

# Building a Battlesnake with Different Approaches

Farid Anjidani (V00971584)  
Chris Chiu (V00973917)  
Zixuan Deng (V00971633)

## Abstract

We explored a multitude of machine learning techniques to compete in the game of Battlesnake, which is a multiplayer version of the traditional game of Snake. We evaluated the different performances by pitting each model into the same arena and discovered that regarding our available computation resources, simplicity was often best. When it came to playing Snake, the snake that was coded with conditional statements often outperformed and outlived the snake that was trained with baseline heuristics in a reinforcement learning setting. Although these results were surprising, we concluded that more work still needs to be done and perhaps adjusting the heuristic values a bit more would lead to different results.

## References

- [1] Chung, J., Luo, A., Raffin, X., & Perry, S. (2020). Battlesnake Challenge: A Multi-agent Reinforcement Learning Playground with Human-in-the-loop. arXiv preprint arXiv:2007.10504.
- [2] Altersaddle. (n.d.). Altersaddle/untimely-neglected-wearable: A simple Battlesnake written in Python. Retrieved from <https://github.com/altersaddle/untimely-neglected-wearable>
- [3] "ML: Monte Carlo Tree Search (MCTS)." GeeksforGeeks, 14 Jan. 2019, [www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/](http://www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/).
- [4] Fool-Yang. "Fool-Yang/AlphaSnake-Zero: A Reinforcement Learning AI That Plays the Synchronous Strategy Game Battlesnake Based on the ALPHAGO ZERO'S Algorithms." GitHub, [github.com/Fool-Yang/AlphaSnake-Zero](https://github.com/Fool-Yang/AlphaSnake-Zero).

## Methodology

### Smart-moving Strategy Snake using Conditional Statements

In this approach, there are many rules that let the snake avoid edges of the map, collisions with other snakes and itself, and paths that are closed to other snakes moving paths, spaces that are dangerous. And it would also consider some features such as the movements of other snakes in collision range, find winning movement when other snakes have only one available movement, and find available food while moving

### Reinforcement Learning using Amazon Sagemaker

Reinforcement learning develops strategies for sequential decision-making problems. In reinforcement learning, we have two main parts: agent and environment. The environment gives the agent its state and based on the state the agent makes an action and environment based on that action gives the agent a reward. The agent goal is to make best actions based on the state it receives, and to maximize the expected cumulative reward.

training metrics	Episode 33
best snake episode len max	351
Forbidden move max	4
Forbidden move mean	0.69
Forbidden move min	0
Killed another snake max	4
Killed another snake mean	0.54
Killed another snake min	0
Snake hit body max	5
Snake hit body mean	1.32
Snake hit body min	0
Snake hit wall max	2
Snake hit wall mean	0.12
Snake hit wall min	0
Snake was eaten max	4
Snake was eaten mean	1.12

Fig 1: Training Metrics

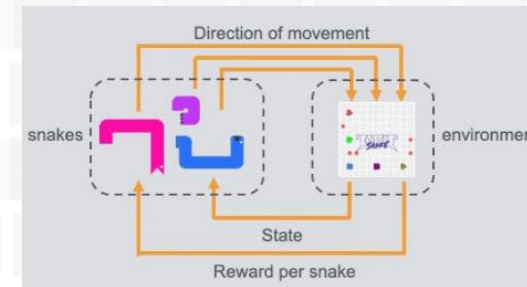


Fig 2: Reinforcement Learning Model

### AlphaZero Snake Using Monte Carlo Tree Search

The input state is  $21 \times 21 \times 3$  as the snakes' head are fixed in the first channel. The walls and snakes' bodies are in the second channel. The foods are in the third channel. After each training the new network compete against the old ones and the result will be updated if it performs better than previous ones

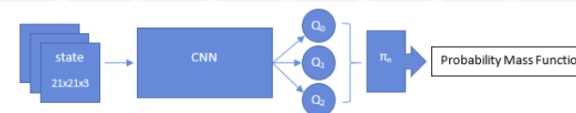
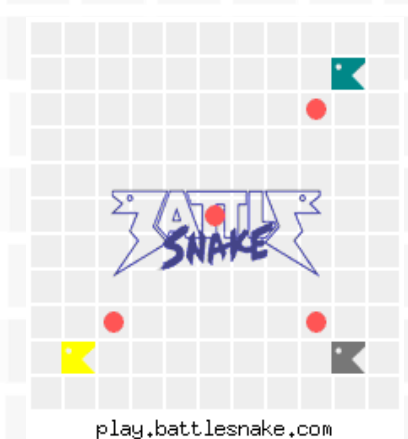
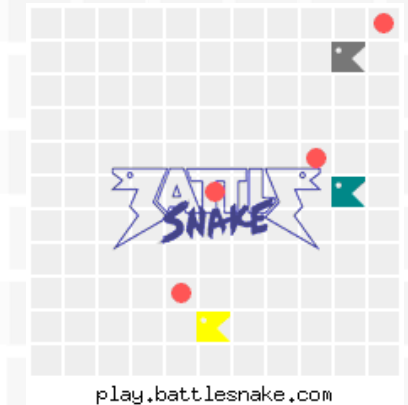





Fig 3: AlphaZero Model

## Game Results



-  Legless Lizard – Top 100 Snake
-  Snaky97 – Conditional Statements
-  Snaky97AWS – RL with Base Heuristics