School of
Computer Science

University College Dublin

Learning to Play Wolfenstein 3-D

Gary Mac Elhinney

Supervisor: Dr. Arthur Cater

What is Wolfenstein 3-D?

- First-person shooter
- Released Mid 1992
- Source code released 1995

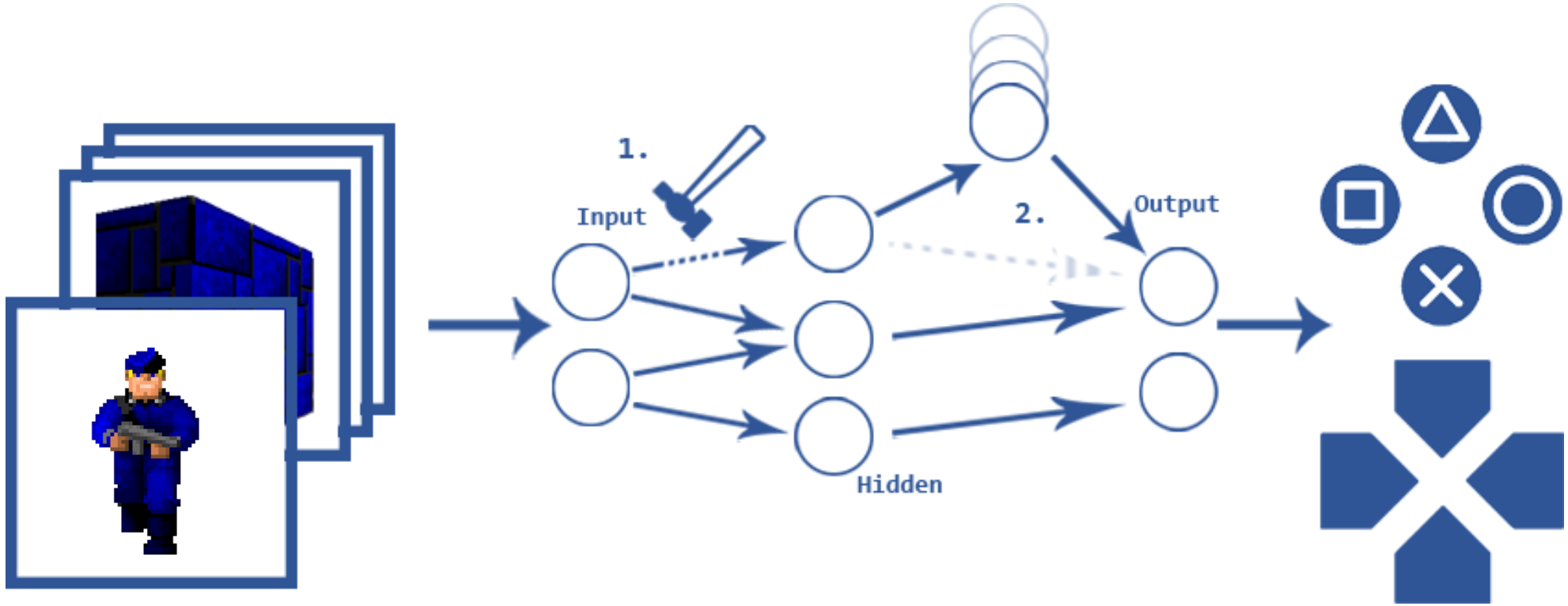
… and learning?

“the acquisition of knowledge or skills through study, experience, or being taught.”

- Google

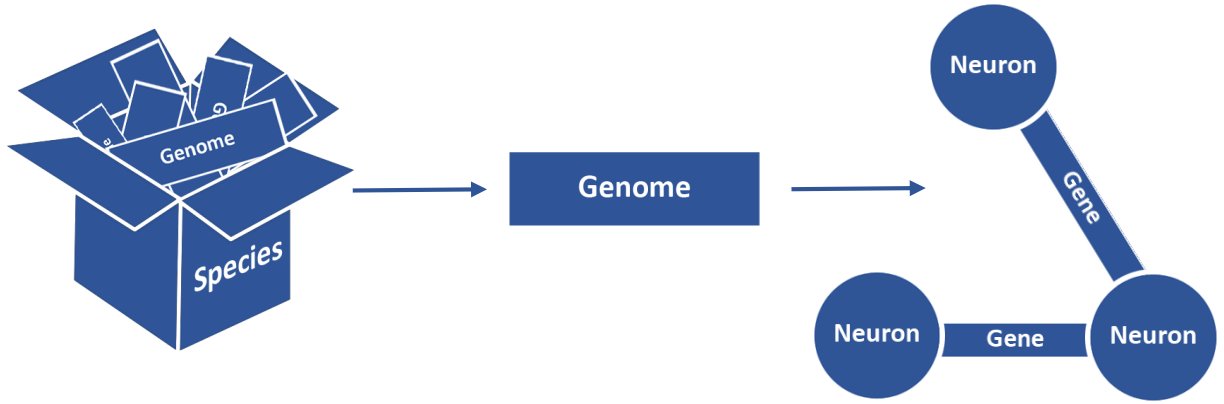
So how do we model learning with a computer and reward its progress?

