Killer Queen's behavior

V.1

Its name comes from 「**Killer Queen**」, an antagonist in the popular manga series *JoJo's Bizarre Adventure* that can automatically track and blow up its targets.

Seemingly unescapable, the heroes manage to discover that Killer Queen's ability is only targeting the hottest object in the room.

Our agent is inspired by this idea that a behaviour doesn't have to be complex to be fearsome. Its logic is fairly simple:

> Each time Killer Queen is asked to target someone, it ranks players from **hot** (high priority) to **cold** (low priority). Then, it chooses the one with the highest heat.

At the end, everything comes down to how you calculate "heat".

Calculating heat

For each possible action, there's a different **heat function**. They are all based on the following metrics.

Metrics

Role probabilities

Bayesian probabilities of being a specific role or in a specific team, using certain information. Eg: using DIVINE on someone, knowing other WEREWOLVES, etc.

See killerQueen/roleEstimations.py

Hostility

Measures the hostility of each agents towards ours, using their textual declarations. We established a list of patterns associated with a score, that indicates if an agent is hostile or not. Eg :

```
Agent 01 : COMINGOUT [KillerQueen] WEREWOLF
```

Agent 01 is saying we are a WEREWOLF. We add +10 to its hostility metric.

```
Agent 02 : DIVINE [KillerQueen] HUMAN
```

Agent 02 is saying we are a HUMAN. We add -40 to its hostility metric.

The final scores goes then through a sigmoid function, to soften the values.

To see the full list of patterns, check killerQueen/textMetrics.py

Complexity

Measures the complexity of an agent. If we're facing two hostile agents, we want to prioritise the one that is higher in the leaderboard because, as it is more dangerous. To estimate the complexity of their behavior, we use a very simple metric: counting the number of brackets in their speech. Eg:

```
ESTIMATE [KillerQueen] WEREWOLF
```

"I estimate KillerQueen is a werewolf"

... is less complex than ...

```
BECAUSE (DAY 4 (DIVINED [KillerQueen] WEREWOLF)) (AND (VOTE
[KillerQueen] (REQUEST ANY ([KillerQueen])))
```

"Everyone should vote for [KillerQueen] because on day 4, I divined they were a werewolf."

See [killerQueen/textMetrics.py]

Heat functions

There is a total of 4 heat functions used to calculate the priority of each target for a specific action:

- `heatVote()` used by every role
- `heatDivine()` used by SEER and MEDIUM
- `heatAttack()` used by WEREWOLF
- `heatGuard()` used by BODYGUARD

Example: the definition of `heatAttack()`

```
rolePriority = roles['SEER']*3 + roles['MEDIUM']*2 + roles['BODYGUARD']*2 + roles['VILLAGER'] -
roles['POSSESSED'] -2*roles['WEREWOLF']

menace = profile['hostility']*profile['complexity']
heat = rolePriority*menace
```