

# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126  
Pranav Kasela, 846965  
Federico Moiraghi, 799735



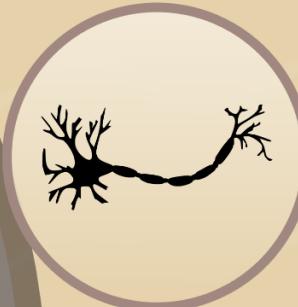
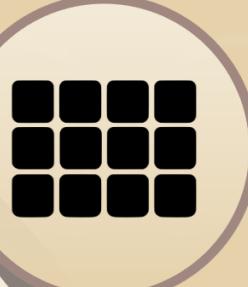
# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126  
Pranav Kasela, 846965  
Federico Moiraghi, 799735



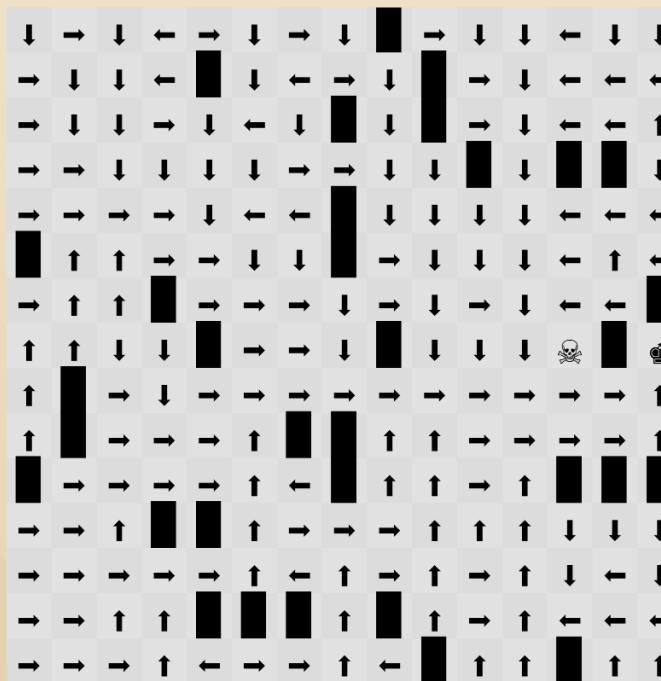
# *Smaug's Treasure*

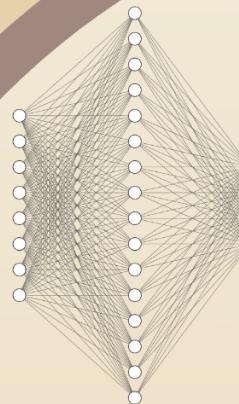
Will Bilbo, our Agent and Hero, manage to retrieve  
the treasure guarded by the sleeping Smaug?



# Q-Table

The simplest approach: Bilbo only knows where he is



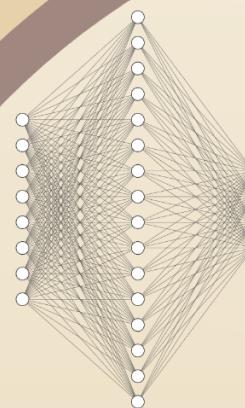


## Deep Q-Learning

Deep learning is used to reach a better generalization of the problem.

During the first game, Bilbo woke up the dragon: now it is angry and it has destroyed everything as it started to rage randomly.