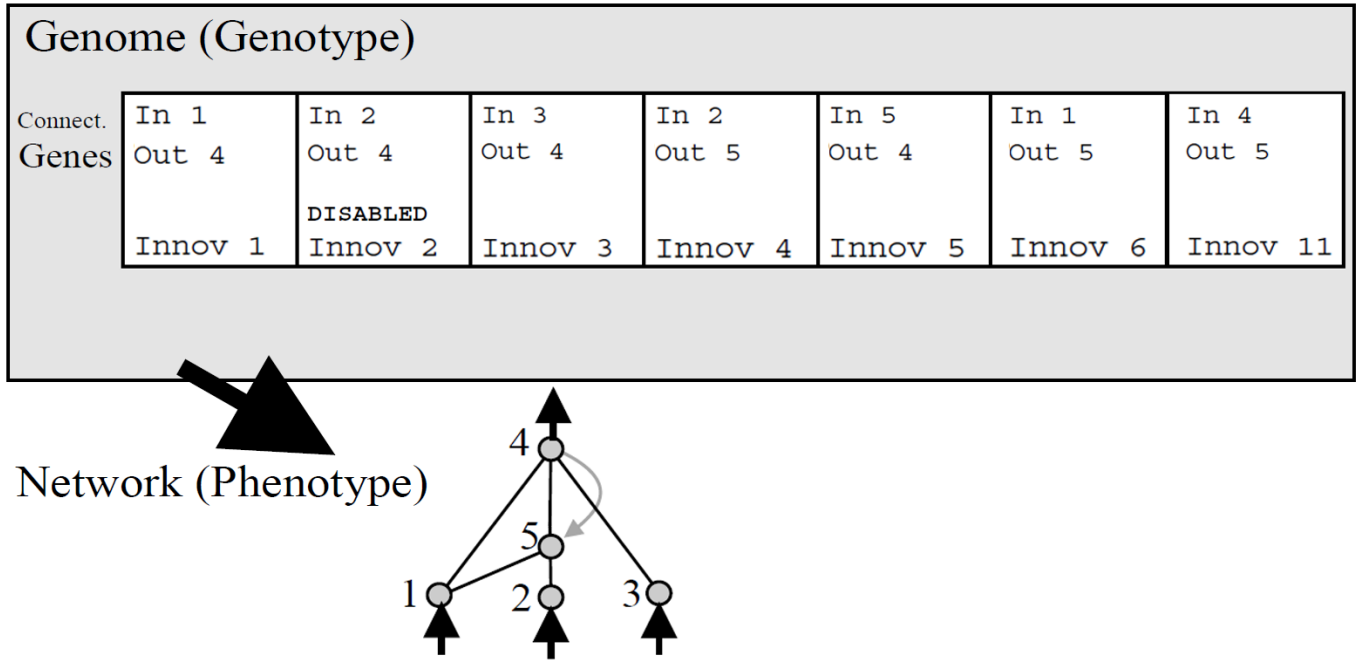
Neural Networks …



NEAT algorithm

- Genes represent links between Neurons in a network
- Genomes represent attempts, each with its own neural network
- Divides population into Species containing Genomes with similar networks
- Network similarities measured using innovation numbers
- Starts with minimal Topologies i.e. each network initially only has an input and output layer

