Brief:

1. Analyser et modifier. Sur le package `part1` il faudra comprendre et adapter/corriger.
2. Apprendre et imiter. Sur le package `part2` il vous faudra produire des classes simple et un peu d'algorithm.
3. Prise en main. Sur le package `part3` créez des armes et votre armee, preparer vous a la bataille.
4. Tournois entre armee.

Toutes les parties sont fournis avec des test unitaire simple, mais pas forcément complets, le rendu pourra être testé avec des tests plus complexes.

Part1

See README.md in part1

Part2

See README.md in part2

Part 3

See README.md in part3

Lord of the Ring: Battle of the 3 Army

> You can find the Tournament Code at git@github.com:lumy/battle3army-tournament.git the code will be accessible when part3 is done. The possible Soldiers are the one we developped, but we never wrote any archer, the magician always heals in second position. While you're doing your `preparation` function you are allowed to clone the tournament code (test your army) by creating a new branch and doing a MergeRequest. No Pull Request will be accepted if submitted after the day before the deadline.

Let's test our armies against each other.

Create a new package `<name>_war` (eg. `lumy_war romain_war`) Inside create a file `preparation.py` with:

- a Global Variable: `RACE`. It should be equal to one of `dwarfs elves humans` that is how you are gonna choose your army.

- function `preparation(TrainingCamp)`: It should return an Instance of `Army` (of your chosen race)

Your function `preparation` will prepare your army, (TrainingCamp holds data about how many warriors can be recruited, and training time data). Some weapons will be available at the training Camp.

Some possible algorithms: - choose a race and create the `Army` instance - recruit all available warriors - train a warrior to warlord - if enough time : train 2 healers, 2 specialized units. - if still enough time: train between vampire and knight. - Equip weapons to units with the least `attack` attribute. - return army

- The Tournament will take place like this :
  - You meet every other player for two battles with celerity reversed.
  - in case of a draw, a last match will be done in StraightFight[1] mode.
  - Last opponent alive wins a point, the one with most point in then end win a TournamentPoint.
  - Bonus TournamentPoint if code respects all specifications.
  - TournamentPoint lost if the code deviates from specifications.
  - Opponent with maximum TournamentPoint at the end of tournament wins.
  - In the end, in case of draw between serveral players : The battles are replayed without the possibility of creating a warlord, in this case no time will be consumed and the function will return your Unit.

Notes: during the battle we will only use and verify your `preparation.py` file. All the other files will be provided by us. But the structure we got at the end of part3 will be the same: `part3.battle part3.dwars part3.fighter ...` Here you will have to think about an army able to win the most points.

1. Straight fight: armies are lined up in battle: All fights occur at the same time army1[0] vs army2[0] army1[1] vs army2[1] . . . When someone Dies his oponnent goes up in the list. army1[0] vs army2[0] army1[2], army1[1] vs army2[1] army1[3] vs army2[3] . . .