



Présentation Battle Royal

R5.A05 - IOT



Sommaire

L'API

- Choix des routes
- Architecture de l'API
- Modifications de l'Engine

LES AGENTS

- Création d'agent
- Optimisation générale
- Optimisation des statistiques
- Optimisation des choix des actions

KAFKA GRAPHANA

- Mise en place de Kafka
- Confluent
- Graphana

VISUALISATION

- Rappel de l'architecture
- Page de visualisation
- Métriques



L'API

Construction de l'API



Choix des routes

POST

Ajouter un personnage

POST

Ajouter une action et une cible

GET

Obtenir tous les personnages

GET

Obtenir un personnage particulier

GET

Voir le statut de l'Engine : tour en cours, nombre de personnages en vie

DELETE

Supprimer tous les personnage



➤ Architecture de l'API

Routes

main.py

Production des messages

data.py

~~data.json~~

➤ Architecture de l'API



git





Modifications de l'Engine

Vérifications des valeurs

Validation dans le constructeur

```
def __init__(self, cid :str, teamid :str, life :int, strength :int, armor :int, speed :int):  
    self._id = cid  
    self._teamid = teamid  
    self._life = life  
    self._strength = strength  
    self._armor = armor  
    self._speed = speed  
    if self._total_stats() > 20:  
        raise ValueError("Total stats (strength + life + armor + speed) cannot exceed 20.")  
    if self._life < 0 or self._strength < 0 or self._armor < 0 or self._speed < 0:  
        raise ValueError("Stats cannot be negative.")  
    self._action = None  
    self._target = None  
    self._dead = False  
    self._data = Data()
```

Renvoi d'erreur

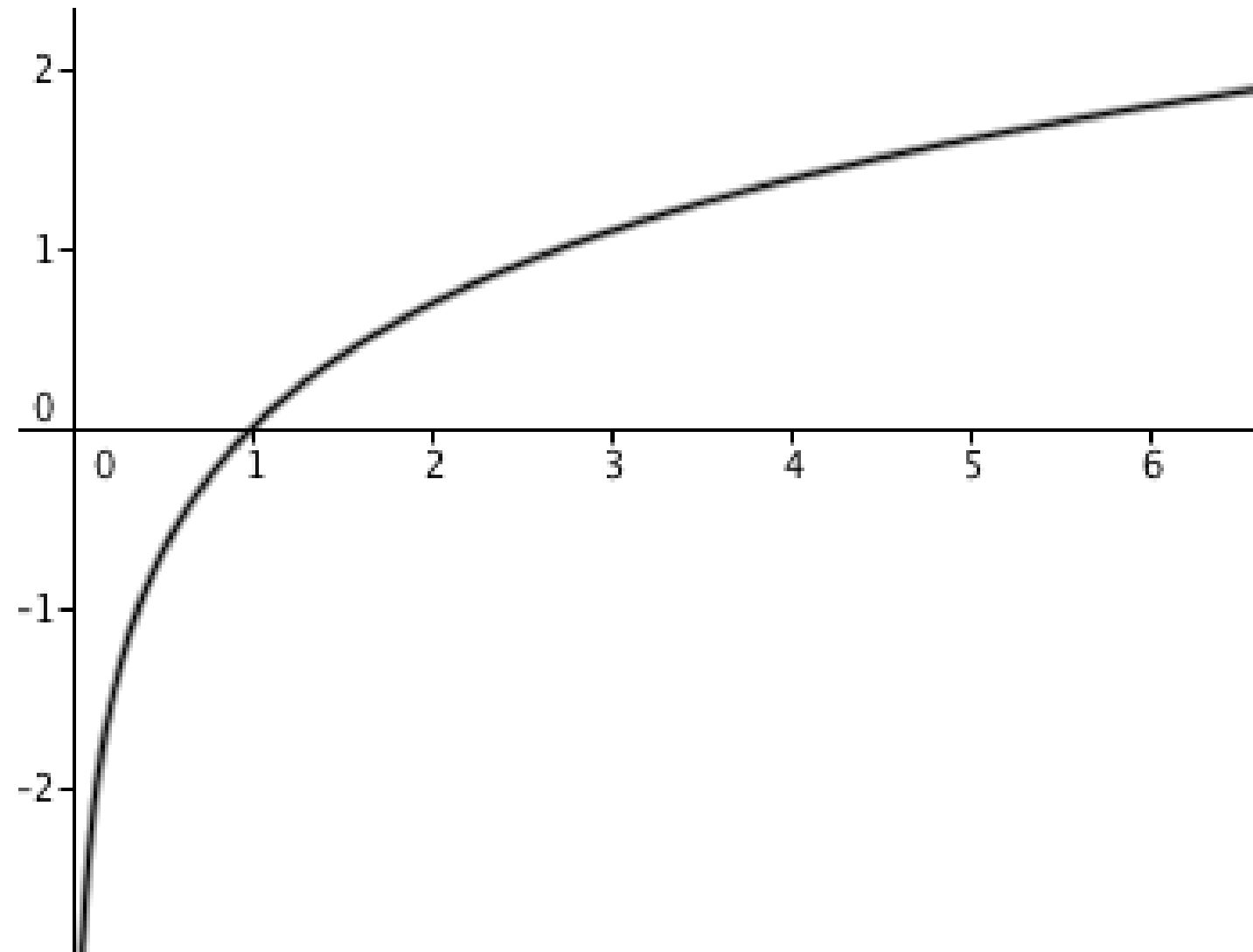
```
1 {"error": "Total stats (strength + life + armor + speed) cannot exceed 20."}
```



Modifications de l'Engine

Refonte du système de blocage

- + Les personnages attaquent dans l'ordre décroissant par vitesse.
- + L'augmentation de dégâts entre deux niveaux est de plus en plus faible.



Les agents

Création d'agent

Optimisation générale

Optimisation des statistiques des agents par renforcement

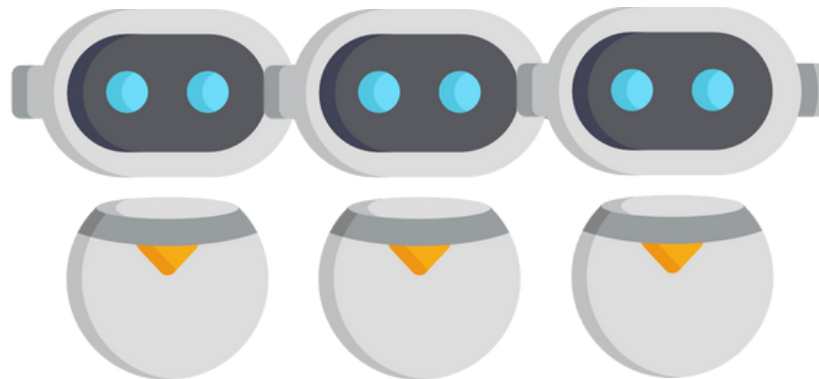
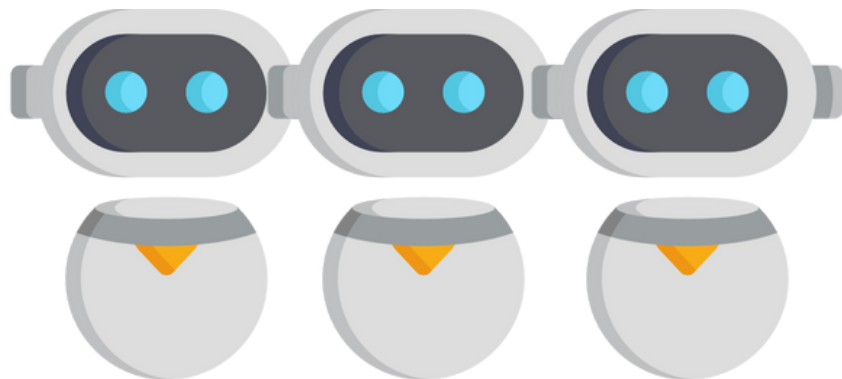
Optimisation des choix des actions des agents par observation algorithmique



Création d'agent

```
def create_agents_rnd(number_of_teams): #3 agents par team
    all_agents = []
    for i in range(number_of_teams):
        for j in range(3):
            agent = Agent(f"player {os.urandom(4).hex()}", f"equipe {i}", 0, 0, 0, 0)
            random_initialize_stats(agent)
            all_agents.append(agent)
            agent.add_to_arena()

    return all_agents
```



```
"player 0397c43a": {
    "armor": 4,
    "cid": "player 0397c43a",
    "dead": false,
    "life": 1,
    "speed": 10,
    "strength": 5,
    "target": "None",
    "teamid": "equipe 0"
},
"player 2cdb13c0": {
    "armor": 2,
    "cid": "player 2cdb13c0",
    "dead": false,
    "life": 3,
    "speed": 10,
    "strength": 5,
    "target": "None",
    "teamid": "equipe 1"
},
```