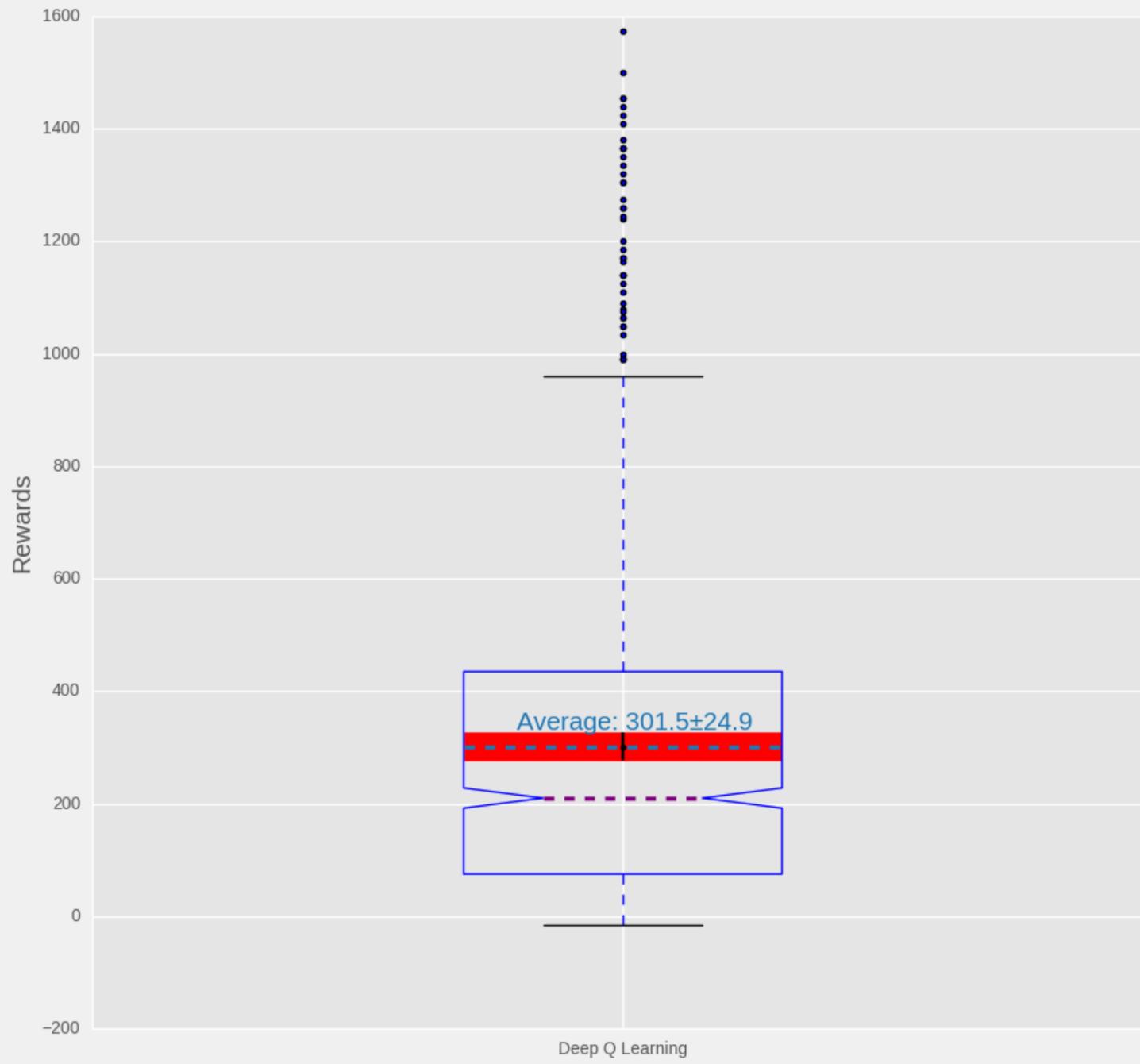
But Bilbo is as greedy as always and he wants to collect all the treasure he can.

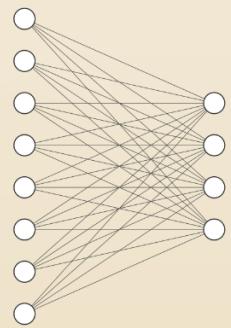


## Deep Q-Learning

The experience replay model is implemented, which uses old experience (data) to train the NN.

- 64 experiences (oversampled):
  - 32 with low reward
  - 32 with high reward (death or treasure)
- Convergence Time: ~69000s (~19h)
  - 20,000 episodes with 150 epochs

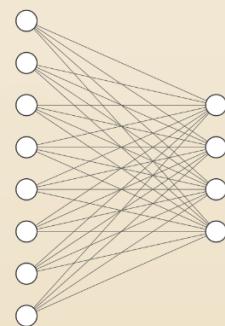




## DeepQL + GA

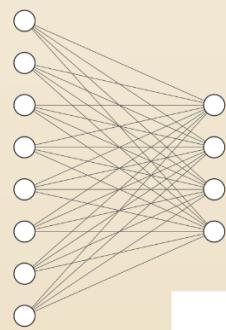
Bilbo learns he by himself is too slow to learn how to beat the Dragon, so he creates 150 copies of himself with different memories and keeps only the best 15 from each generation, recreating the population, for the next generation, from these 15.

Genetic Algorithm is used to find the optimal weights of the Neural Network: Bilbo knows relative position of the treasure and dragon and the edges.