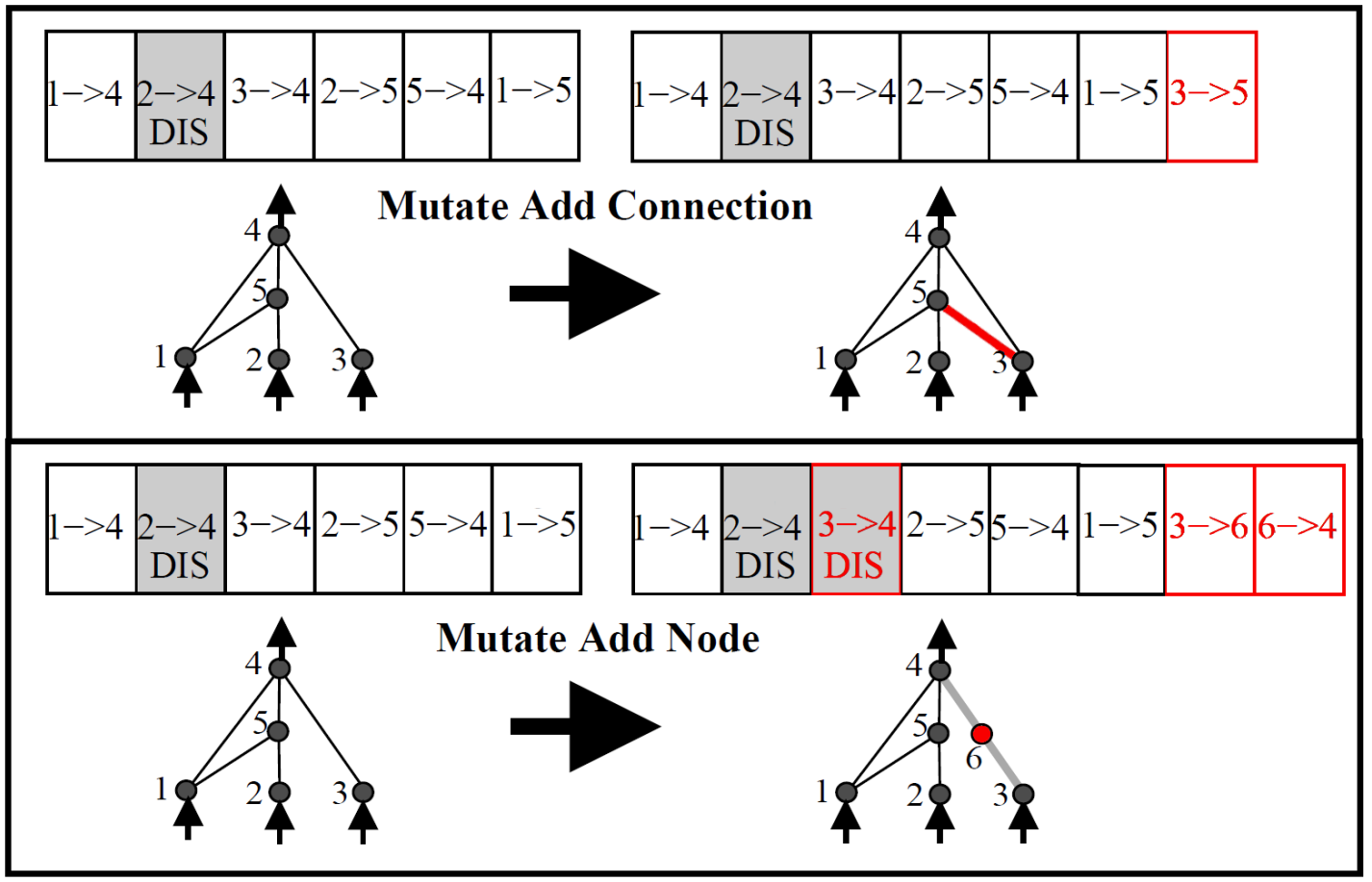




NEAT Link & Node Mutations

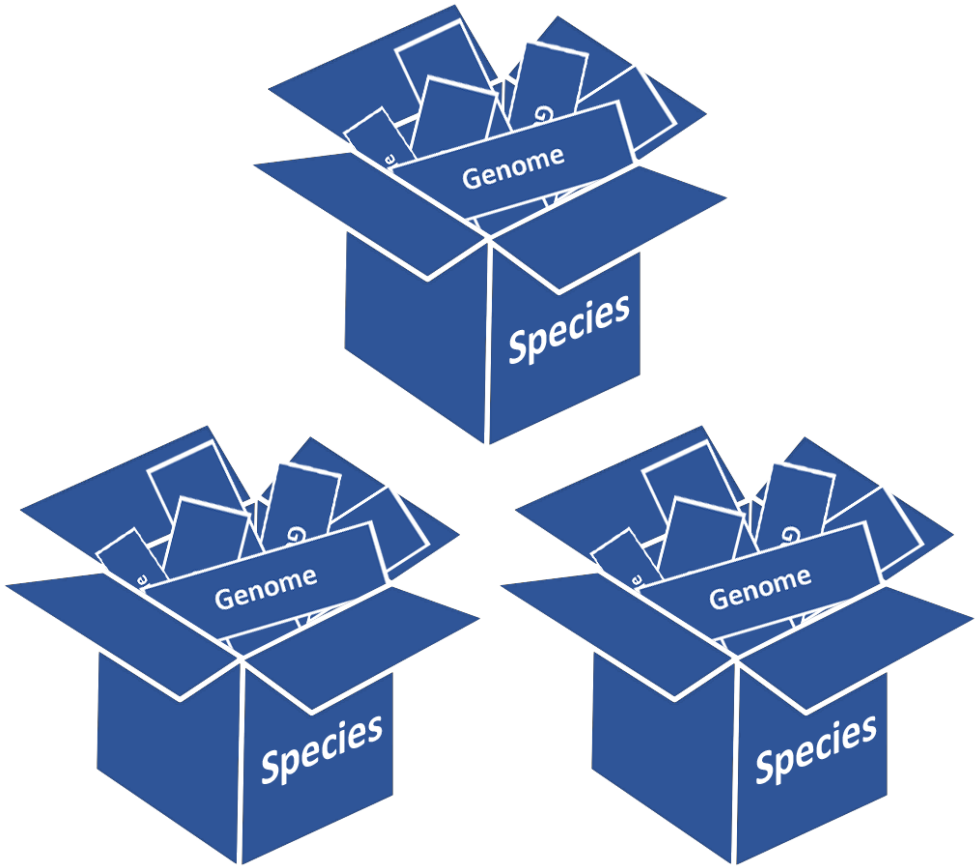
Both mutations add new Genes to Genomes

- Link mutations simply add new Links between Neurons.
- Node mutations add nodes in-between existing Neuron links.
  1. The old link is disabled
  2. In-node -> new node gets a weight of 1
  3. New node -> out-node gets weight of old link



