# Introduction:

This report describes the design and functionality of **MyVerySmartAgent**. The agent uses effective techniques and a suspicion-based decision-making to evaluate the behaviour of other players and adjust its strategy, based on whether it is a spy or a resistance member. The goal of the agent is to outperform baseline agents by intelligently proposing missions, voting, and betraying missions when necessary.

# Agent Strategy:

Suspicion System

The agent assigns a suspicion score to each player, starting at 1. These scores adjust based on mission outcomes and voting behaviour.
My initial impression was to rely solely on good techniques, ignoring players' past behaviours. I struggled with this approach because it felt too simplistic and overlooked important context. Ultimately, I concluded that incorporating suspicion scores provides a clearer understanding of players' intentions.

Mission Proposals

- **As a Resistance member**: The agent proposes teams of the lowest suspicion players, including itself.
- **As a Spy**: It selects least suspicious spies and highly suspicious non-spy players to balance trust and sabotage.

This part became clear when I played with friends. This was facilitated by the fact that we have a suspicion score system as a way to assess players' behaviour. One potential strategy for the spies is to include the most suspicious players in mission proposals, as they are likely to be identified by skilled agents (if there are). By doing so, we can keep the suspicion levels of the spies with the lowest scores concealed, allowing them to operate more by being on missions more frequently.

Voting Behaviour

The agent votes based on team composition and accumulated suspicion:

- **Resistance**: It approves teams with low suspicion totals and rejects those that seem risky.
- **Spy**: It approves missions that have enough spies to fail the mission and rejects others.

The agent's voting strategy reflects its role in the game. As a Resistance member, the agent evaluates the risk of a mission by comparing the sum of the team members' suspicion scores to a threshold. This threshold is based on the number of players and their initial suspicion scores at the start of the game. Additionally, the agent's leeway decreases after each round, as it gathers more information and becomes less tolerant of suspicious behaviour. One way to improve the agent's strategy is to have its leeway decrease exponentially instead of linearly. This would make the agent much more cautious in later rounds. However, this will lead to a limited learning experience because if the agent becomes overly cautious too quickly, it might miss out on valuable opportunities to observe and learn from player behaviours.

Betrayal Logic

- **Spy**: The agent betrays when there are enough spies to guarantee mission failure, otherwise, it avoids betraying to reduce suspicion.

There's no point in betraying if it won't cause the mission to fail as it only exposes the spy needlessly. I experimented with a strategy where spies avoid betraying if other spies were already enough to fail the mission. The idea was to stay under the radar. However, this didn't improve performance because random agents sometimes didn't betray even when they should have, making it harder to predict outcomes. The inconsistent behaviour from certain agents nullified the advantage of staying hidden.

# Learning Techniques and Design Choices

Suspicion Adjustment:

Suspicion scores are recalculated based on voting and mission outcomes. This adaptive system allows the agent to make increasingly informed decisions as it learns more throughout the rounds.

In **MyVerySmartAgent**, the suspicion score changes based on several factors:

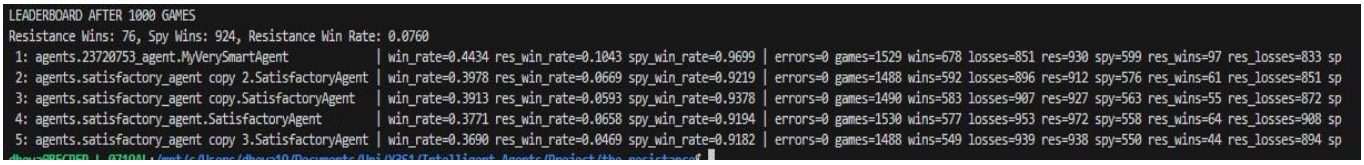
1. **Mission Success/Failure**:
    - When a mission **succeeds**, suspicion decreases for team members, as they are less likely to be spies.
    - When a mission **fails**, suspicion increases for all players on the mission, since they could be responsible for the sabotage.
2. **Voting Behavior**:
    - Players who vote **against a successful mission** see their suspicion score increase, as their vote suggests potential sabotage intentions, otherwise decrease suspicion for players who voted for **against a successful mission**
    - Players who vote **for a failed mission** have their suspicion increased, as supporting a failed mission is suspicious, otherwise decrease suspicion for players who voted **for a failed mission**

Leeway Mechanism:

The agent becomes more cautious with each round once it has learned more information on other players, reducing its leeway as it less tolerant of suspicious behaviour. The suspicion scores become more influential in decision-making towards the later rounds.