» Création d'agent

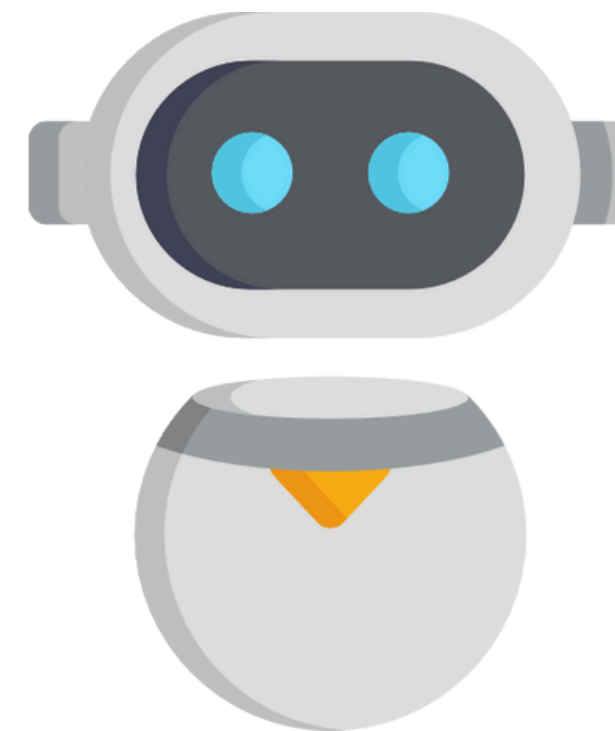
20

speed

armor

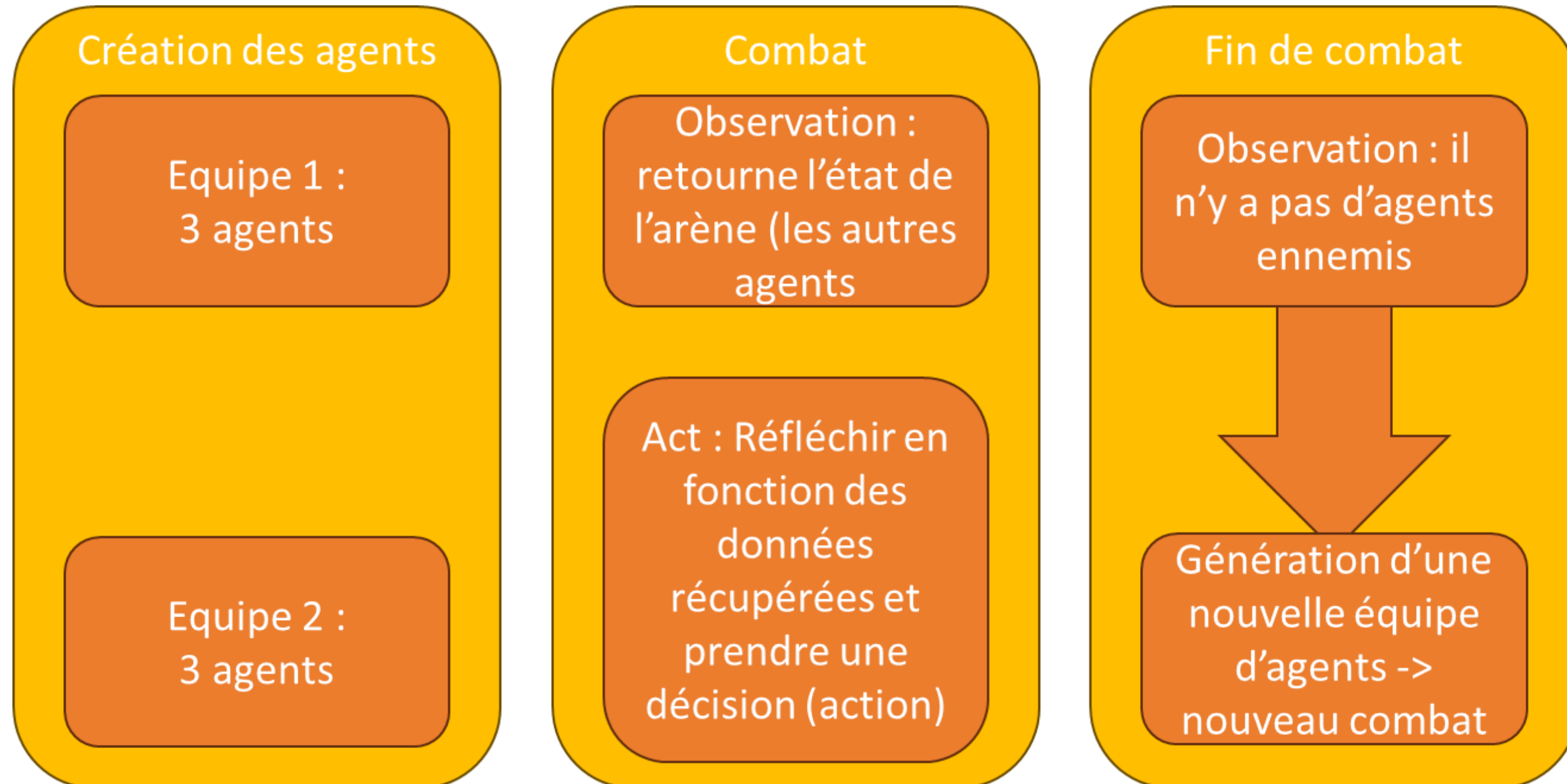
strength

life

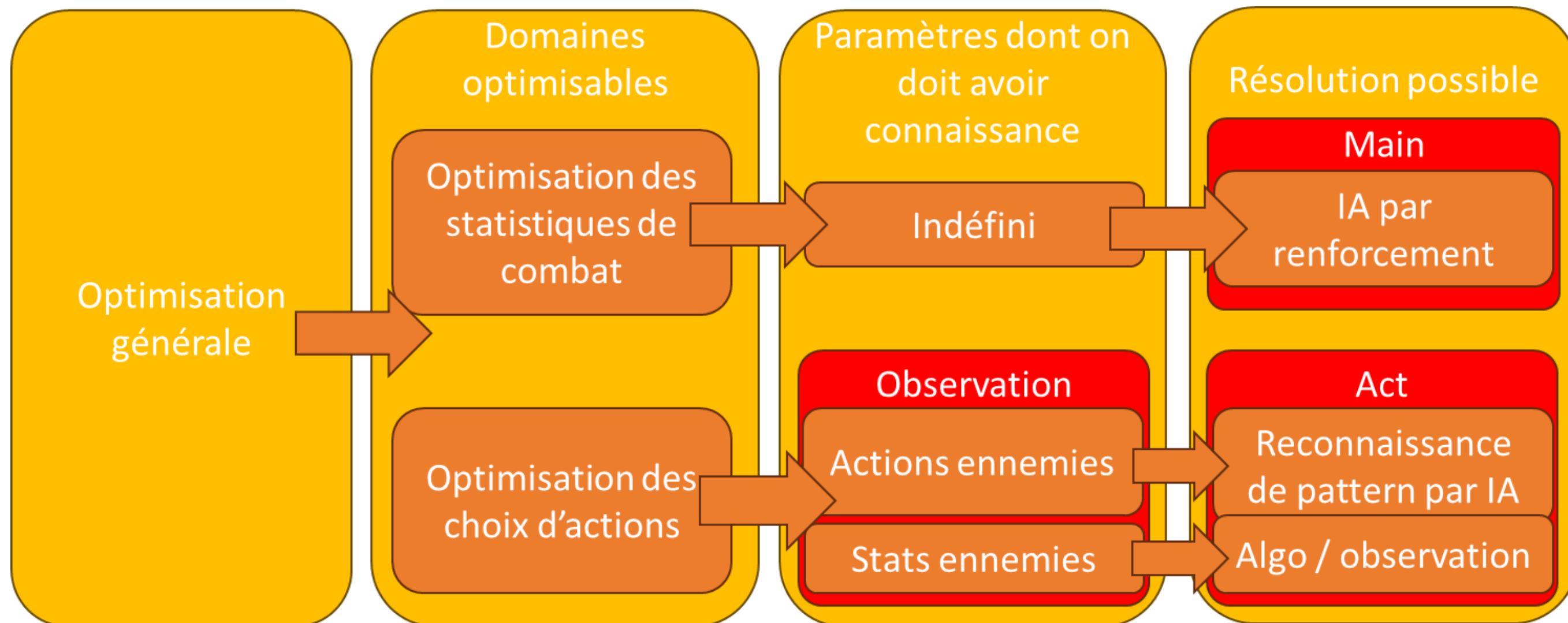


```
def random_initialize_stats(agent):  
    stats_points = 20  
    for i in range(stats_points):  
        field = random.choice(["_life", "_strength", "_speed",  
                               "setattr(agent, field, getattr(agent, field) + 1)"])
```

