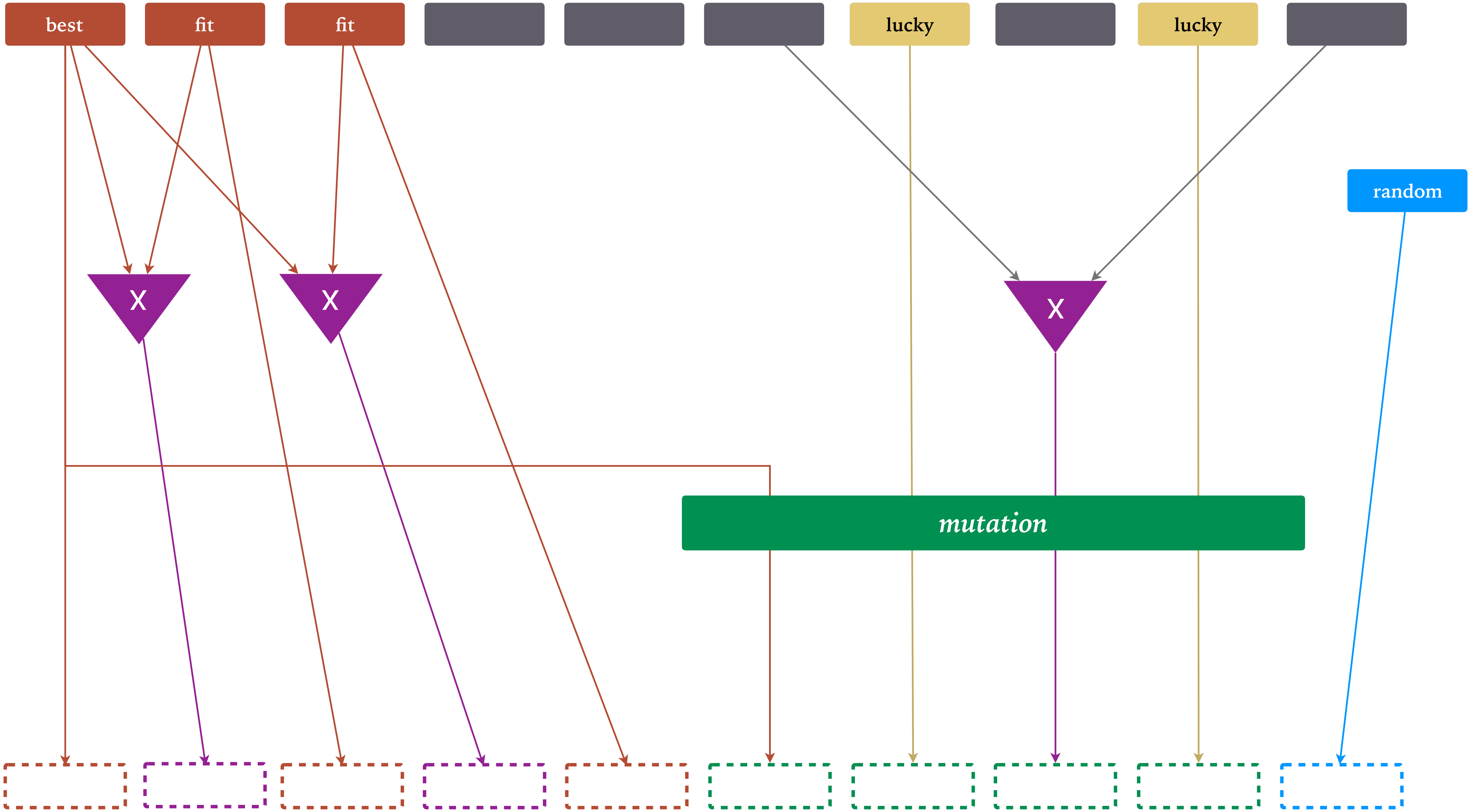
MUTATION

- *vary genomes slightly*
- real-valued, uniform mutation

- 20% chance to apply to each neuron's weight
- $w \mapsto aw + b; \quad a \sim U[0.85, 1.15], \quad b \sim U[-0.2, +0.2]$