# Performance:

```
LEADERBOARD AFTER 1000 GAMES
Resistance Wins: 287, Spy Wins: 713, Resistance Win Rate: 0.2870
 1: MyVerySmartAgent  | win_rate=0.5435 res_win_rate=0.3450 spy_win_rate=0.8526 | errors=0 games=1838 wins=999 losses=839 res=1119 spy=719 res_wins=386 res_losses=733 spy_wins=613 spy_losses=106
 2: BasicAgent        | win_rate=0.4657 res_win_rate=0.2951 spy_win_rate=0.7588 | errors=0 games=1849 wins=861 losses=988 res=1169 spy=680 res_wins=345 res_losses=824 spy_wins=516 spy_losses=164
 3: SatisfactoryAgent | win_rate=0.4654 res_win_rate=0.2822 spy_win_rate=0.7711 | errors=0 games=1865 wins=868 losses=997 res=1166 spy=699 res_wins=329 res_losses=837 spy_wins=539 spy_losses=160
 4: RandomAgent       | win_rate=0.3208 res_win_rate=0.1966 spy_win_rate=0.5198 | errors=0 games=1842 wins=591 losses=1251 res=1134 spy=708 res_wins=223 res_losses=911 spy_wins=368 spy_losses=340
dheya@BECPER-L-0719AL:/mnt/c/Users/dheya19/Documents/Uni/Y3S1/Intelligent Agents/Project/the_resistance$
```




In the first scenario, the agent was tested against RandomAgent, BasicAgent, and SatisfactoryAgent. It achieves on average around 53% win rate (res_win_rate ≈0.33 spy_win_rate ≈0.88) and peaking at around 55% (res_win_rate ≈0.34 spy_win_rate ≈0.90), with a ≈9% difference from second place.

In the second scenario, the agent was also evaluated in a scenario featuring only several SatisfactoryAgents and consistently outperformed the other agents. It achieves on average around 44% win rate (res_win_rate ≈0.11 spy_win_rate ≈0.97) with a ≈5% difference from second place.

| **Agent (features added)** | **Overall win rate** | **Resistance win rate** | **Spy win rate** |
|---|---|---|---|
| Skeloton (random template) | 39% | 35% | 46% |
| + betrayal logic | 44% | 26% | 80% |
| + suspicion score (only looks at mission outcome) | 45% | 26% | 83% |
| + voting history (considers voting history that led to that outcome) | 48% | 27% | 85% |
| + leeway | 51% | 29% | 86% |
| + adjusted suspicion values | 53% | 33% | 88% |

# Possible Strategies:

In designing MyVerySmartAgent, the following strategies were considered:

- Bayesian Networks: Could estimate the probability of each player being a spy based on mission outcomes and voting patterns. However, with only five rounds, there wouldn't be enough data to generate accurate probabilities, so this approach was not pursued.
- Monte Carlo Simulation: Could simulate game outcomes to optimize decisions, but was avoided due to its computational cost and the game's strict time limits.

**Merits of Multiple Strategies**

1. Bayesian Networks:
    - Offers a probabilistic way to assess suspicion, updating based on player actions.
    - Pros: Uses prior knowledge, captures how one player's actions influence others.
2. Monte Carlo Simulations:
    - Explore various game outcomes, helping optimize team selection.
    - Pros: Handles uncertainty, assesses risk, and allows extensive exploration of scenarios.

**Justification for Chosen Strategy:**

The rule-based system is effective because it is simple and fast, making decisions based on suspicion scores, mission outcomes, and voting patterns. It adapts dynamically to the game's changing conditions without exceeding the strict time limit. It allows the agent to perform well within the game's constraints.