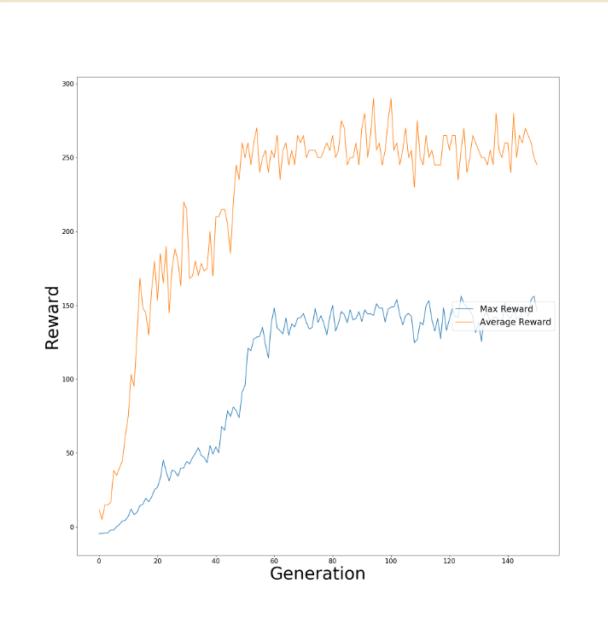


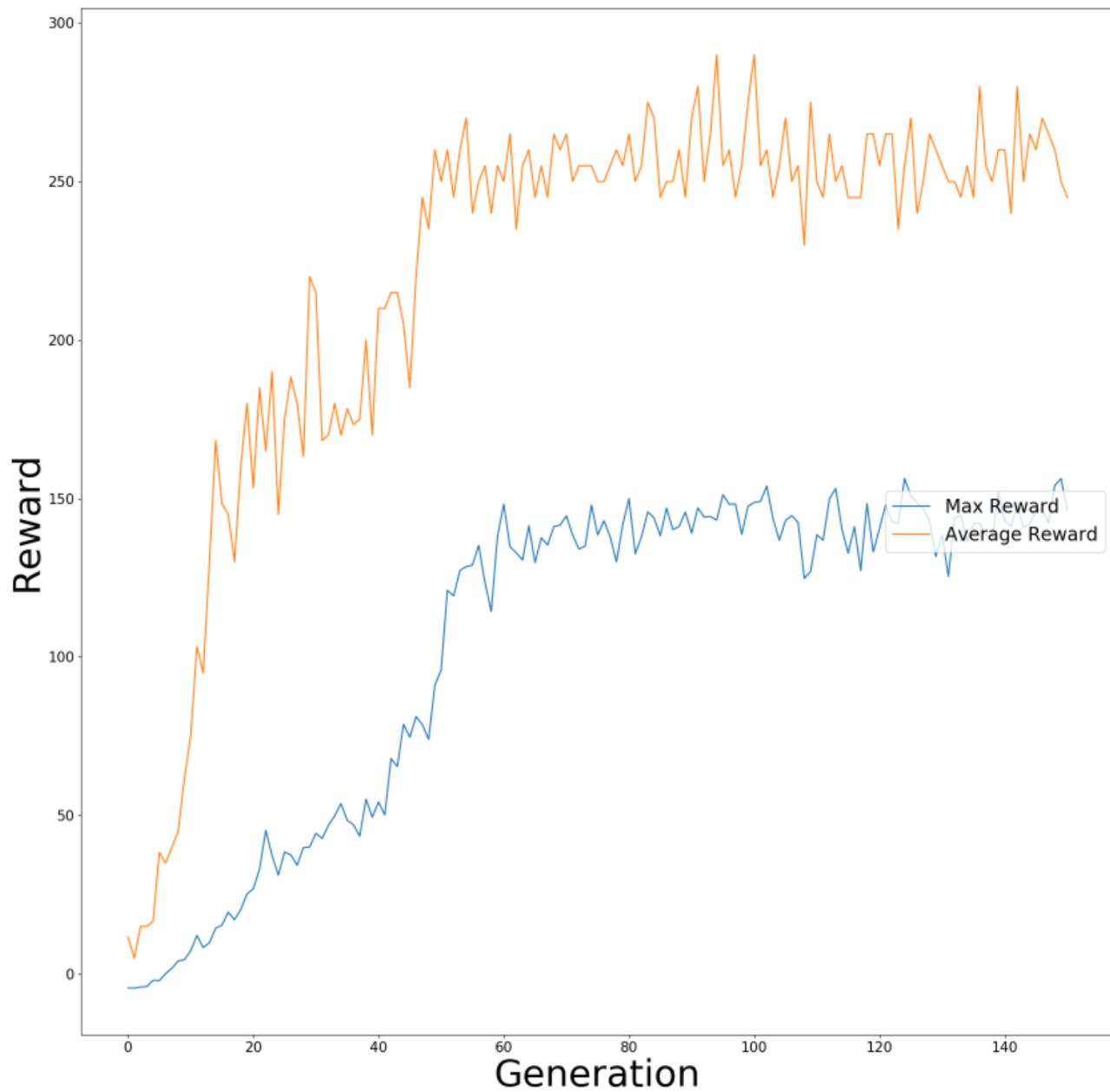
## DeepQL + GA

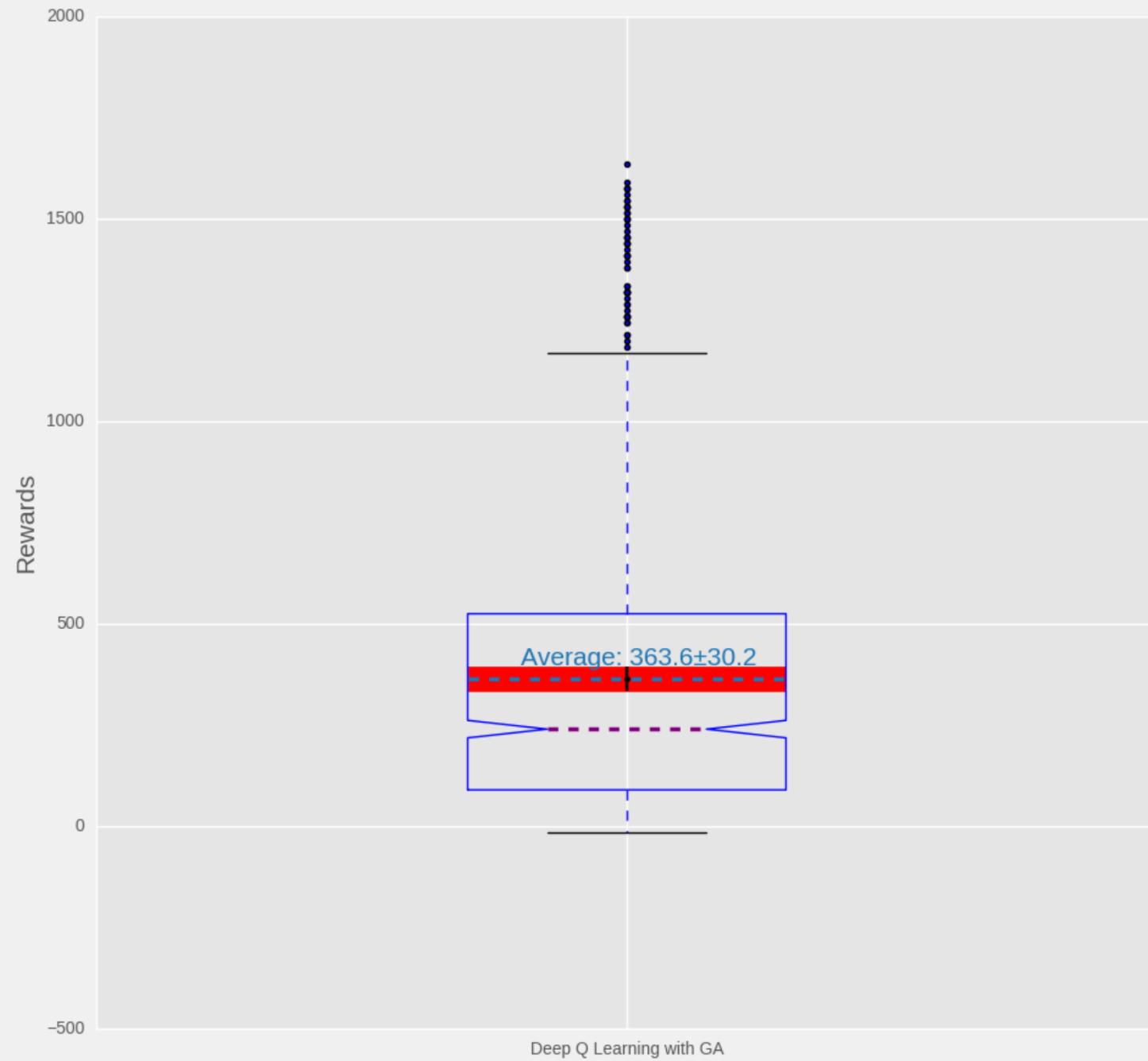
- Initial Population: 150
- Mutation Rate: 5%
- Elitism: 10%
- Mutation: multiplying by a random number between -2 and 2.
- Fitness: Average Score of 3 games of 150 epochs
- Total Generations: 150
- Convergence Time: ~10000 seconds (~3 hours)
  - 6-7x Faster than experience replay method