- mutants:
    - the best one
    - the lucky ones
    - offsprings

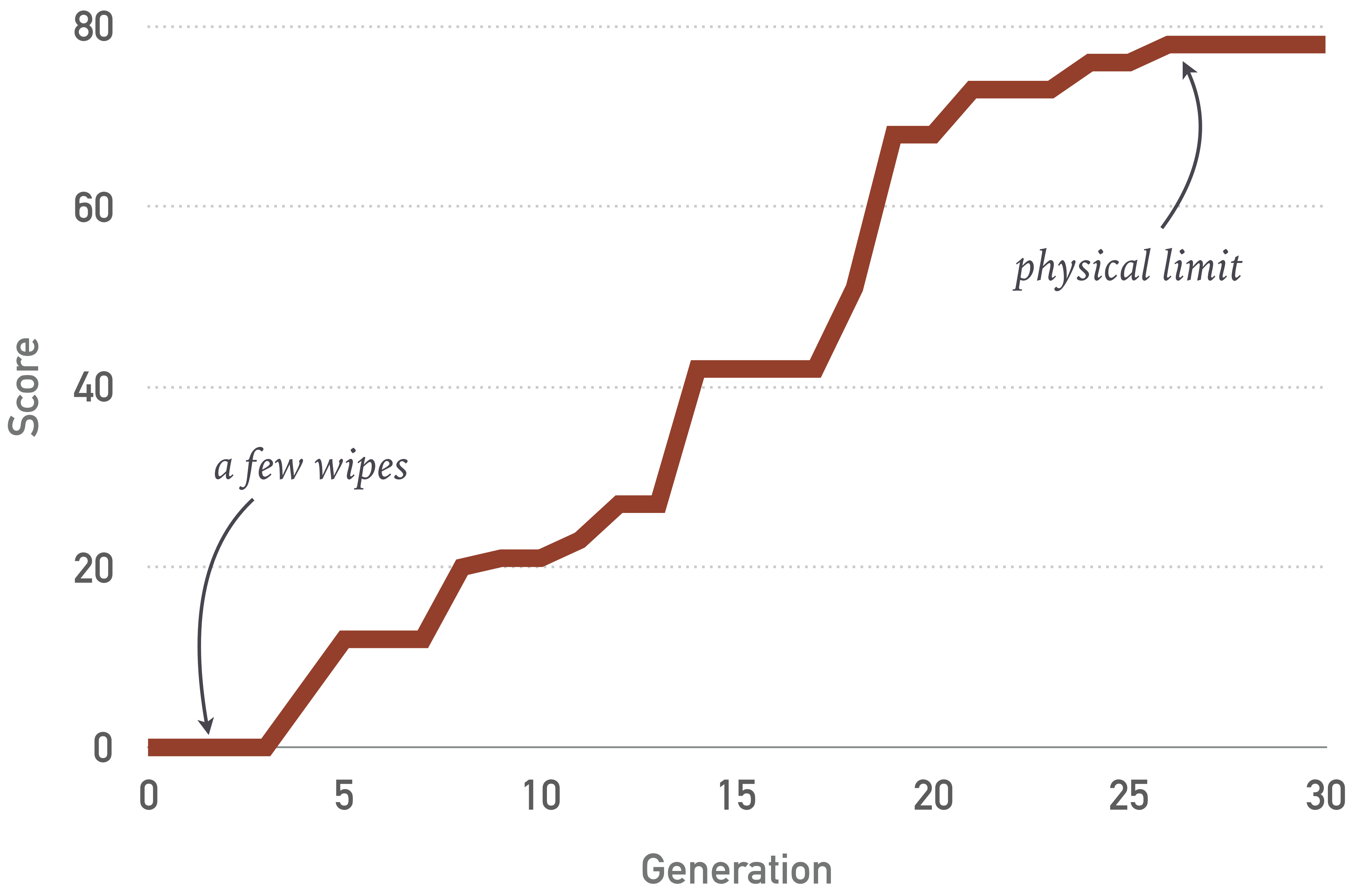
current generation



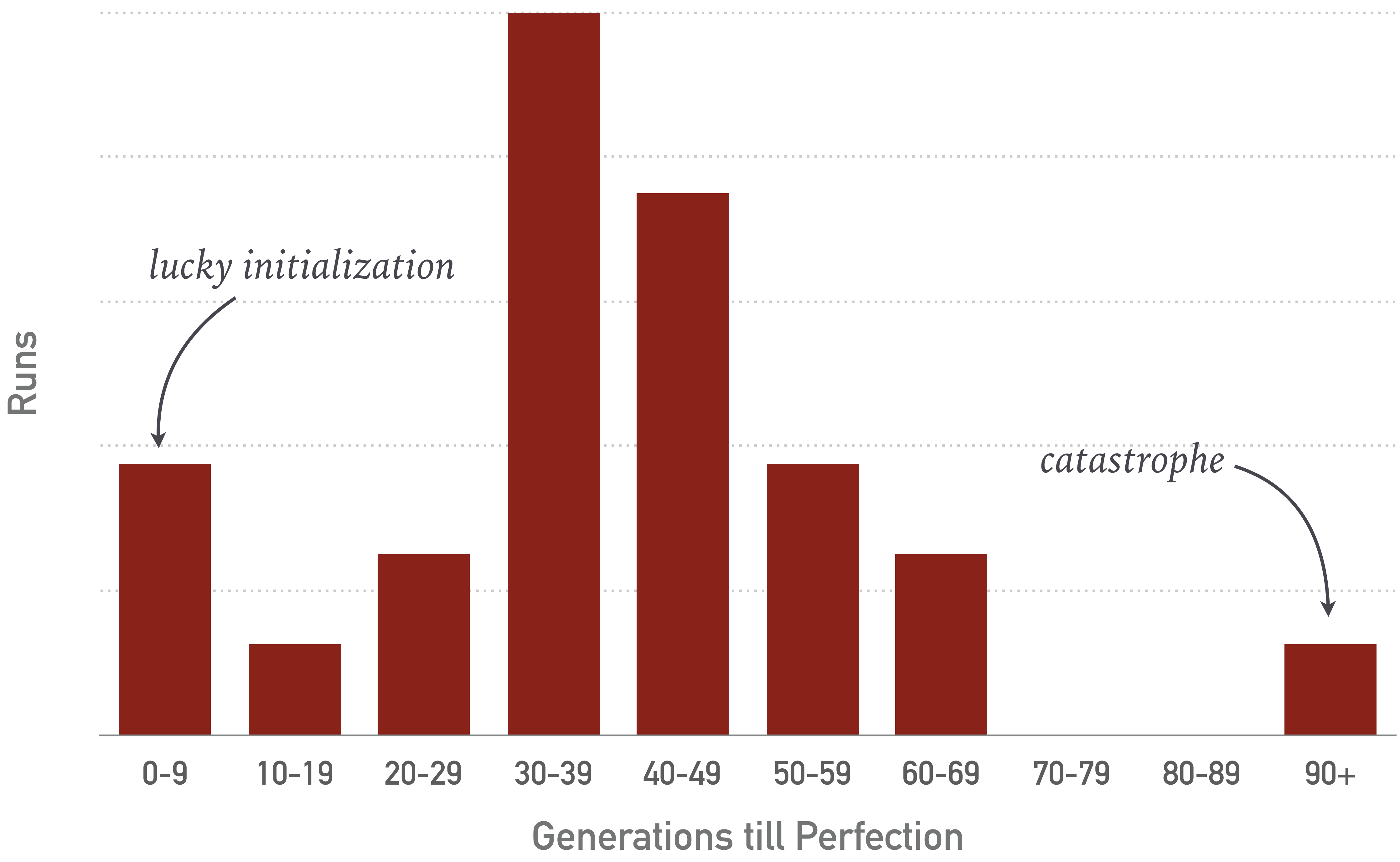
next generation

RESULTS

