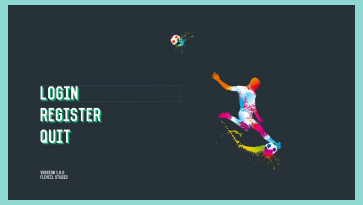
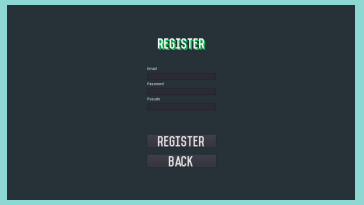
FOOTBALL MANAGER

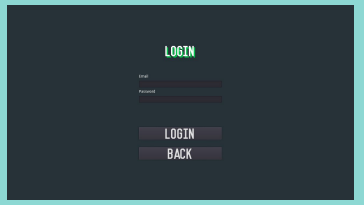
Léveil Florian
B3A YNOV-INFORMATIQUE
Développement Logiciel

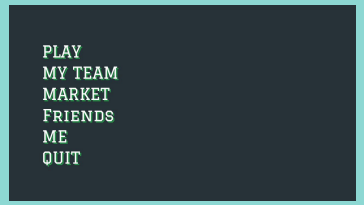
SOMMAIRE

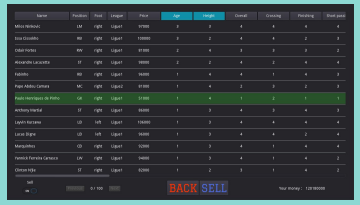
PRÉSENTATION DU PROJET

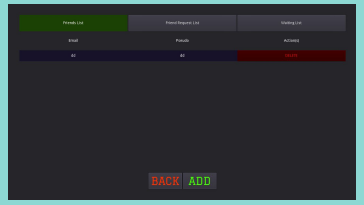


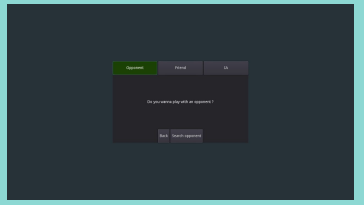








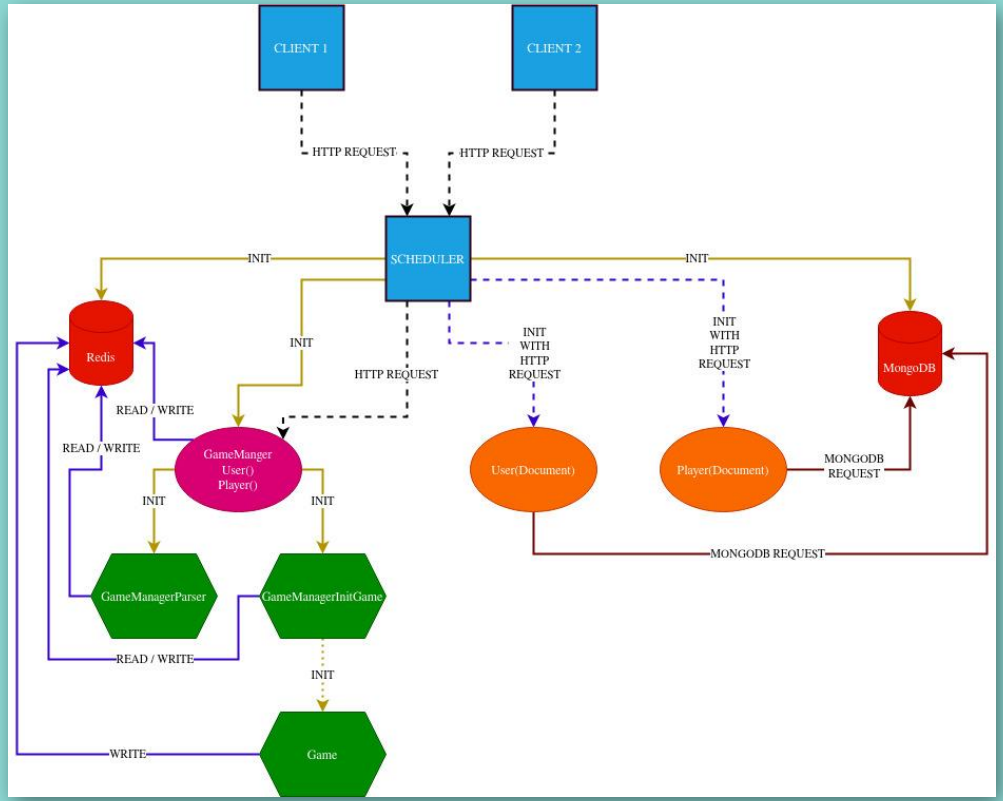




TECHNOLOGIES UTILISÉES

- Python 3.9
- Godot (Godot Script)
- Flask `Flask`
- JWT Flask `Flask-JWT-Extended`
- MongoEngine `flask-mongoengine`
- Redis `redis`

ARCHITECTURE



ALGORITHME

Calcul de distance de passe

Pour une courte passe:	Pour une longue passe:	Pour une passe moyenne:
percent_max = 45	percent_max = 25	100 - courte passe - longue passe
total_weight = 17	total_weight = 24	

```
for key, weight in weight_stat.items():
    _player_stat = getattr(player_with_ball, key)
    _value_to_return += (_player_stat * weight)

_to_return = int((_value_to_return * percent_max) / (total_weight * 4))
```

Calcul du succès de l'action

```
for key, weight in weight_stat.items():
    _player_stat = getattr(player_with_ball, key)
    _value_to_return += (_player_stat * weight)

_to_return = (_value_to_return * 100) / (total_weight * 4)

return int(50 + (attack_power - defense_power))
```

```
class WEIGHT_LONG_PASS(object):
    def __init__(self):
        self.VISION = 7
        self.STRENGTH = 10
        self.LONG_PASSING = 7
```

```
class WEIGHT_DEFENSE_STAT(object):
    def __init__(self):
        self.AGE = 10
        self.HEIGHT = 10
        self.OVERALL = 10
        self.AGILITY = 12
        self.DRIBBLING = 14
        self.INTERCEPTIONS = 14
        self.MARKING = 14
        self.STANDING_TACKLE = 14
        self.SLIDING_TACKLE = 14
```

DEMO

AMÉLIORATIONS

CONCLUSION