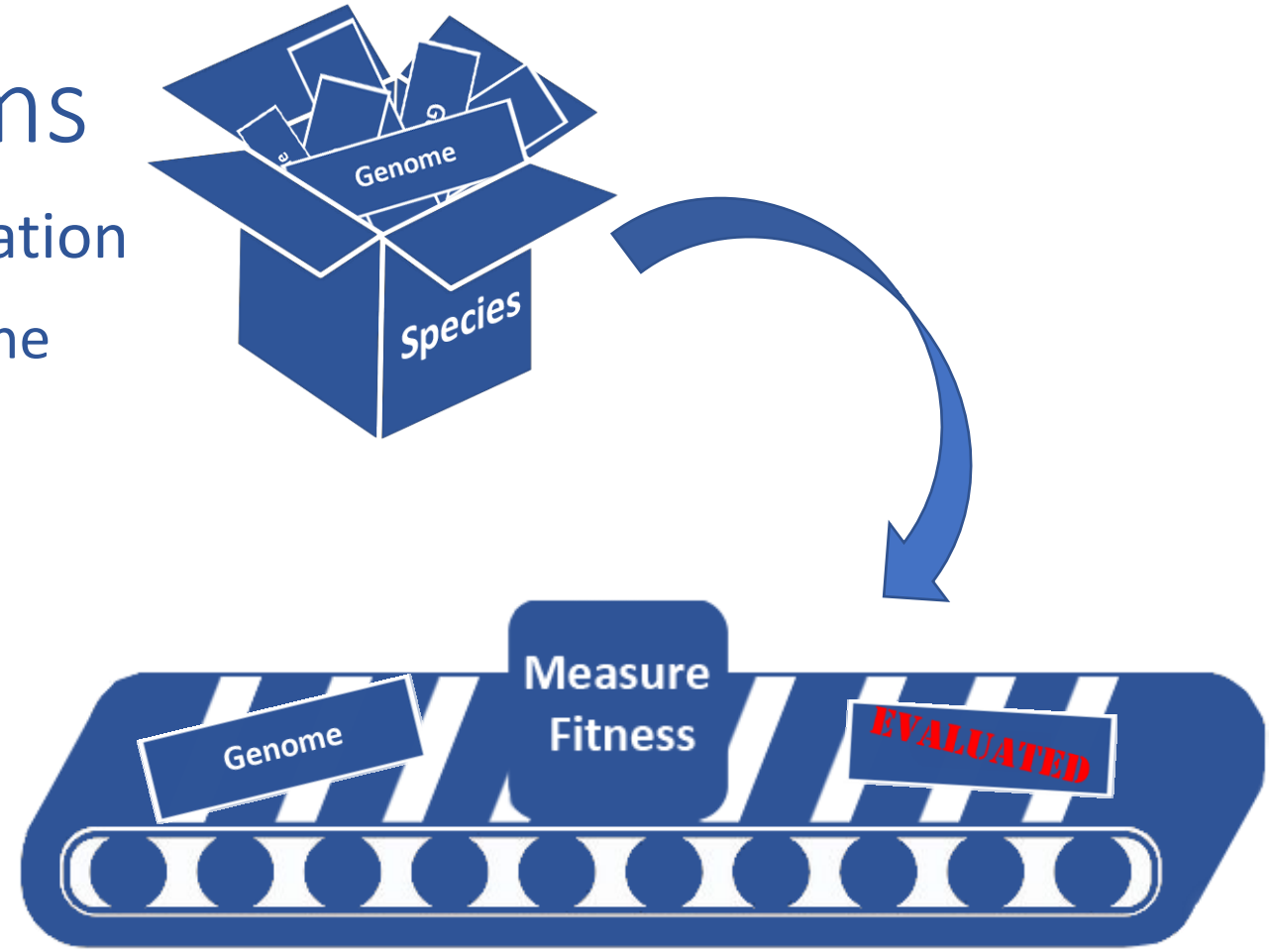
And Genetic Algorithms

1. Initialization: Set up initial population



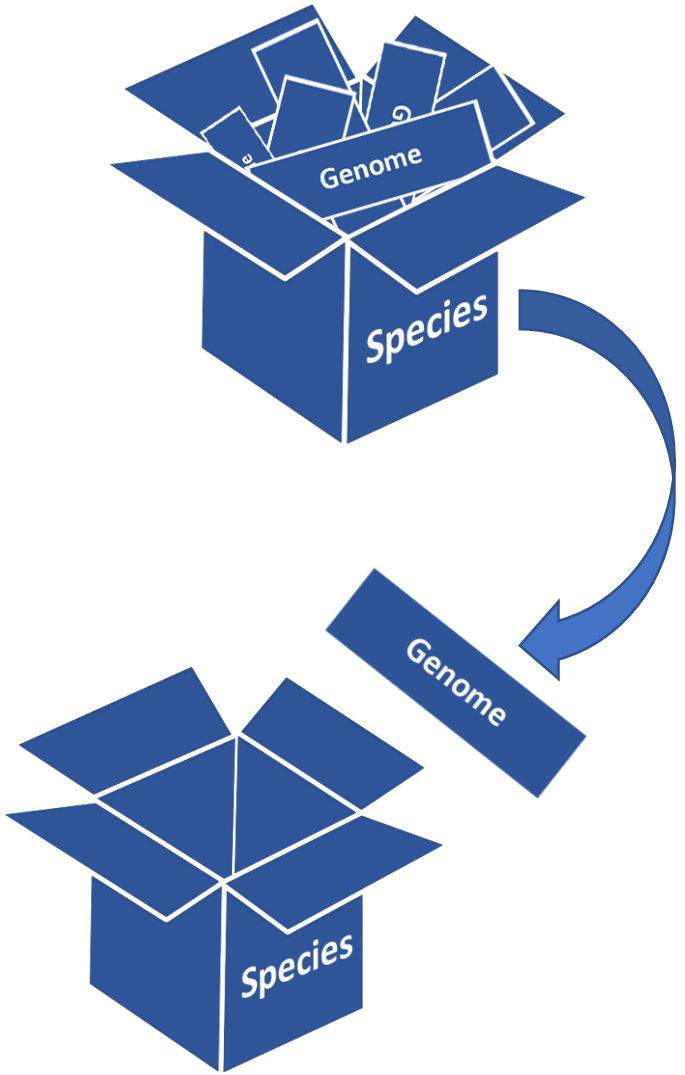
And Genetic Algorithms

1. Initialization: Set up initial population
2. Evaluation: Evaluate members of the population and assign a fitness



