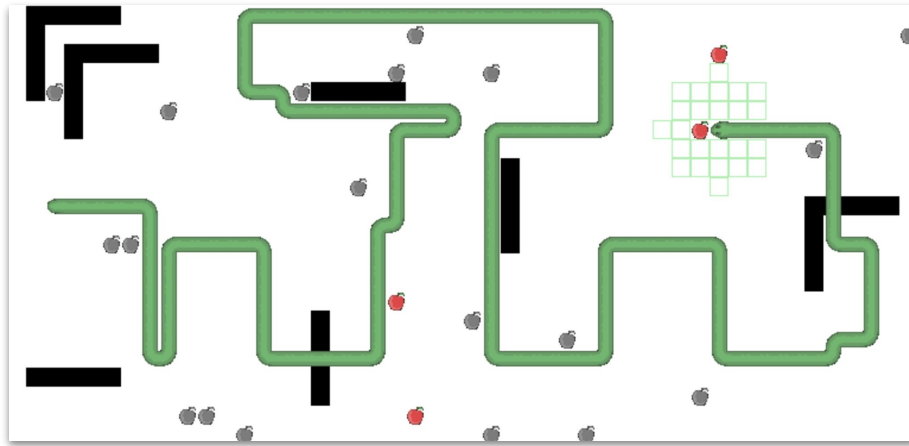# Snake Game
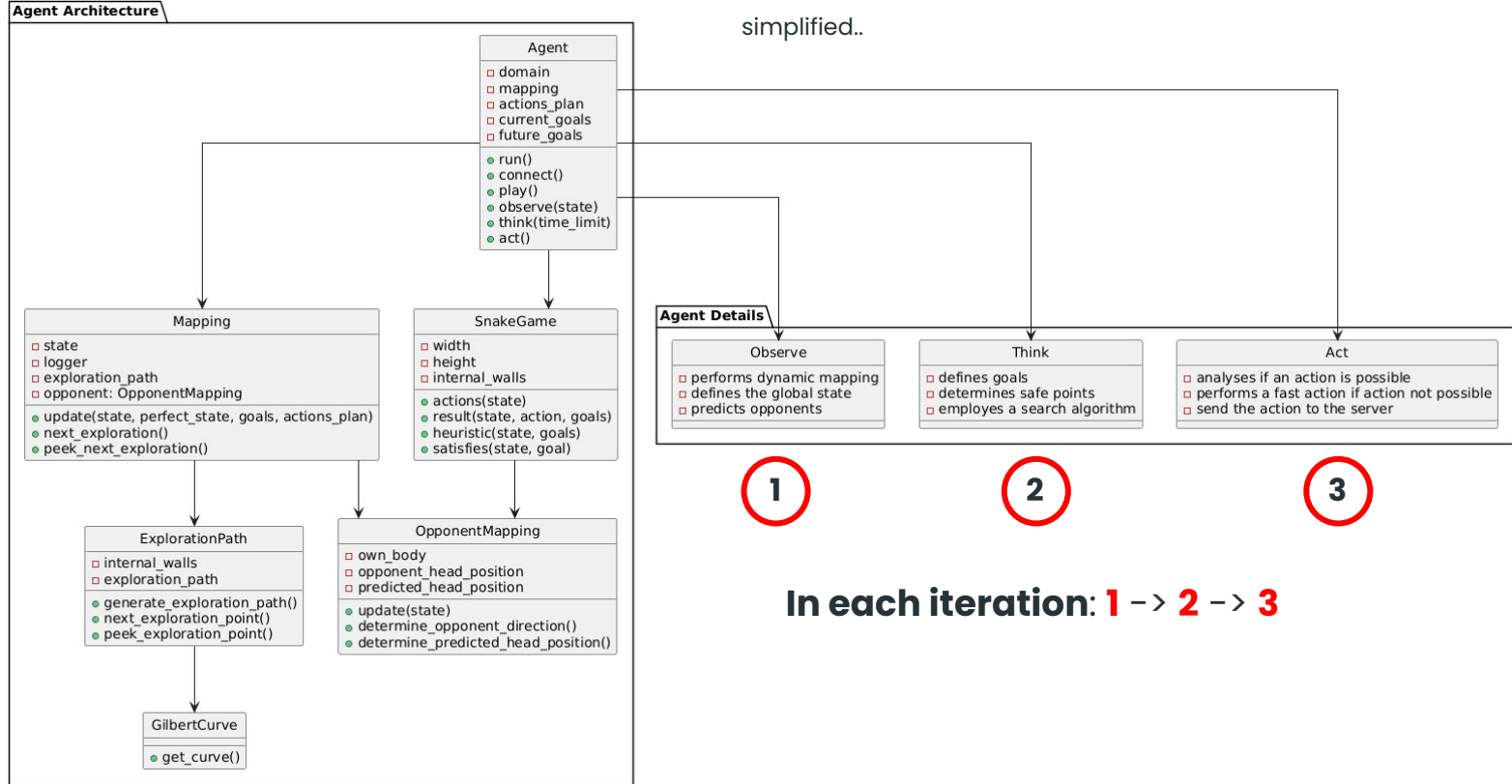# IA Project