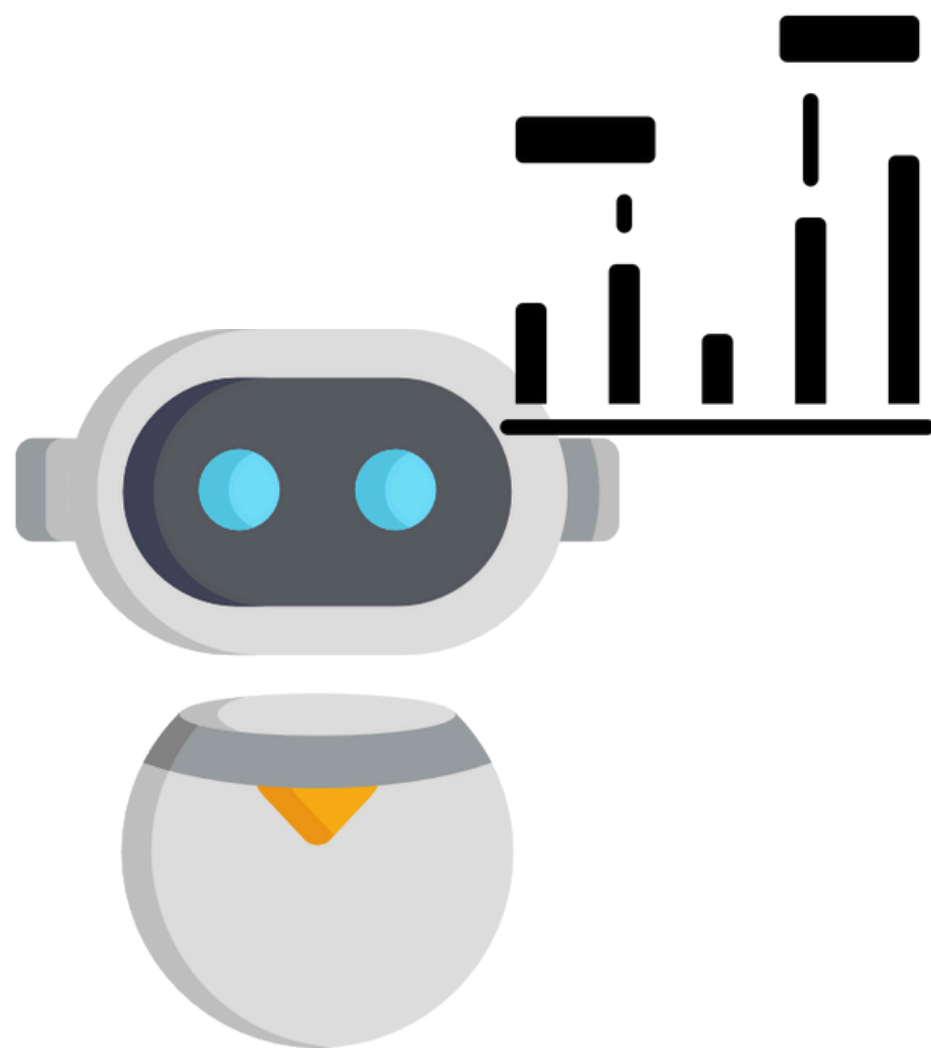
» Optimisation générale



Optimisation générale



Optimisation des actions des agents par observation algorithmique



```
bonus = {'bonus_dodge': 15., 'bonus_hit': 15., 'bonus_block': 15.}
action = ActionEnum.HIT

# // DEFINE BEST ACTION BY GIVING BONUSES TO POSSIBILITIES
agents_targeting_me = len([agent for agent in observation if agent._target == self])
bonus_dodge = (self._speed - CONST_SPEED_MOYENNE) * agents_targeting_me

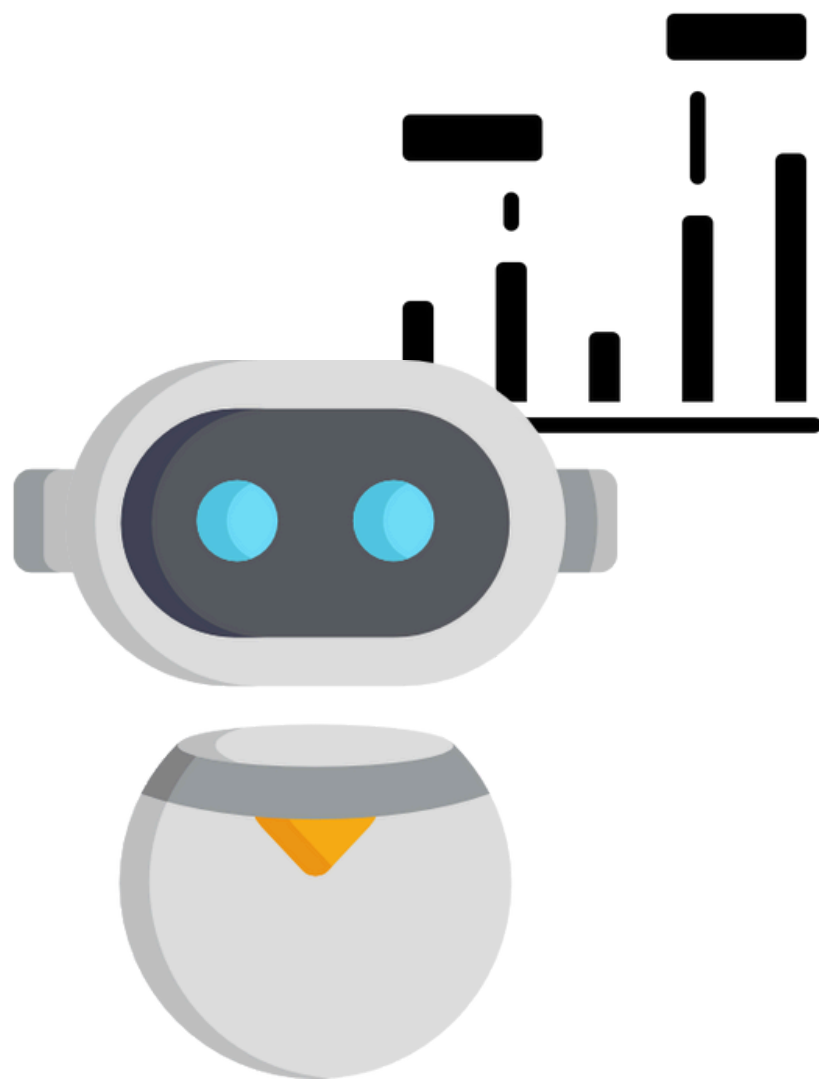
# Si j'ai beaucoup de degats et de speed, je veux vouloir attaquer
bonus_hit = (self._speed - CONST_SPEED_MOYENNE) + (self._strength - CONST_STR_MOYENNE) + (5 * (3 - len(ennemies)))

# Si j'ai beaucoup de defense et peu de speed (donc que j'ai plus de chance de recevoir des degats)
# je veux vouloir reduire
bonus_block = (CONST_SPEED_MOYENNE - self._speed) * (self._armor - CONST_ARMOR_MOYENNE)

# FOCUS : choisir l'agent qui a le plus de degats/speed et le moins d'armure/vie
# Find the agent with the best score
agent_to_focus = max(ennemies, key=lambda agent: agent._strength + agent._speed - (agent._life + agent._armor))
for chosen_bonus_key in ['bonus_dodge', 'bonus_hit', 'bonus_block']:
    bonus[chosen_bonus_key] += locals()[chosen_bonus_key]

for other_bonus_key in bonus:
    if other_bonus_key != chosen_bonus_key:
        bonus[other_bonus_key] -= locals()[chosen_bonus_key] / 2
```

Optimisation des statistiques des agents observation algorithmique



Définition de variables moyennes (moyenne de speed de tous les agents)

Analyse des stats de l'agent par apport à l'agent moyen

Speed : +1,2
Strength : +1,4
Life : - 0,7
Armor: -1,9

Attribution de bonus sur chaque action

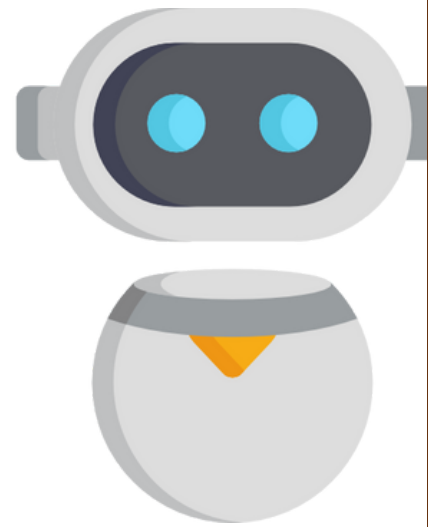
Action.HIT : 33% + 20% = 53%
Action.BLOCK : 33% - 25% = 8%
Action.DODGE : 33% + 5% = 38%

Sélection d'une action finale

Action.HIT

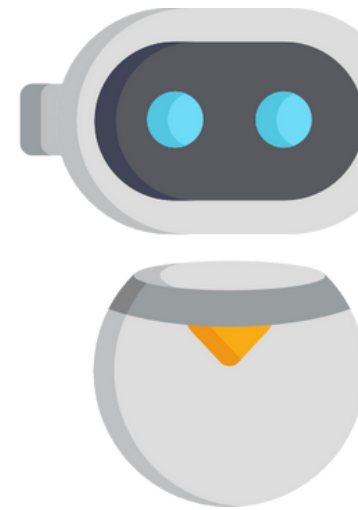
```
action_bonus = random.choices(list(bonus.keys()), weights=list(bonus.values()), k=1)[0]
```