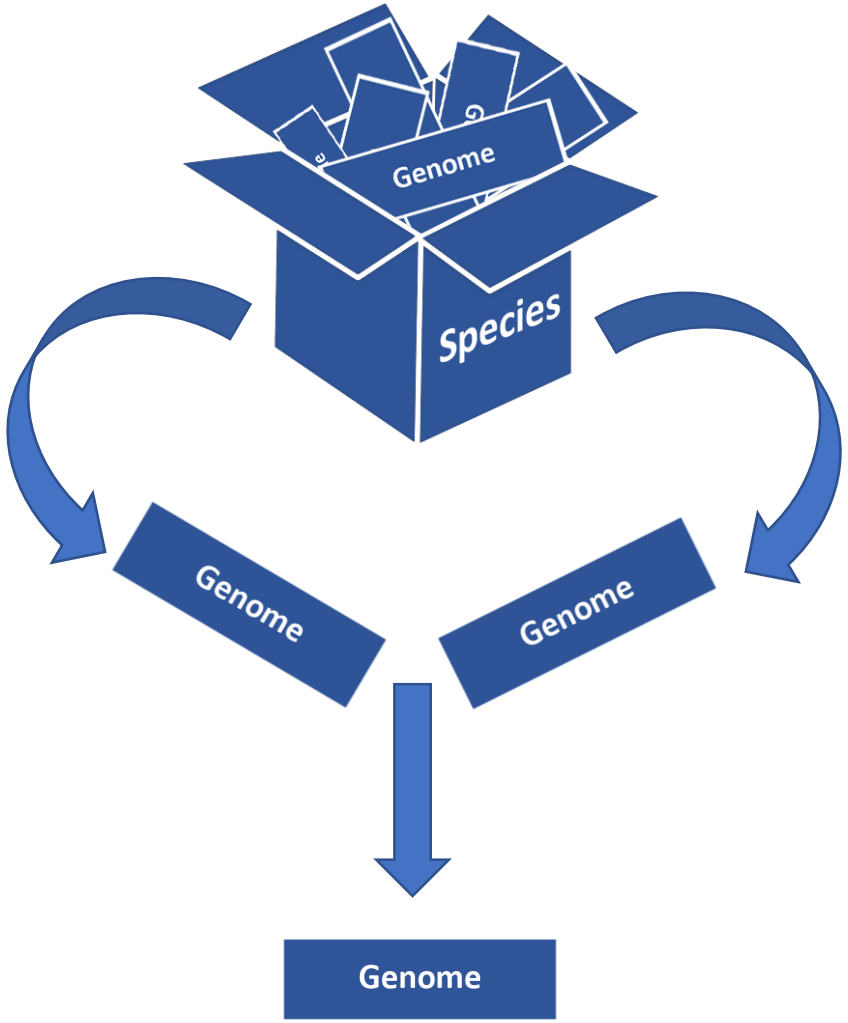
And Genetic Algorithms

1. Initialization: Set up initial population
2. Evaluation: Evaluate members of the population and assign a fitness
3. Selection: Improve population by selecting best few members for new generations



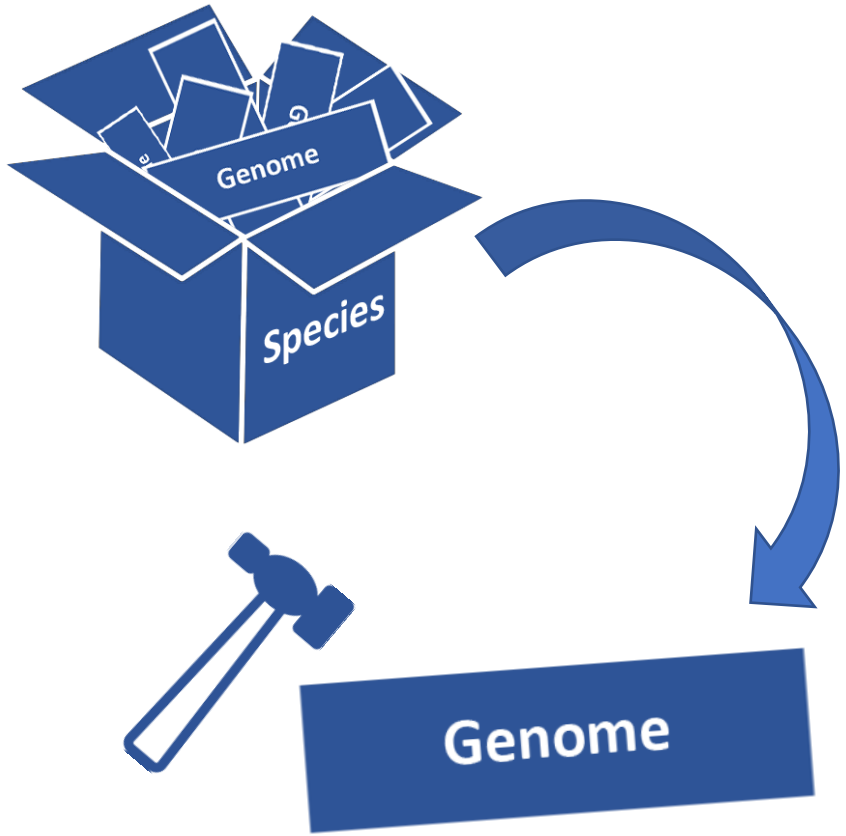
And Genetic Algorithms

1. Initialization: Set up initial population
2. Evaluation: Evaluate members of the population and assign a fitness
3. Selection: Improve population by selecting best few members for new generations
4. Crossover: Create new members by combining aspects of two different members