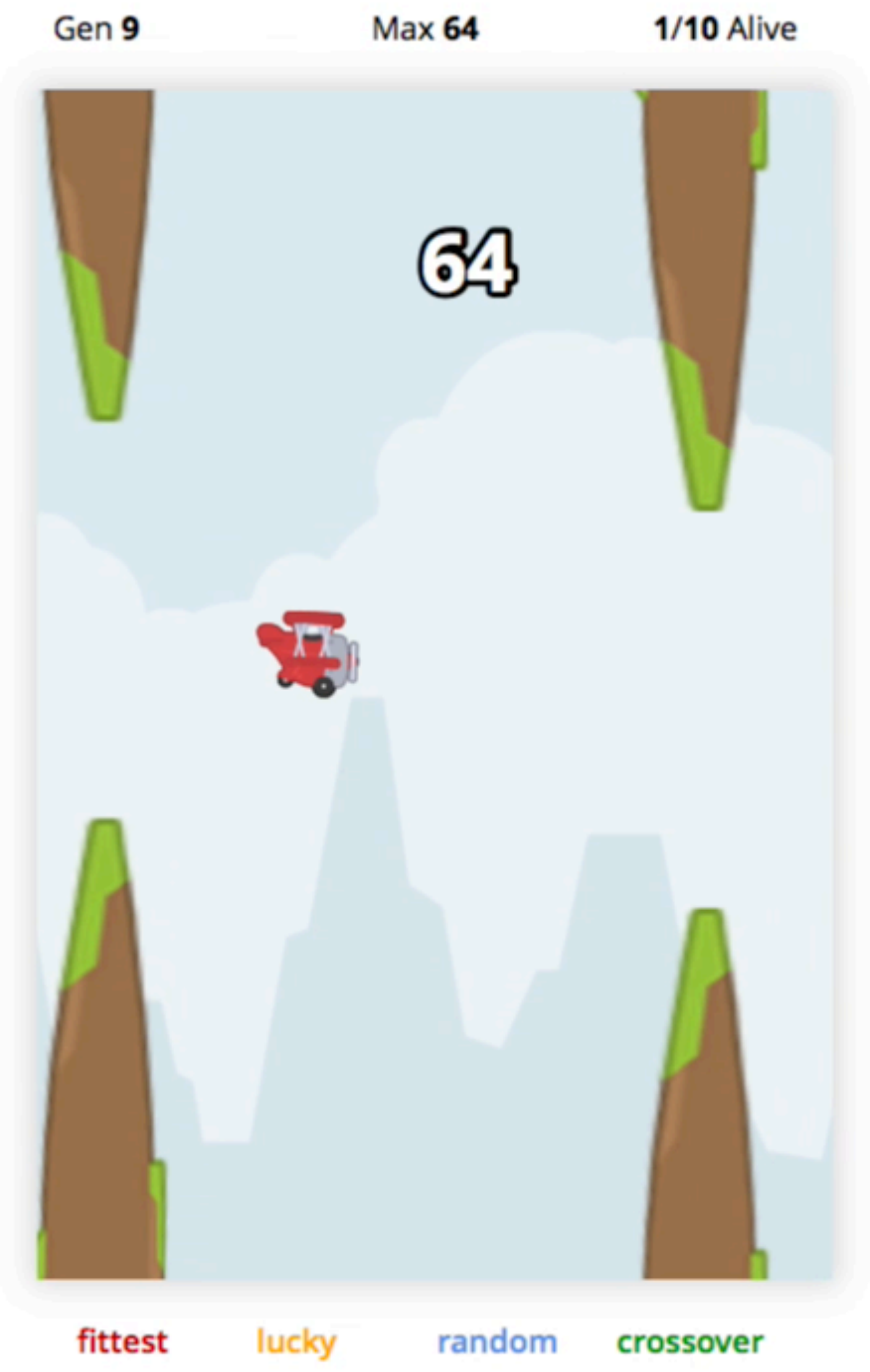
LEARNING SPEED



CONVERGENCE SPEED



A Champion is Born



TRY IT YOURSELF!

- online demo: stefann.eu/flappy-learn
- checkpointing, can save/load the bird's brain
  - repo contains parameters of the champion

- source: github.com/stefan-niculae/flappy-learn
- written in JavaScript from the ground up
  - using the Phaser game engine
  - assets from kenney.nl (paid)

THANK YOU!