**Spike:** 10
**Title:** Tactical Bot vs Naive Bot – PlanetWars AI

**Author:** Marco Giacoppo, 104071453

**Goals / deliverables:**
Create and compare two AI bots for PlanetWars: one naïve and one tactical. The naïve bot (Rando) selects planets randomly, while the tactical bot (SmartBoy) evaluates distance and enemy strength before attacking. This spike includes:

- The development of both bots in Python
- Tactical logic using distance + ship count
- Performance testing across three maps (15 matches total)

**Technologies, Tools, and Resources used:**

- Python 3.11
- Pyglet (for game visualisation)
- Terminal for running matches
- Map files (map002, map003, map006)

**Tasks undertaken:**

1. Understand the PlanetWars Game Structure; Open main.py and players.py to understand how bots are loaded and updated. Bots receive a Player object and can access helper methods like _my_planets() and planet_order().

2. Create the Naive Bot (BestBot.py); Inside the /bots folder, create a new Python file called BestBot.py. The bot picks the strongest planet and sends 75% of its ships to the weakest enemy or neutral target each turn.

3. Create the Tactical Bot (SmartBot.py); Create a second bot in /bots named SmartBoy.py. Implement tactical decision-making by:

    - Selecting the strongest or viable planets as source
    - Using a scoring function that adds planet ship count + distance (weighted) and growth rate to evaluate targets
    - Sending only the necessary ships (with a safety buffer) to win
    - Adapting behaviour based on map state and game phase

4. Add Logging; In both bots, add print() statements inside the fleet launch section to show source, destination, and fleet size. This helps verify the bot’s decisions during matches.

5. Test Your Bots; Run matches using the command:

   *python3 main.py --gui -m map002 -p Rando SmartBoy*

   Replace map002 with any available map. Press [ and ] to switch player views and observe their behaviour.

6. Record Results; For each match, record the state by writing it into a file I called results_log.txt. The format will be; map name, players, winner and game tick.

```
# ---- Log winner and result to file ----

# Get map and player info directly from args
map_name = args.map if args.map else "unknown"
players = args.players if args.players else ["UnknownBot1", "UnknownBot2"]
winner = game.get_winner_name()

# Print to console
print(f"Game Over. Winner: {winner}")

# Write to results log
with open("results_log.txt", "a") as f:
    f.write(
        f"Map: {map_name} | Players: {players[0]} vs {players[1]} | Winner: {winner} | Tick: {game.tick}\n"
    )
```

7. Analyse Performance: Compare win rates and identify patterns. Note which strategies outperform others and under what conditions.

8. Write the Report; Summarize your process, outcomes, and recommendations for future bot improvements. Include your result table and a discussion of how tactical decision-making influenced performance.

**What we found out:**

SmartBot initially performed significantly better across early maps tested. The advantage of tactical targeting was especially evident on maps with varied planet spacing and values. SmartBot was able to prioritize closer, weaker planets more efficiently than BestBot.

SmartBot's strength lies in its scoring heuristic. For each enemy or neutral planet, it calculates a "tactical score" using:

```
def tactical_score(p):
    return (p.ships - p.growth * 3) + src.distance_to(p, sqrt=True) * 0.3
```

This function considers both the number of defending ships (*p.ships*) and the distance from the source planet. The bot selects the planet with the **lowest** score, prioritizing targets that are weak and nearby.

SmartBot then launches a fleet using:

```
if src.ships > required + 10:
    gameinfo.planet_order(src, target, required)
    print(f"[SmartBotCounter] Attacking {target.ID} from {src.ID} with {required} ships")
```

This logic ensures SmartBoy only commits to attacks when it has enough ships, helping avoid reckless or inefficient moves.

However, further testing revealed more insights. SmartBot tends to struggle on simpler maps with fewer planets. On these maps, BestBot often focuses early aggression directly on SmartBot's starting planet. Since SmartBot is designed to evaluated options tactically and expand, it can be caught off-guard by this brute-force approach.

In contrast, SmartBot performs much better on complex maps where there are many planets near the starting position. The additional options give SmartBot the ability to expand rapidly, build up ship production, and leverage its tactical decision-making to overwhelm BestBot's predictable behaviour.

All in all, map complexity directly affects SmartBot's effectiveness. On simple maps, it gets boxed in and overrun. On complex maps, it thrives with efficient expansion and multi-source attacks.

