

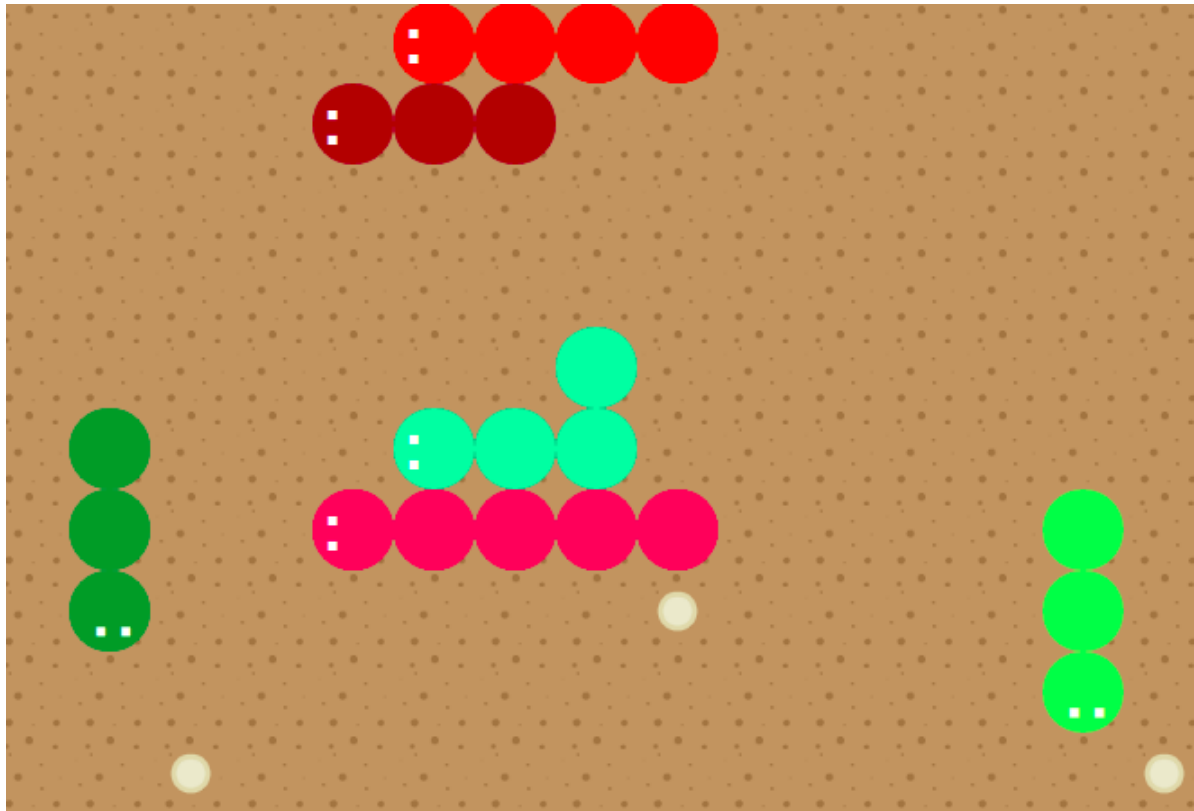
# Project: Snakes 3v3

CS410: Artificial Intelligence

Shanghai Jiao Tong University, Fall 2021

Due Time: 2021.12.29 23:59 (for presentation slides) / 2022.01.02 23:59 (for all materials)

## Introduction



In this project you will play Snakes 3v3, which is a multi-agent version of the traditional Snake game. The game is played by controlling the direction of the snakes' head to eat beans, thus making the snakes longer and longer.

*Important:* This is an alpha version, and hence some details may be slightly changed in the future.

## Rules

1. There are **two** parties in this game, and each player manipulates **three snakes** in a grid of  $n \times m$ .  $n$  is the vertical height and  $m$  is the horizontal width.
2. A snake is a finite, non-repeated and sequential sequence composed of a series of coordinates. The adjacent coordinates in the sequence are adjacent, that is, the x-axis or y-axis coordinates of the two coordinates are the same, and the first coordinate in the sequence represents the snake head. The player can only control the snake by controlling the **head's orientation** (up, down, left and right).
3. The snake moves forward at a **constant speed** (forwarding means inserting the head of the sequence into the next coordinate in the direction in which the head of the snake points, and deleting the coordinate at the end of the sequence). The initial position of the snake is a random position in the grid, and the initial length is **3**. The length of each bean eaten increases by **1**. In the game, the number of beans remains constant (5), that is, if they are eaten, the same number of beans will be randomly generated.

4. Snake's heads in the body of its own (the sequence is repeated), and the body of other snakes (with the same coordinates as other sequences) are judged to be **dead**. When an illegal direction is entered (such as moving to the left, you cannot choose to the right), a legal direction is randomly selected. Unlike traditional games, when the snake head crosses the boundary, it can cross to a symmetrical position, which is not counted as death. When the snake dies, it can be reborn according to the initial settings.
5. The game ends when the specified number of steps is reached. After the game is over, the party with the **larger sum of the length of the snakes wins**.
6. Observation is a dictionary with key from 1 to 7 and `board_width`, `board_height`, `last_direction`, `controlled_snake_index`. Here, 1 represents beans, while 2 to 7 represent snakes. The value of the dictionary is a list with  $[h, w]$  coordinates as elements, representing positions of beans or snakes.  $h$  represents the vertical distance from the origin in the upper left corner,  $w$  represents the horizontal width from the origin. Among the values corresponding to 2 to 7, the list elements from left to right represent the position of the head to the tail of the corresponding snake; The value of `board_width` is the width of the board; The value of `board_height` is the height of the board; The value of `last_direction` are directions of each snake in the last step; The value of `controlled_snake_index` is the controlled snake index.
7. The `action_space` is a list with length of `n_action_dim`. Here, `n_action_dim=1` and every element in it is a Discrete object in Gym ([Discrete Link](#)), like `[Discrete(4)]`.
8. For more details, please refer to the source code below.

## Source

---

Clone or download the repository [Competition 3v3snakes](#) to get started.

## Evaluation Guide

---

In this project, the final score will be determined by:

- **Performance:** We will periodically release some opponents for you to evaluate the performance and improve your algorithm. However, **only the final submission counts towards your grades**, which will be competed with the others and ranked.
- **Report:** You need to provide a technical report to describe your algorithm design. Use the [template](#) and limit your submission **eight content pages** (including all figures and tables). Additional pages containing references are allowed.
- **Presentation:** In the last class, each group will give a 5-min presentation (+extra 1 minutes for Q&A).

We will announce the platform for evaluation once it is all set. To evaluate your algorithm in the platform, you only need to provide one single file named `submission.py` containing `my_controller()`. Here is an [example](#).

*Important:* Late submissions are **NOT** accepted.