João Pinto - 104384
Guilherme Santos - 113893
Pedro Pinto - 115304

23.12.2024

Agent Architecture
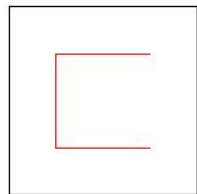
simplified..

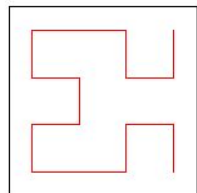**Agent**
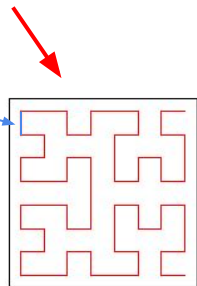- □ domain
- □ mapping
- □ actions_plan
- □ current_goals
- □ future_goals

- ● run()
- ● connect()
- ● play()
- ● observe(state)
- ● think(time_limit)
- ● act()

Mapping
- □ state
- □ logger
- □ exploration_path
- □ opponent: OpponentMapping

- ● update(state, perfect_state, goals, actions_plan)
- ● next_exploration()
- ● peek_next_exploration()

SnakeGame
- □ width
- □ height
- □ internal_walls

- ● actions(state)
- ● result(state, action, goals)
- ● heuristic(state, goals)
- ● satisfies(state, goal)

ExplorationPath
- □ internal_walls
- □ exploration_path

- ● generate_exploration_path()
- ● next_exploration_point()
- ● peek_exploration_point()

OpponentMapping
- □ own_body
- □ opponent_head_position
- □ predicted_head_position

- ● update(state)
- ● determine_opponent_direction()
- ● determine_predicted_head_position()

GilbertCurve
- ● get_curve()

Agent Details

Observe
- □ performs dynamic mapping
- □ defines the global state
- □ predicts opponents

Think
- □ defines goals
- □ determines safe points
- □ employes a search algorithm

Act
- □ analyses if an action is possible
- □ performs a fast action if action not possible
- □ send the action to the server

① **②** **③**

In each iteration: **1** –> **2** –> **3**

Explanation of the Algorithm

- Implements a **mapping module** to track explored areas and dynamic objects.

Goal

Safe points

Explored areas

2x sight_range

1. find path to **Goal**
2. find path to **Safe points**
   (ensuring a safe exit, even with dead ends)

- Determines optimal actions based on defined **goals** and **constraints**.

- Uses **A* search** algorithm to plan paths toward goals like **food** and **safe points.**

- Incorporates **Gilbert curves** for unvisited areas while avoiding obstacles and opponents, minimizing redundant movements (a dual of Hilbert Curves).