Optimisation des statistiques des agents observation algorithmique



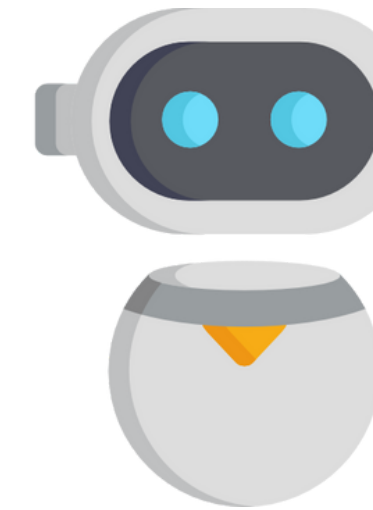
Si rapide et fort
(écart de speed
et de strength
important avec
les autres agents)

Bcp de chances
de taper



Si rapide défensif
(écart de speed
et d'armor
important avec
les autres agents)

Bcp de chances
d'esquiver / taper

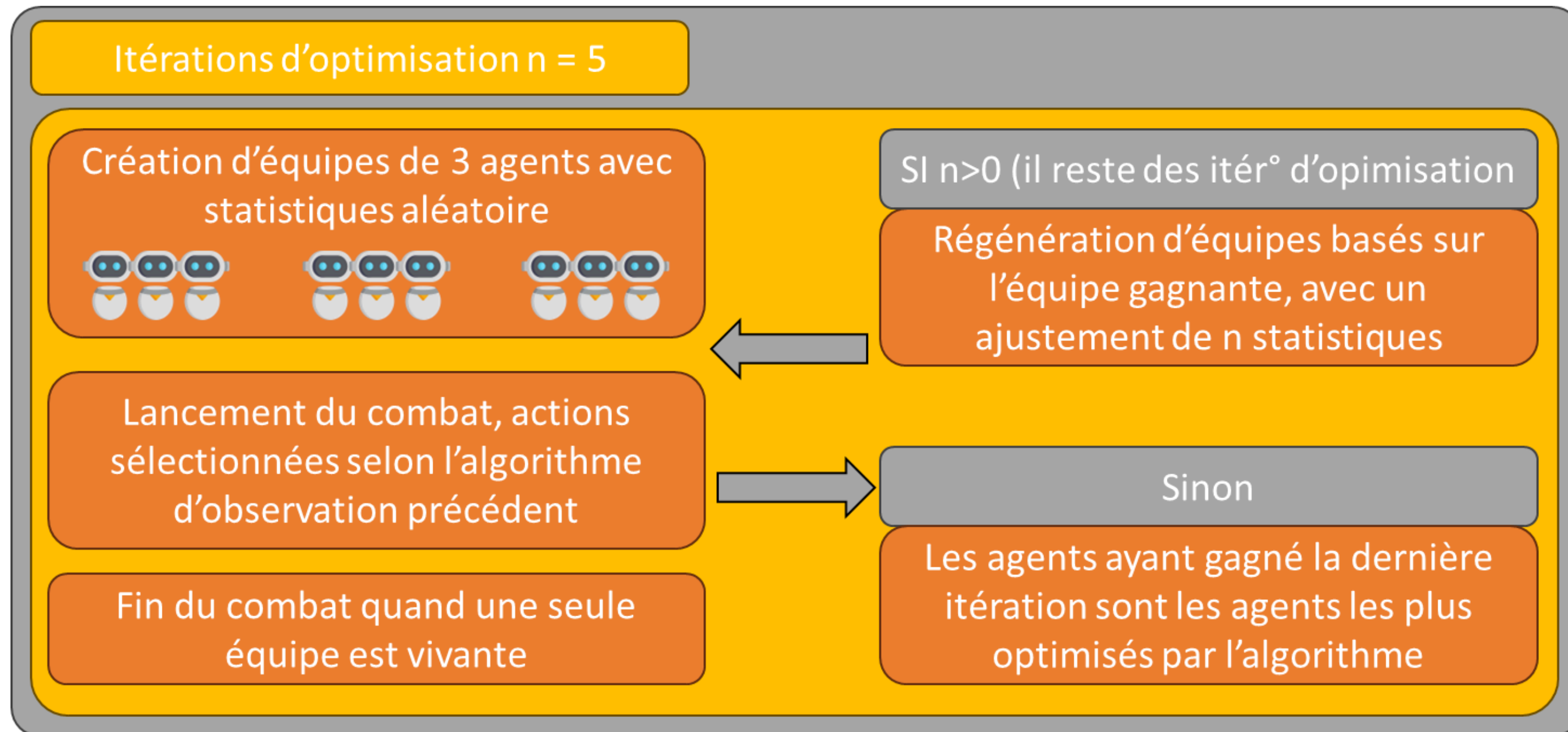


Si lent défensif
(écart de speed
d'armoret vie
important avec
les autres agents)

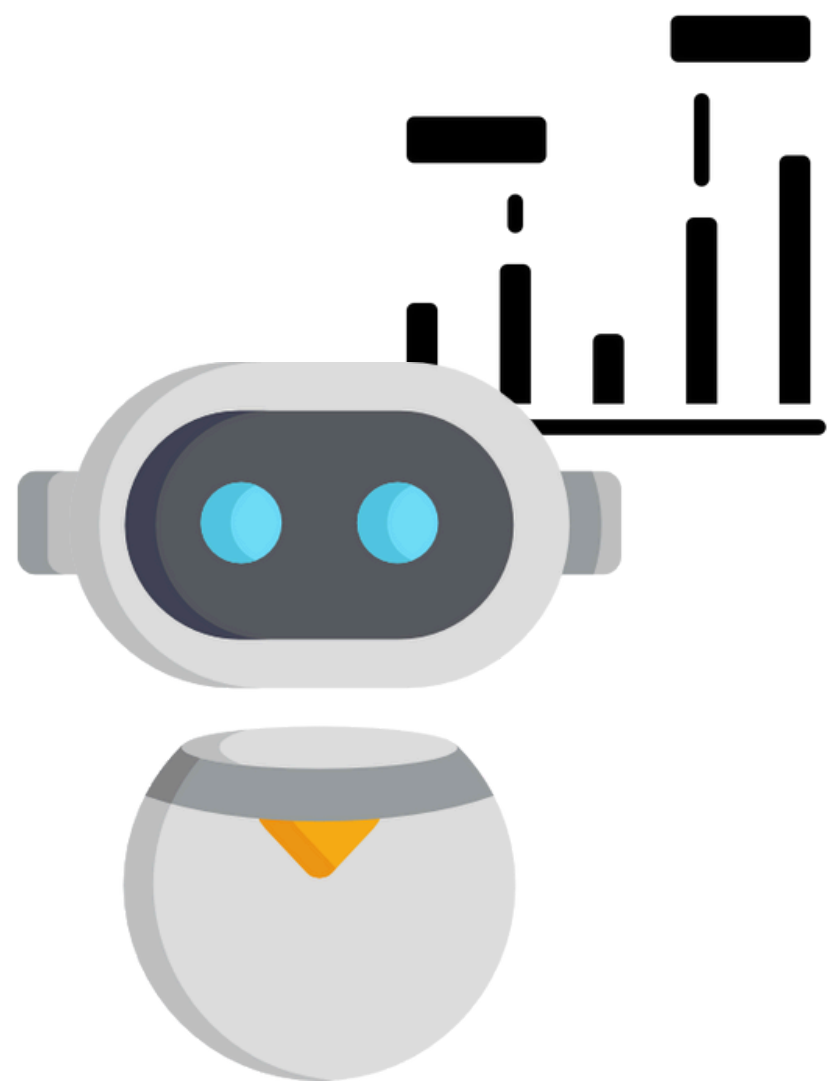
Bcp de chances
de bloquer

```
agents_targeting_me = len([agent for agent in ennemies if agent._target == self])  
bonus_dodge = (self._speed - CONST_SPEED_MOYENNE) * agents_targeting_me
```

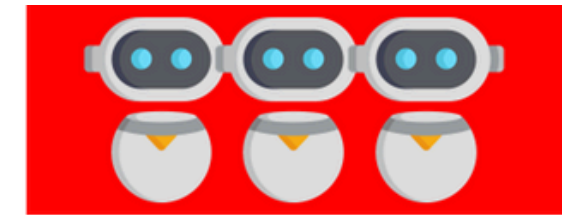
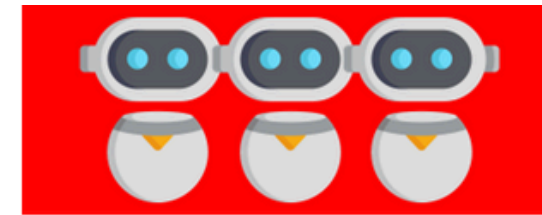
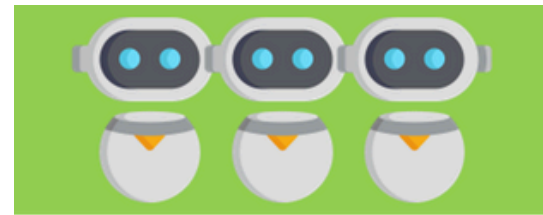