# DeepQL + GA







# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735



# Methodological Approaches for Multi-agent RL Games

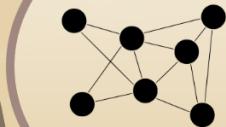
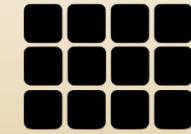
Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735



# *Dedalus Hunting*



Five brave warriors fight to death in a labyrinth.  
Only the strongest survives.



# **Q-Table**

The easiest approach: each combination of Agent's and enemies' position is a state.

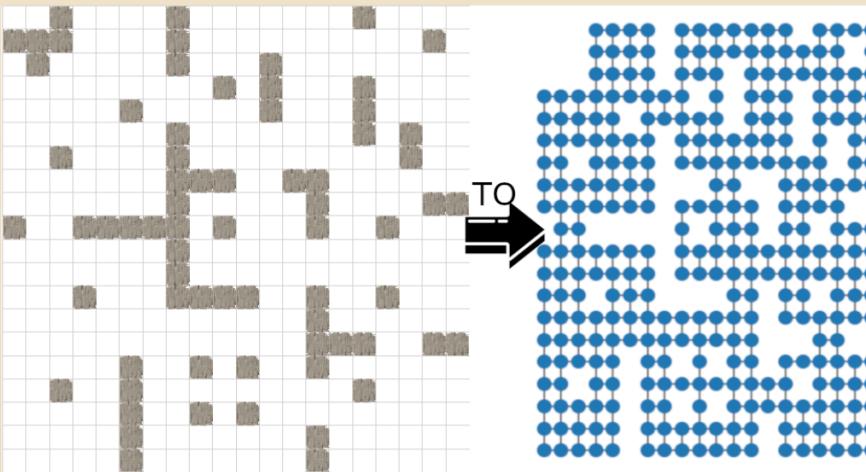
The number of states grows exponentially with the number of agents.

## **Q-Table + Graph**

Agents know exactly the shortest path (just the direction) to reach the closest enemy, and know if they can attack; but they do not know where they are.

The number of states is fixed, and the complexity of the shortest-path algorithm grows linearly with the number of Agents.

# Q-Table + Graph



# Q-Table + Graph

Attacking Reward:

- +10 for each enemy damaged
- 20 for each enemy killed
- -2 for useless attack

Moving reward:

- 0 if getting closer to the enemy
- -2 if moving away from the enemy
- -5 for hitting a wall or the border

# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735



# Methodological Approaches for Multi-agent RL Games

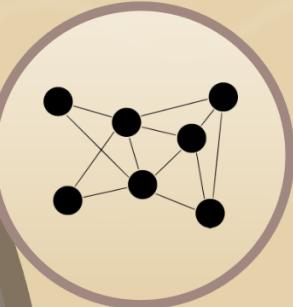
Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735



# *Beowulf*



A dungeon, two Agents and one awful Dragon.  
The hunt is open!



# **Q-Table with a Graph**

Agents do not know who is the enemy and who is their mate.  
They only know how to reach them (the first step in a graph)  
and if they are close. They know if their health is low or not  
and they can see up to 3 nodes in a graph.

Complexity grows linearly with number of Agents and world size.

# Q-Table with a Graph

## Attacking Reward:

- reward + (health - 3)/(health + 1)\*reward if ≠ 0
  - reward = +5 for each enemy attacked
  - reward = +25 for each enemy killed
  - reward = -10 for each friend attacked
- -5 if modified reward= 0 or if no damage done

## Moving Reward:

- +2 if getting near the enemy
- -2 if getting far from the enemy
- -5 if tries to move against a wall or the border

# Q-Table with a Graph

State:

- binary health:
  - 1 if self.health > 3
  - 0 otherwise
- direction and distance to the other players:
  - direction: 0(up), 1(down), 2(right), 3(left)
  - distance:
    - 0, 1, 2 if distance of the shortest path  
= 0, 1, 2 respectively
    - 3 if distance of the shortest path > 2

# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735



# Methodological Approaches for Multi-agent RL Games

Riccardo Cervero, 794126

Pranav Kasela, 846965

Federico Moiraghi, 799735