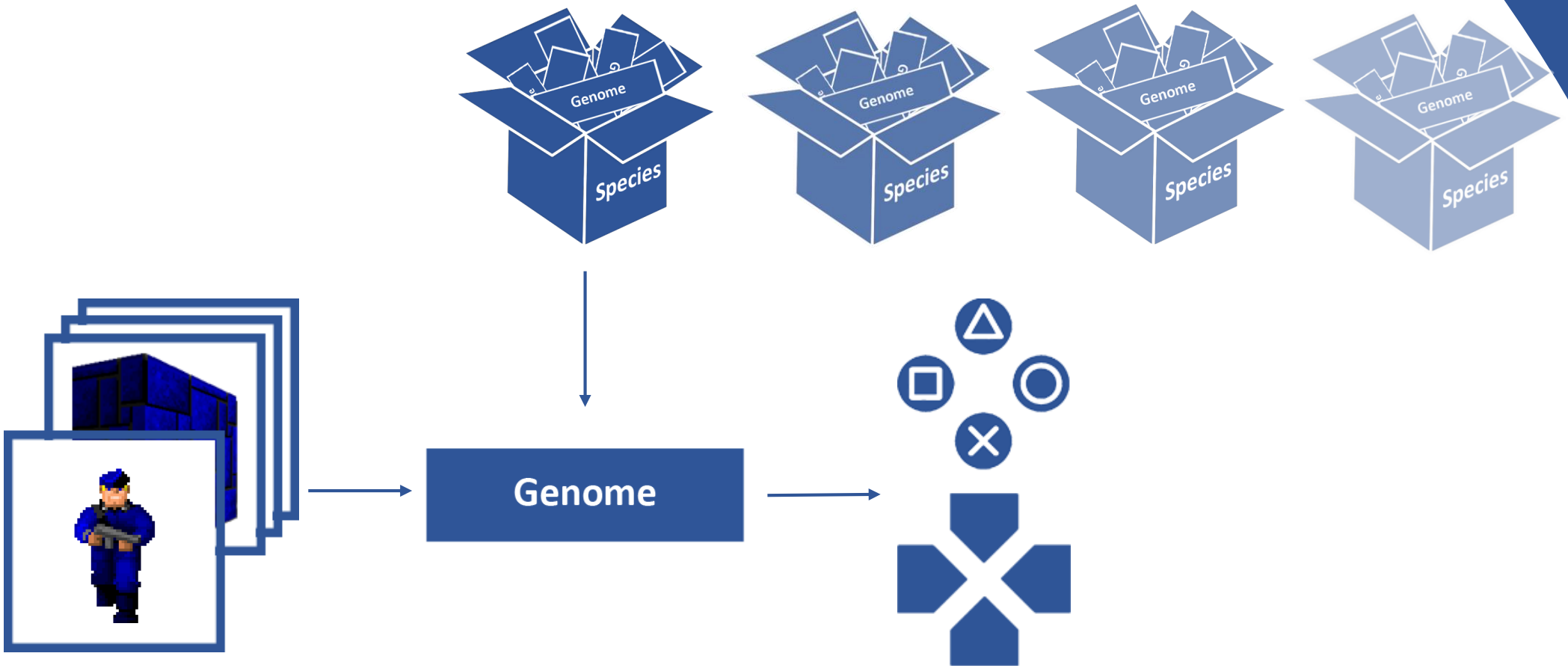
And Genetic Algorithms

1. Initialization: Set up initial population
2. Evaluation: Evaluate members of the population and assign a fitness
3. Selection: Improve population by selecting best few members for new generations
4. Crossover: Create new members by combining aspects of two different members
5. Mutation: Make small, random changes to individuals to add genetic diversity
6. Repeat!



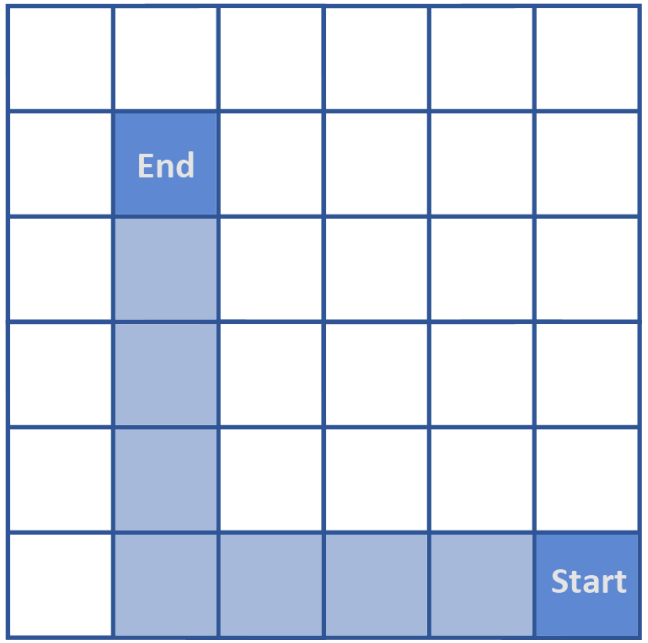
NEATDoop

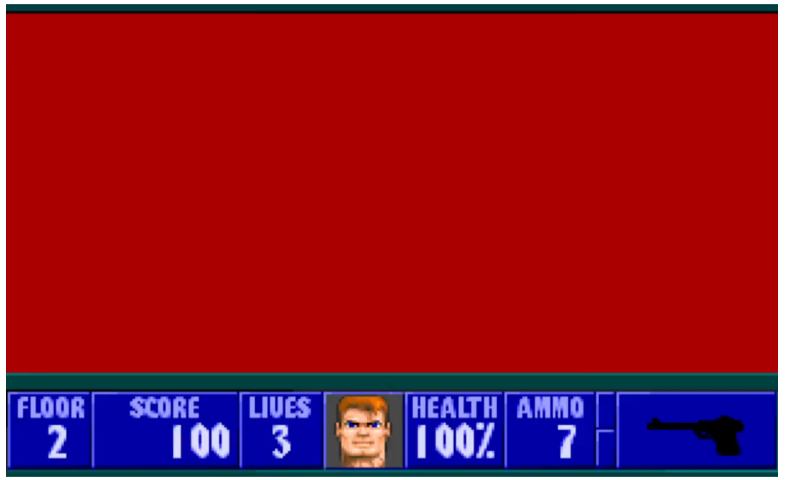
- Developing Object Oriented Program (Doop).