Optimisation des actions des agents par renforcement



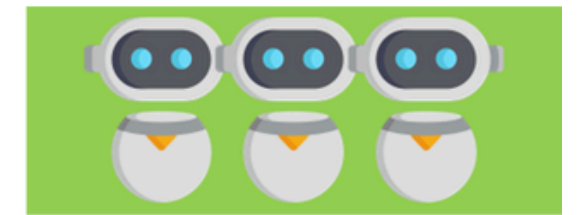
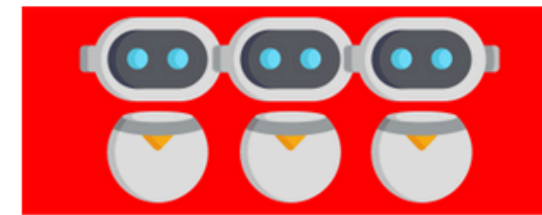
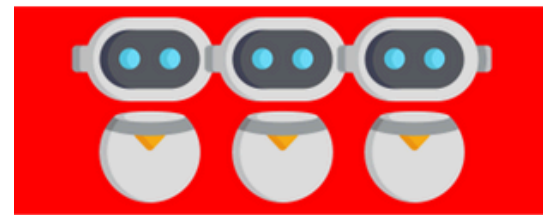
Optimisation des actions des agents par renforcement



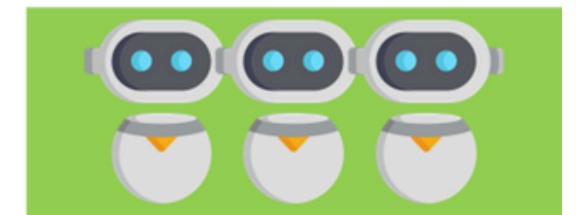
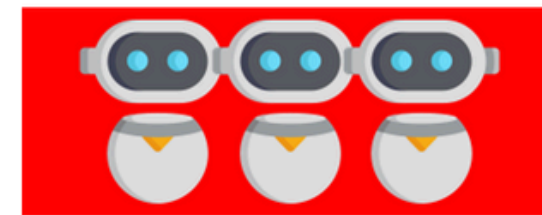
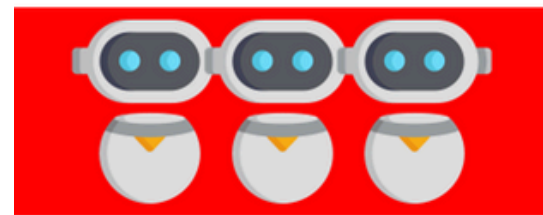
Tour 1



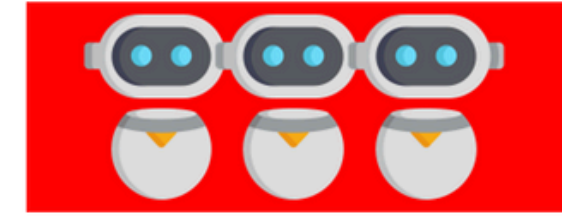
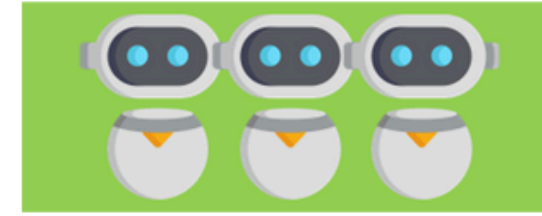
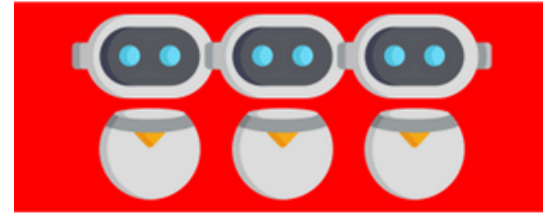
Tour 2



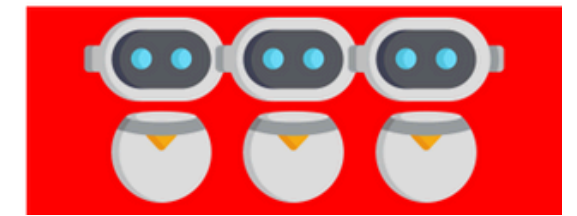
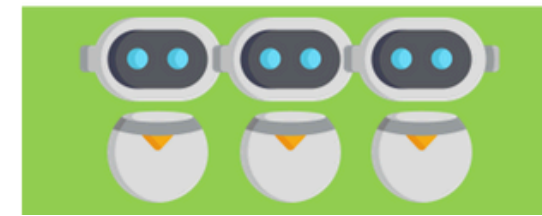
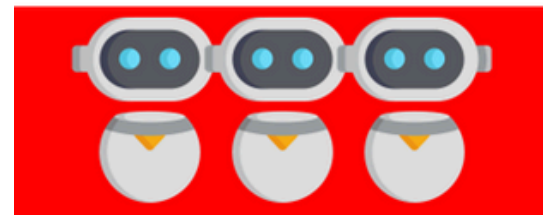
Tour 3



Tour 4



Tour 5

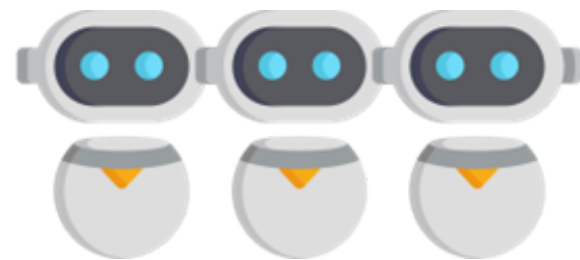
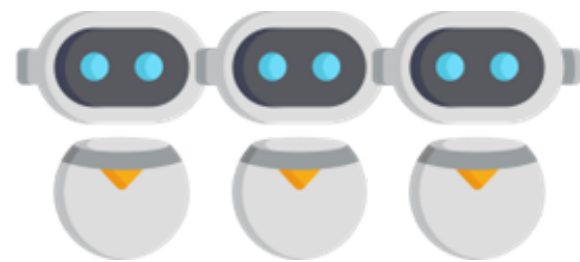
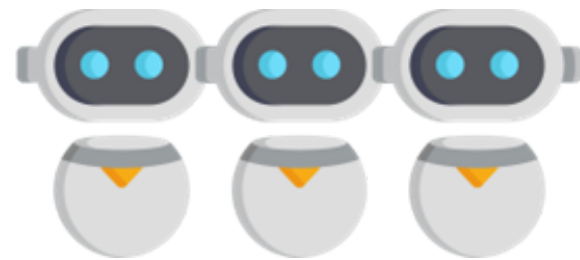
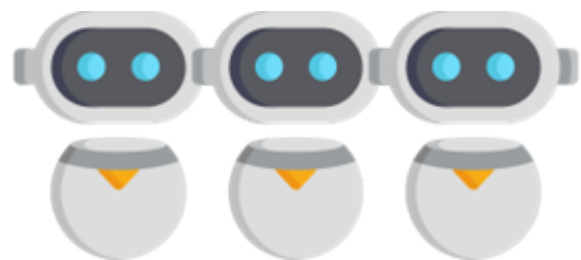
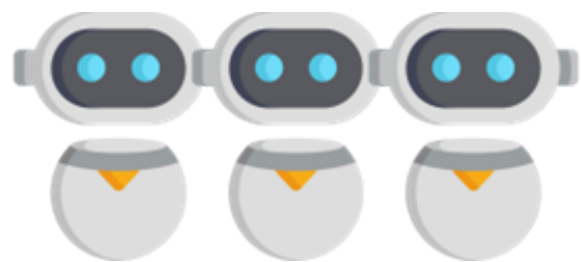
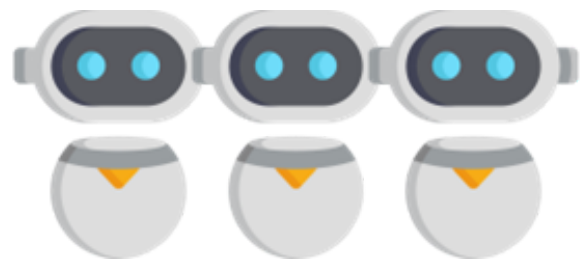


Optimisation des actions des agents par renforcement

Strength : 5
Speed : 6
Armor : 3
Life : 6

Strength : 5
Speed : 6
Armor : 5
Life : 4

Strength : 8
Speed : 2
Armor : 3
Life : 7



'''

Strength : 5
Speed : 8
Armor : 4
Life : 3

'''

'''



