

```
21013 v [master v]
.gitignore
__init__.py
apprentissage.py
first_tree.pkl
main.py
monequipe.py
README.md
soccertools.py
strategies.py
test_interact.pkl

5 def staticMove(d): # Debug
6     return Vector2D(0, 0)
7 def randomMove(d): # Debug
8     return Vector2D.create_random()
9 ▼ def runnerMove(d): # Course directe vers la balle
10     a = d.goToBa()
11     a = d.maximizeMove(a)
12     return a
13 ▼ def defendMove(d): # Placement defensif
14     a = d.goToMi()
15     a = d.maximizeMove(a)
16     return a
17 ▼ def keeperMove(d): # Rester dans les cages
18     a = d.goToHo()
19     a = d.maximizeMove(a)
20     return a
21 ▼ def attackMove(d): # Offensive directe vers les cages ennemies
22     a = d.goToGo()
23     a = d.maximizeMove(a)
24     return a
25 ▼ def intercMove(d): # Trajectoire optimale d'interception
26     a = d.goTo(d.interBall())
27     a = d.maximizeMove(a)
28     return a
29 ▼ def supplyMove(d): # Placement demarque
30     pass # TODO Demarquer
31
32 # Actions basiques
33 def staticShot(d): # Debug
34     return Vector2D(0, 0)
35 def randomShot(d): # Debug
36     return Vector2D.create_random()
37 ▼ def directShot(d): # Tir direct vers le but ennemi
38     t = d.goToGo()
39     d.maximizeShot(t)
40     return t
41 ▼ def passerShot(d): # Passe a un allie
42     pass # TODO Passer
43
44 ▼ class GlobalStrategy(SoccerStrategy):
45 ▼     def __init__(self, name = None, moves = None, shots = None):
46         if not name:
47             self.name = "GlobalStrategy"
48         else:
49             self.name = name
50             self.moves = moves
51             self.shots = shots
52     def move(self, d): # Choisir le deplacement optimal

4 ▼ def __init__(self, player, state, teamid):
5     self.player = player
6     self.state = state
7     self.ball = state.ball
8     self.goal = []
9     self.team = []
10 ▼     if(teamid == 1):
11         self.goal.append(state.get_goal_center(1))
12         self.goal.append(state.get_goal_center(2))
13         self.team.append(state.team1)
14         self.team.append(state.team2)
15         self.side = 0
16 ▼     else:
17         self.goal.append(state.get_goal_center(2))
18         self.goal.append(state.get_goal_center(1))
19         self.team.append(state.team2)
20         self.team.append(state.team1)
21         self.side = 1
22
23 ▼ def maximize(self, vector, norm = 0): # Maximiser la norme d'un vecteur
24     v = vector
25 ▼     if (v.norm != 0):
26         while(v.norm < norm):
27             v.product(10)
28     return v
29
30 ▼ def maximizeShot(self, vector): # Maximiser le tir
31     return self.maximize(vector, maxPlayerShoot)
32
33 ▼ def maximizeMove(self, vector): # Maximiser le mouvement
34     return self.maximize(vector, maxBallAcceleration)
35
36 ▼ def goTo(self, pos): # Aller a une position
37     return pos-self.player.position
38
39 ▼ def goToGo(self): # Aller vers le but ennemi
40     return self.goTo(self.goal[1])
41
42 ▼ def goToHo(self): # Aller vers le but allie
43     return self.goTo(self.goal[0])
44
45 ▼ def goToBa(self): # Aller vers la balle
46     return self.goTo(self.ball.position)
47
48 ▼ def goToMi(self):
49     v = self.ball.position + self.goal[0]
50     v.product(0.5)
51     return self.goTo(v)
```

Football et stratégie

Un projet d'intelligence artificielle

- Réaliser une intelligence artificielle pour joueur de football
- Découvrir le langage Python
- Utilisation de Git