Learning with NEATDoop

- Population: 250
- Network inputs: 275
- Network outputs: 11






```
fitness = (MAX_DISTANCE - distFromEnd) * 10 +
          (distFromSpawn * distFromSpawn * 10) +
          numKills * 50 +
          numPickups * 100 +
          lvlDoneReward;
```

School of
Computer Science

University College Dublin

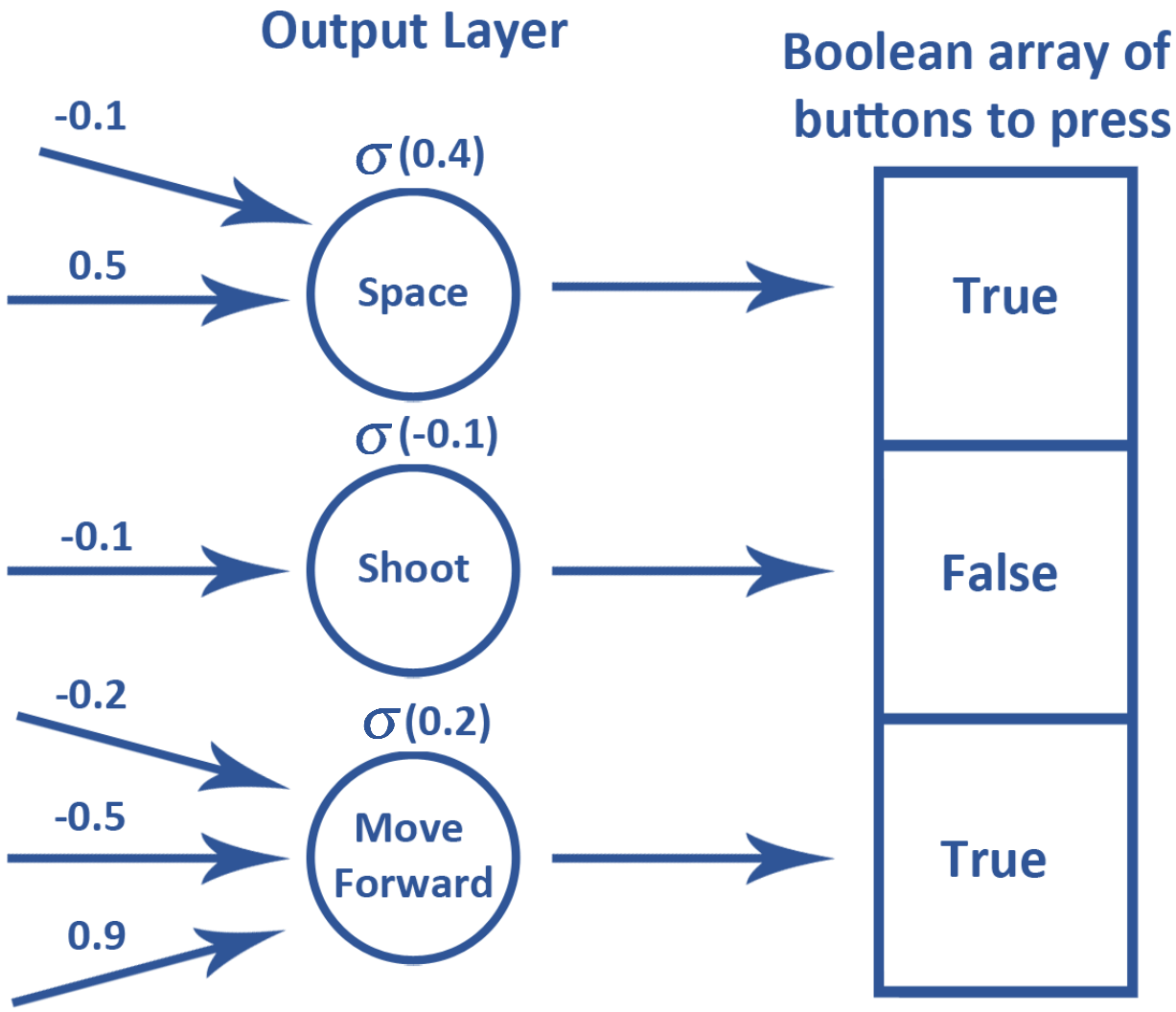
Questions?

Output Node Activation

- If sum of inputs to output node is > 0, the nodes corresponding button is activated
- Modified Sigmoidal transfer function

$$\sigma = \frac{2}{1+e^{-4.9x}} - 1$$

- Steepened sigmoid allows more fine tuning at extreme activations



NEAT Genome Crossover

- Matching Genes are inherited randomly from either parent
- Disjoint & Excess Genes are inherited from the more fit parent unless both parents have the same fitness
- In the example given, both parents have identical fitness