Kafka / Graphana

Métriques de notre cluster



Données sur confluent

9 messages shown

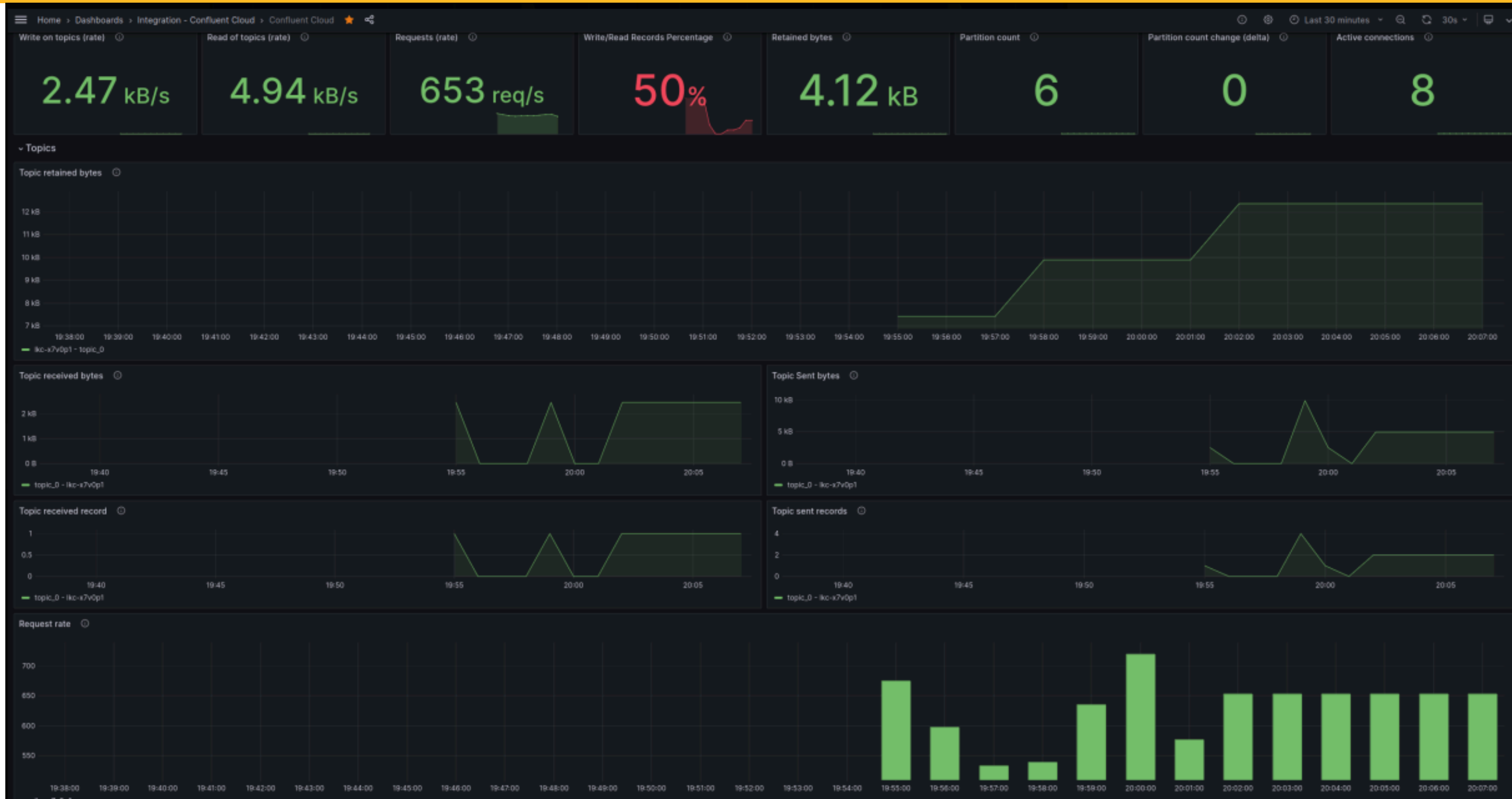
Auto-refresh on

[↓ CSV](#) [↓ JSON](#)

Timestamp 	Offset	Partition	Key	Value	
1702065210639	0	3		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702065138477	3	4		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702064025166	1	2		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702062983996	0	2		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702062106275	2	4		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702061883897	0	0		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	
1702061701362	0	5		{"set_action":[[1700488455.6944273,{"cid":"66","teamid":null,"life":10,"strength":1,"armor":0,"speed":4,"target":"No...	



Métriques du Cluster

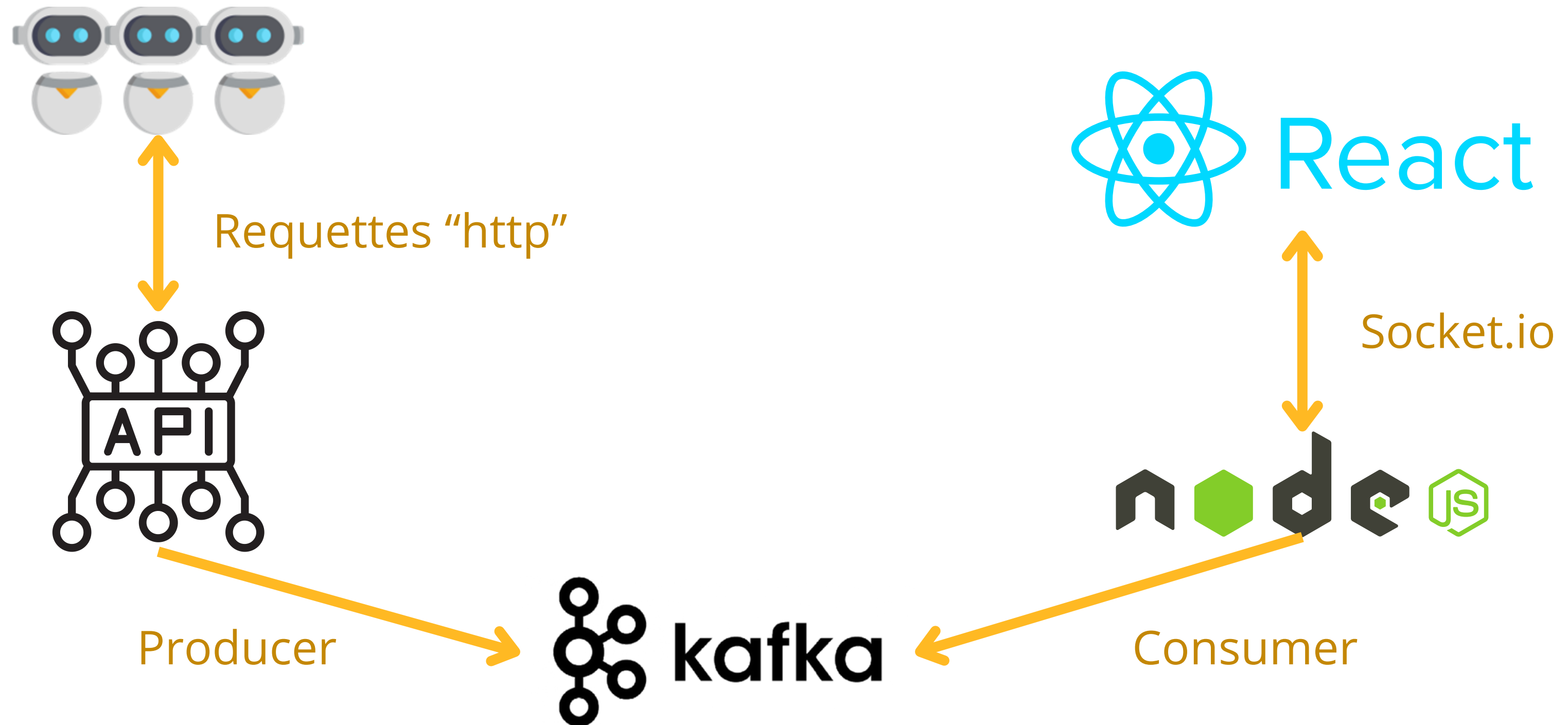




Visualisation

Visualisation de nos données

➤ Rappel de l'architecture



➤ Topics

Général

- Lancement de la partie
- Nouveau tour

Joueur

- Ajout/édition d'un joueur
- Ajout d'une action
- Ajout d'une cible



Page de visualisation

Game Stats

Players

Player	Strength	Armor	Speed	Life	Action	Target
player d042ddbfb	4	5	6	☠	None	
player d283d16c	4	3	6	7♥	BLOCK	player aa648269
player 989dce97	5	6	4	5♥	None	
player 52e720a5	5	5	5	☠	None	
player 06528df5	9	4	4	☠	None	
player aa648269	4	4	4	☠	None	