Principe

SoccerState

- Equipes
- Joueurs
- Balle

SoccerPlayer

- Nom
- Identifiant
- Position
- Vitesse
- **Stratégie**

SoccerStrategy

- **Fonctions Outils**
- **Tactiques élémentaires**
- **Décision**

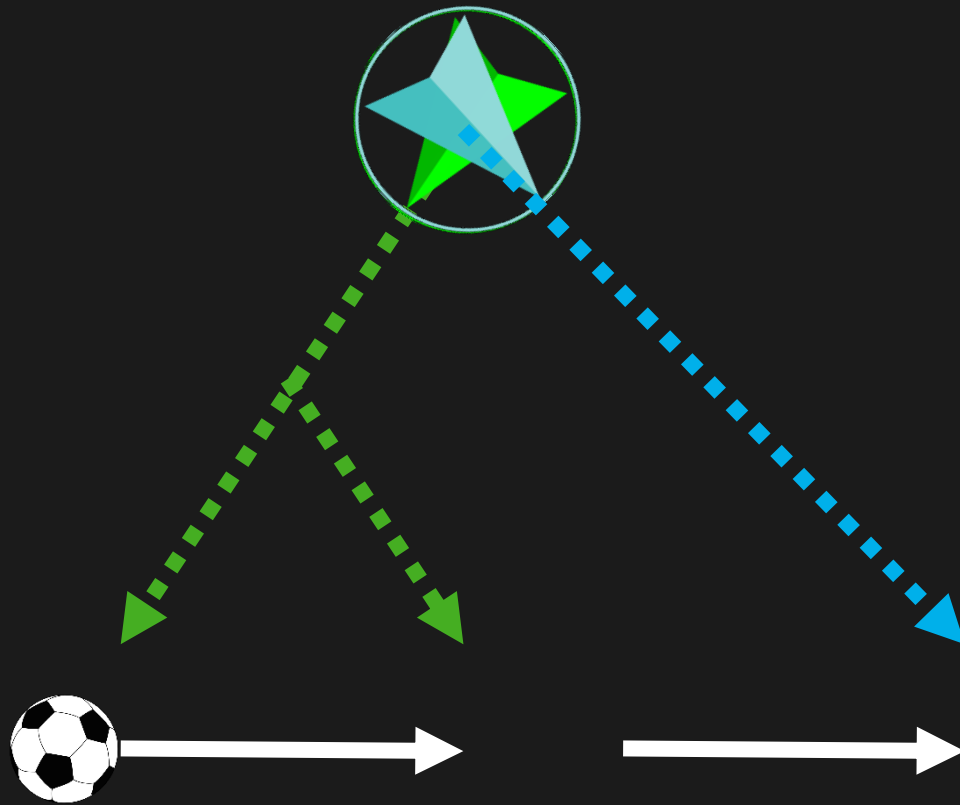
SoccerAction

- Vecteur de **déplacement**
- Vecteur de **tir**

Tactiques élémentaires : coureur, défenseur...