Result table (for first version of SmartBot). I've run these test on the same maps 5 times each and it gives the same result. So I'll just put one on each:

| MAPS | PLAYERS | WINNER | TICK |
|---|---|---|---|
| map010 | SmartBot vs BestBot | BestBot | 195 |
| map011 | SmartBot vs BestBot | SmartBot | 241 |
| map012 | SmartBot vs BestBot | BestBot | 154 |
| map013 | SmartBot vs BestBot | SmartBot | 261 |
| map014 | SmartBot vs BestBot | BestBot | 369 |
| map015 | SmartBot vs BestBot | BestBot | 675 |
| map016 | SmartBot vs BestBot | SmartBot | 125 |
| map017 | SmartBot vs BestBot | SmartBot | 586 |
| map018 | SmartBot vs BestBot | SmartBot | 259 |
| map019 | SmartBot vs BestBot | SmartBot | 2067 |
| map020 | SmartBot vs BestBot | BestBot | 285 |

| | | | |
|---|---|---|---|
| map002 | SmartBot vs BestBot | BestBot | 475 |
| map003 | SmartBot vs BestBot | BestBot | 492 |
| map004 | SmartBot vs BestBot | BestBot | 634 |
| map005 | SmartBot vs BestBot | BestBot | 79 |
| map006 | SmartBot vs BestBot | BestBot | 106 |
| map007 | SmartBot vs BestBot | BestBot | 108 |
| map008 | SmartBot vs BestBot | BestBot | 79 |
| map009 | SmartBot vs BestBot | BestBot | 512 |

Overall Summary:
- Total Matches: 19
- SmartBot Wins: 7
- BestBot Wins: 12
- SmartBot Win Rate: 36.8%

This highlights that SmartBot still struggled against BestBot on maps that were small or densely packets, where early aggression mattered more than long-term planning. To address this, a new version of the bot was created. Here's the details:

SmartBotv2 dinamically adjusts its behaviour based on map structure:
- On small maps (<= 5 planets): it uses **rush mode**, mimicking BestBot's aggressive tactics.
- On mid-size maps (6-10 planets): it starts aggressive, then transitions to tactical logic.
- On large maps (> 10 planets): it uses **tactical logic** from the strart, prioritizing growth and efficient targeting.

The bot uses total map size and current tick to choose when to send a lot of fleets or carefully evaluate using a weighted score. This improvement made SmartBotv2 far more competitive and less predictable.

Result Table (SmartBotv2 vs BestBot):

| MAP | PLAYER | WINNER | TICK |
|---|---|---|---|
| map002 | SmartBotv2 vs BestBot | SmartBotv2 | 78 |
| map003 | SmartBotv2 vs BestBot | SmartBotv2 | 352 |
| map004 | SmartBotv2 vs BestBot | SmartBotv2 | 38 |
| map005 | SmartBotv2 vs BestBot | BestBot | 58 |
| map006 | SmartBotv2 vs BestBot | BestBot | 106 |
| map007 | SmartBotv2 vs BestBot | BestBot | 642 |
| map008 | SmartBotv2 vs BestBot | SmartBotv2 | 346 |
| map009 | SmartBotv2 vs BestBot | BestBot | 513 |
| map010 | SmartBotv2 vs BestBot | BestBot | 612 |
| map011 | SmartBotv2 vs BestBot | SmartBotv2 | 64 |
| map012 | SmartBotv2 vs BestBot | Undecided | 10000 |
| map013 | SmartBotv2 vs BestBot | SmartBotv2 | 100 |

| | | | |
|---|---|---|---|
| map014 | SmartBotv2 vs BestBot | SmartBotv2 | 257 |
| map015 | SmartBotv2 vs BestBot | SmartBotv2 | 340 |
| map016 | SmartBotv2 vs BestBot | SmartBotv2 | 98 |
| map017 | SmartBotv2 vs BestBot | SmartBotv2 | 323 |
| map018 | SmartBotv2 vs BestBot | SmartBotv2 | 449 |
| map019 | SmartBotv2 vs BestBot | SmartBotv2 | 384 |
| map020 | SmartBotv2 vs BestBot | BestBot | 212 |

**Final Win Rate: 68.4%**

This performance shows that hybrid AI with adaptive mode switching based on map layout and game phase leads to a much stronger bot capable of countering naïve, aggressive strategies like BestBot. SmartBot is now not just tactical, its adaptive and map-aware.