Turns

Actual turn : 4

1	17:53:45
2	17:53:45
3	17:53:45
4	17:53:45

Actions

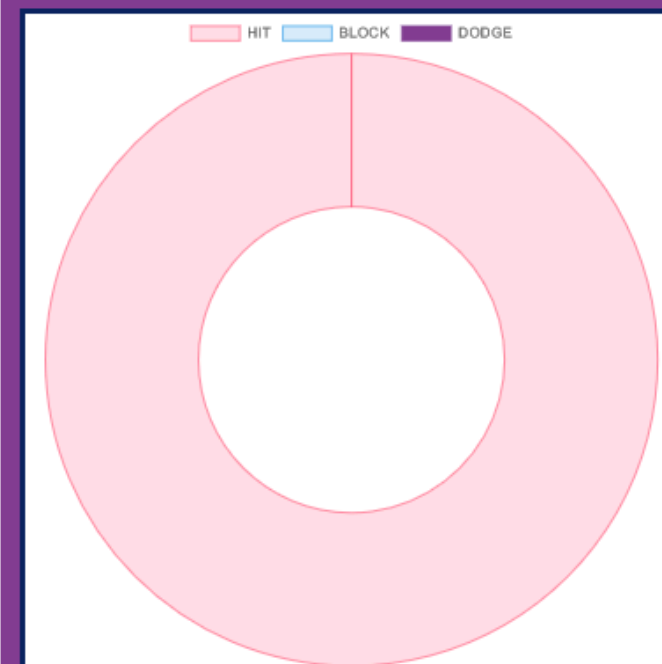
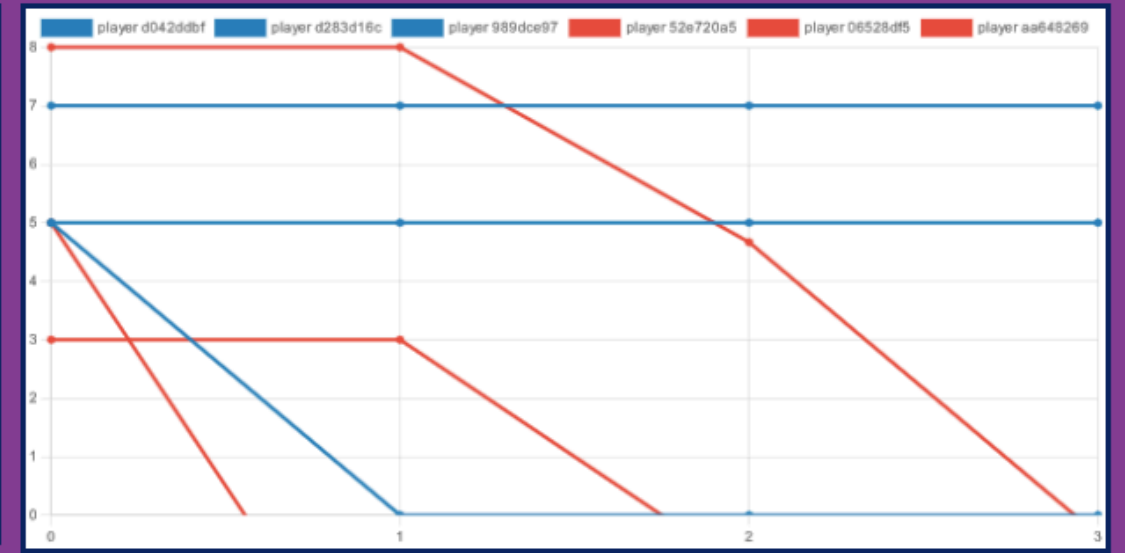
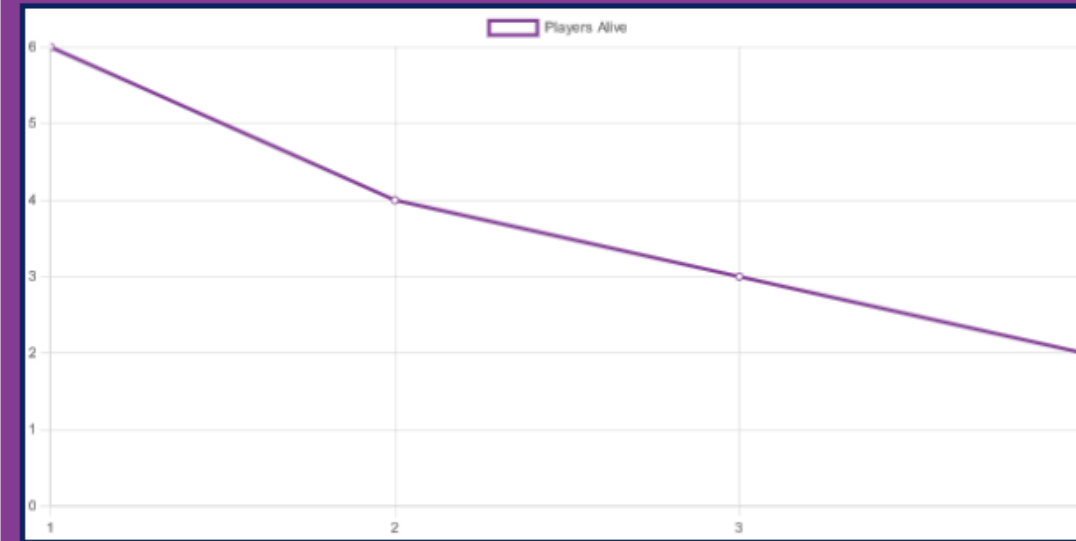
player d042ddbfb 4 player 52e720a5

player d283d16c 0

player 52e720a5 5 player d042ddbfb

player 989dce97 5 player 52e720a5

Game



Damage Dealt	Damage Received	Damage Dodged	Damage Blocked
4	5	0	0



Metric : Les "joueurs"

Players

Player	Strength	Armor	Speed	Life	Action	Target
player d042ddbfb	4	5	6	☠️	None	
player d283d16c	4	3	6	7 ❤️	BLOCK	player aa648269
player 989dce97	5	6	4	5 ❤️	None	
player 52e720a5	5	5	5	☠️	None	
player 06528df5	9	4	4	☠️	None	
player aa648269	4	4	4	☠️	None	