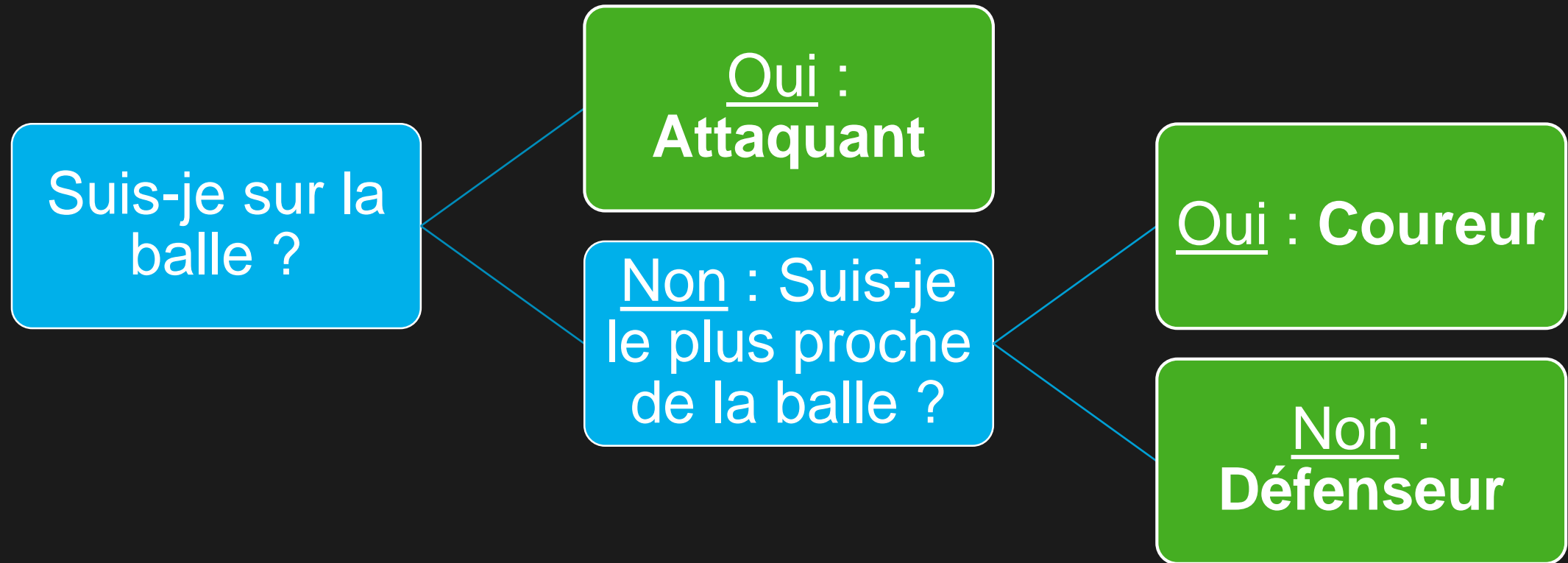


Tactique améliorée : intercepteur

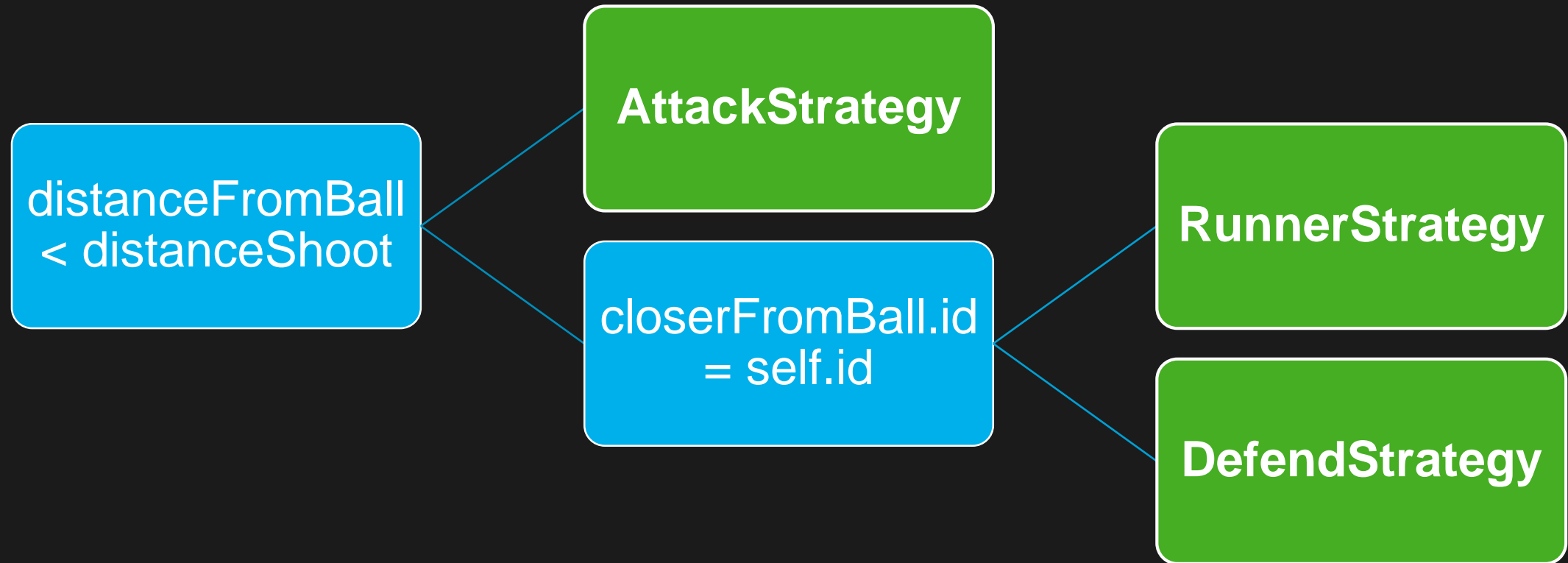


- Coureur : Suivre la balle
- Trajectoire en *courbe du chien*
- Intercepteur : Prévoir le mouvement de la balle
- Trajectoire directe optimale ?

Arbre de décision



Arbre de décision



Expérience et Conclusion

- Gestion du temps et du code dans un projet
- Une stratégie complexe ne vaut pas une tactique bien choisie
- Intelligences artificielles à apprentissage
- Langage Python