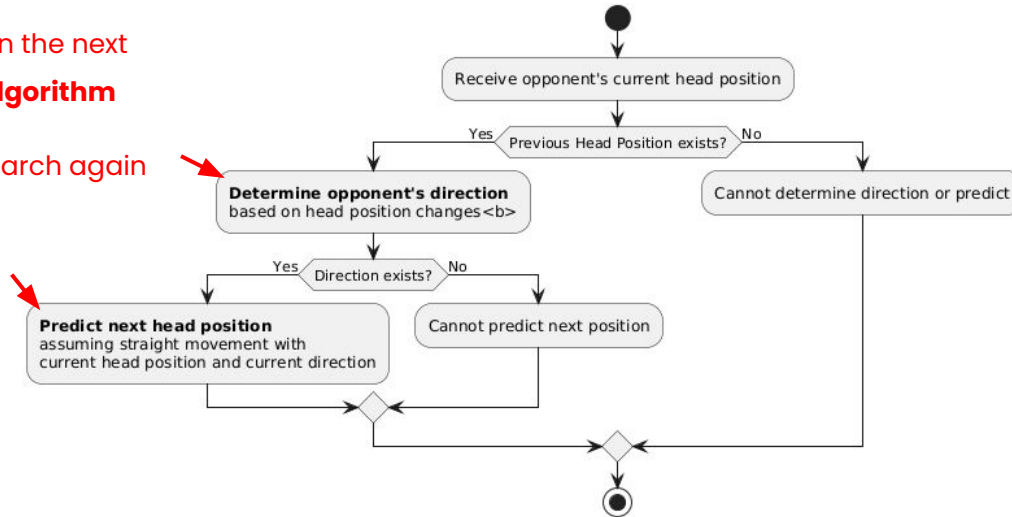- Tracks and predicts **opponent movements** using *OpponentMapping.*

Gilbert Curves

3

Multiplayer mode

- **Updates** information about the **opponent's** position and direction.

- **Predicts** the **next head position** of opponents based on current direction and target food. And **integration in search**.

Predictions to use in the next iteration **search algorithm**

If prediction fails: search again

Receive opponent's current head position

Previous Head Position exists?
Yes — No

Determine opponent's direction
based on head position changes<b>

Cannot determine direction or predict

Direction exists?
Yes — No

Predict next head position
assuming straight movement with
current head position and current direction

Cannot predict next position

This is running inside the *OpponentMapping* for each iteration

Agent Benchmark

- Developed a shell script *test-agent.sh*, designed to **automate the process of testing an AI agent** multiple times, collecting metrics like scores and steps, and calculating the average score over several runs.

- The script enables easy comparison between runs and **tracking progress** over time.

Score	Run #1	Run #2	Run #3	Run #4	Run #5	
without G.Curves	75	86	71	68	48	~ 69.6
without safe points	49	100	22	98	120	~ 77.8
with safe points	104	155	102	94	168	~ 124.6

with G. Curves

Conclusions

- **Key Achievements:**
  - Developed an algorithm that **balances** exploration and goal-driven fruit/super-fruit collection effectively.
  - Achieved safe navigation through obstacles using safe points and traversal logic.
  - Single point heuristic & **Multiple points heuristic** (we use this special heuristic for close goals)
- **Strengths of the Approach:**
  - **Efficiency:** Optimized grid traversal with Gilbert curves, minimizing redundant moves.
  - **Scalability:** Algorithm adapts well to larger grids and complex scenarios.
  - **Deterministic Results:** Predictable and reliable behavior under varying conditions.
  - **Versatility:** Effective handling of limited sight, unseen areas, and exploration priorities.
  - **Attack/Defense:** Easy creation of dead ends by following the curves.
- **Benchmarks Summary:**
  - Consistent performance improvement compared to baseline models.
  - High success rate in fruit collection and grid exploration tasks.