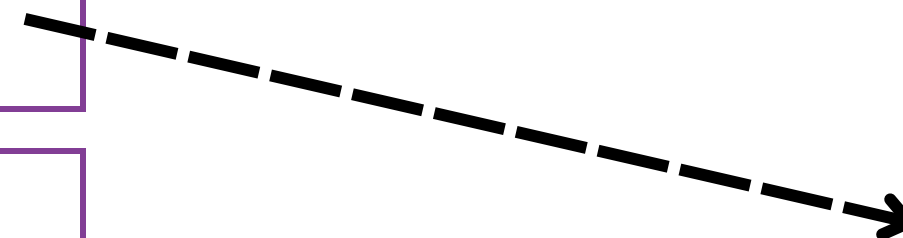


Metrique : Les tours et les actions


Turns

Actual turn : 4

1	17:53:45
2	17:53:45
3	17:53:45
4	17:53:45



Actions

player d042ddb5  4 player 52e720a5

player d283d16c  0

player 52e720a5  5 player d042ddb5

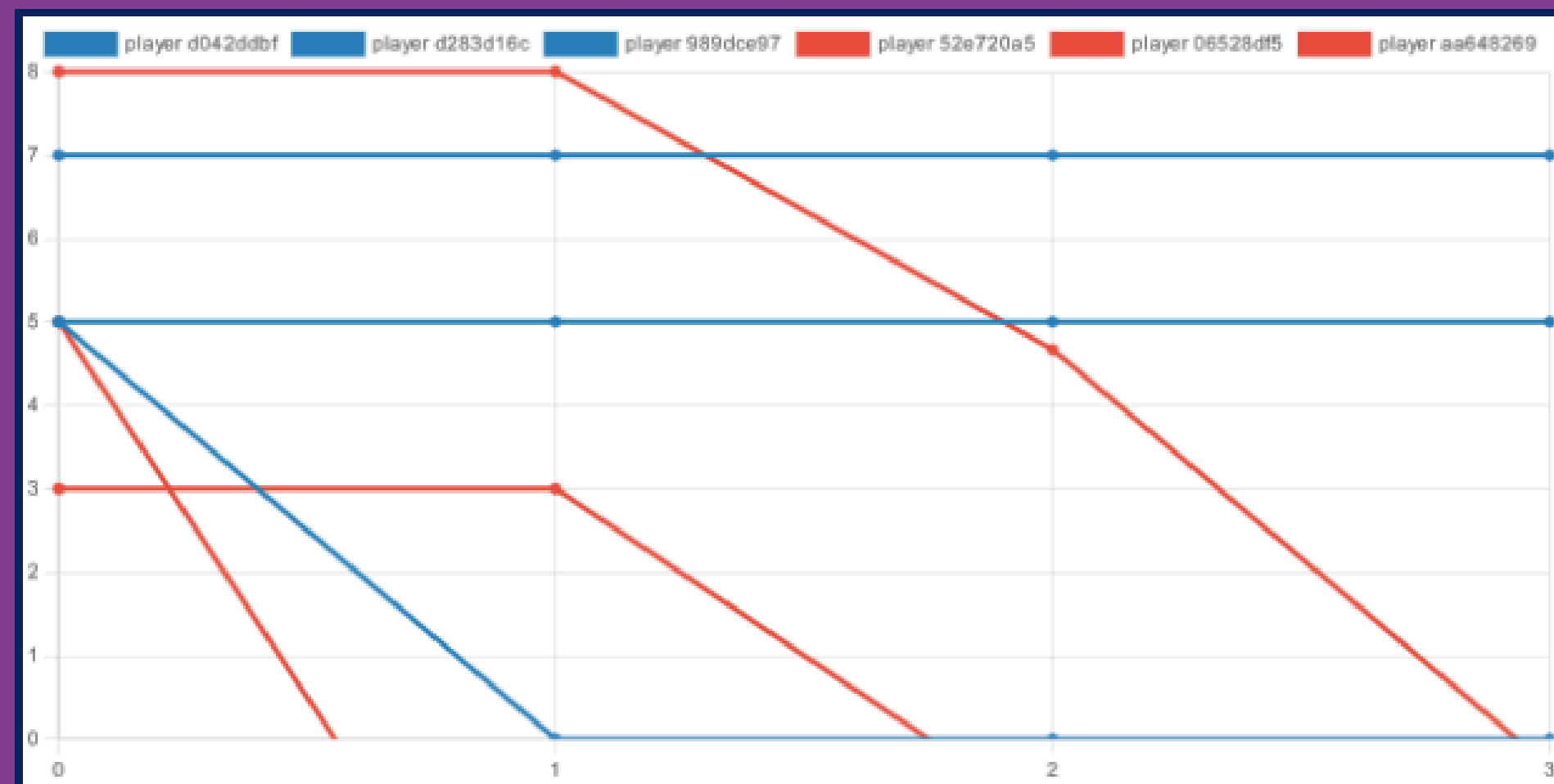
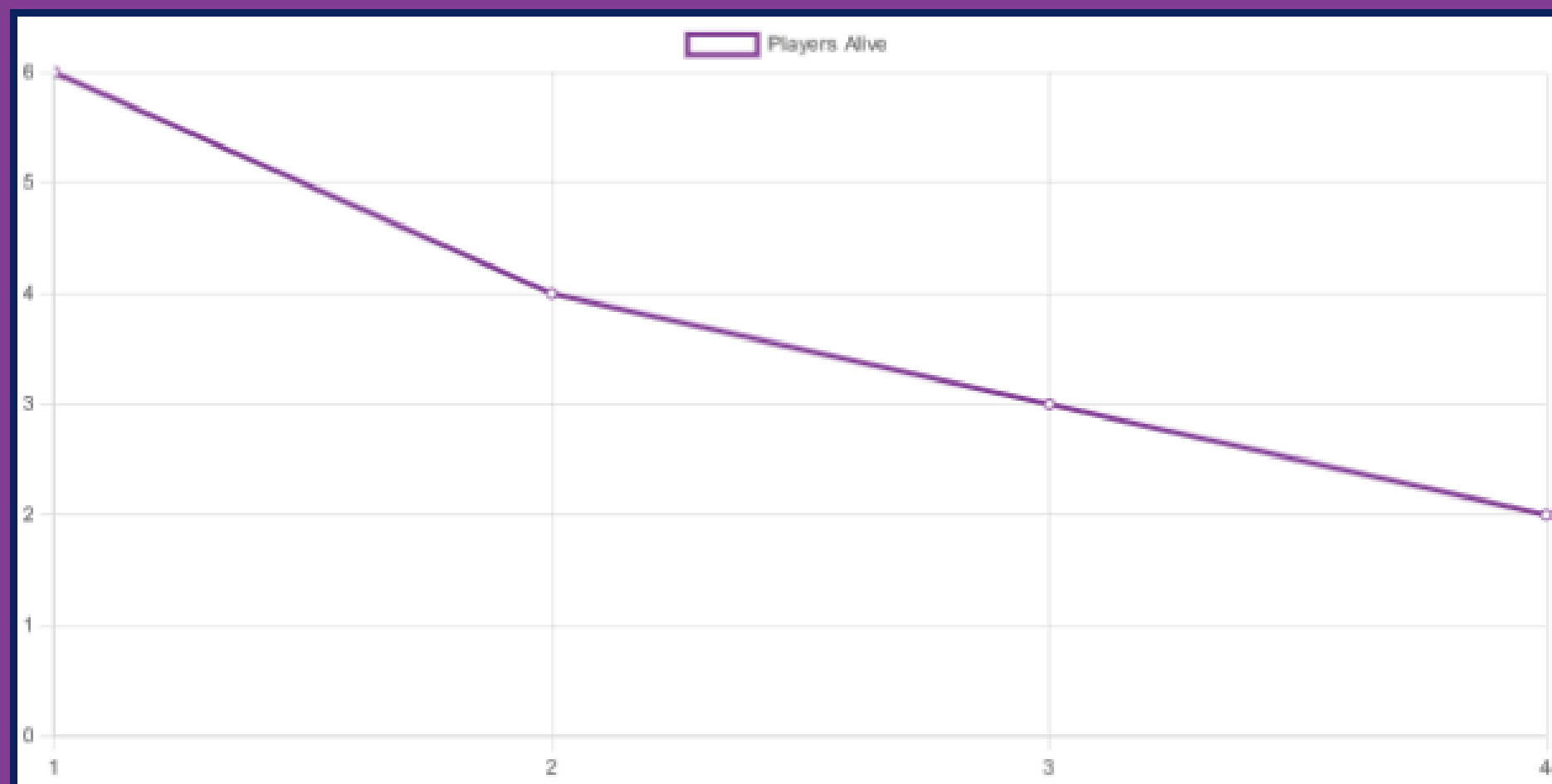
player 989dce97  5 player 52e720a5

player 06528df5  0 player d042ddb5

player aa648269  0

Metrique : Nombres de joueurs en vie & vie

Game

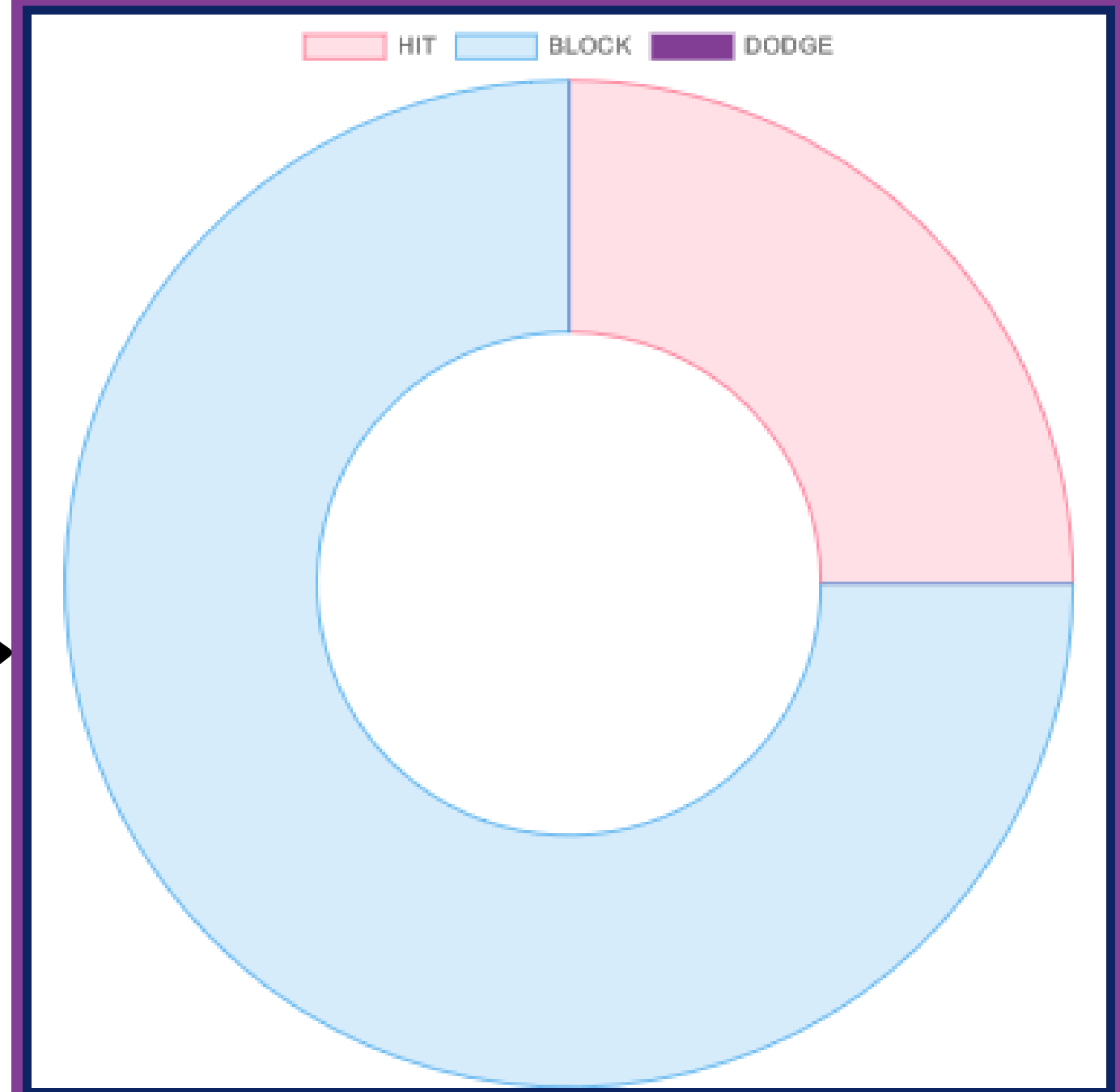




Metric : Les stats des joueurs

Players

Player	Strength	Armor	Speed	Life	Action	Target
player d042ddb	4	5	6	☠	None	
player d283d16c	4	3	6	7♥	BLOCK	player aa648269
player 989dce97	5	6	4	5♥	None	
player 52e720a5	5	5	5	☠	None	
player 06528df5	9	4	4	☠	None	
player aa648269	4	4	4	☠	None	



Damage Dealt	Damage Received	Damage Dodged	Damage Blocked
4	5	0	0

A woman in a grey sweater is looking at a tablet in a meeting room. In the foreground, a laptop displays various business charts and graphs, including a pie chart with a 50% segment highlighted. The text "Démo du jeu" is overlaid on the laptop screen. The background shows other people working at desks with laptops, and a large potted plant is visible on the right. The entire scene is bathed in a warm, yellowish light.

Démo du jeu



**Merci de votre
attention !**

