

Alex Farrell

Mario Vega

Individual/Pair Project Description of data

Description of data:

Pokemon

MAX_LEVEL:	[const] int	[100]
GROUP_TOTAL:	[const] int	[2]
name:	string	
level:	int	
stat_groups:	<dictionary> float	['Physical', 'Attack', 'Defense']
total_stat:	float	[All stats added up]

Trainer

MAX_SKILL:	[const] int	[10]
MVP_VARIABILITY:	[const] int	[10]
MAX_POKEMON:	[const] int	[6]
name:	string	
skill_level:	int	
pokemon_team:	<list> Pokemon	
team_stat_groups:	<dictionary> float	['Physical', 'Attack', 'Defense']

Battle

STAT_VARIABILITY:	[const] int	[5]
challenger:	Trainer	
defender:	Trainer	
battle_won:	bool	[True, False]
mvp_name:	string	

Arena

name:	string	
challenger:	Trainer	
defenders:	<list> Trainer	
arena_conquered:	bool	[True, False]

Operations:

Pokemon

initializer	[name, level, hp, speed, attack, sp_attack, defense, sp_defense]
calculate_stat_groups()	
calculate_total_stat()	
setup()	[Runs previous two methods]
get_name()	[Returns]
get_stat_groups()	[Returns]
get_total_stat	[Returns]

Trainer

initializer	[name, skill_level]
add_pokemon(Pokemon pokemon)	
calculate_team_stat_groups()	
calculate_team_mvp()	[Returns]
get_team_stat_groups()	[Returns]

Battle

initializer	[challenger, defender]
add_variability_to_stats()	
compare_stats()	
determine_mvp()	
display_results()	
battle()	[Runs previous four methods, returns True if challenger won, False if they did not]

Arena

initializer	[name]
set_challenger(Trainer challenger)	
add_defender(Trainer defender)	
battle_next_defender()	
get_status()	[Returns]
exit_arena(bool arena_conquered)	[Calls sys.exit()]

Simulation Outline:

```
poke1 = Pokemon('Alexmon', ...[Overpowered stats]...)
```

```
poke1.setup()
```

```
poke2 = Pokemon('Scrum', ...[Stats are tight I guess]...)
```

```
poke2.setup()
```

```
poke3 = Pokemon('Agile', ...[No documentation, no stats]...)
```

```
poke3.setup()
```

```
poke4 = Pokemon('Rallycat', ...[You gonna get wrecked]...)
```

```
poke4.setup()
```

```
poke5 = Pokemon('Trashmon', ...[Literally trash]...)
```

```
poke5.setup()
```

```
poke6 = Pokemon('Bob Marley', ...[God mode]...)
```

```
poke6.setup()
```

```
Mario = Trainer('Mario', Maximum skills baby)
```

```
Mario.add_pokemon(poke1)
```

```
Mario.add_pokemon(poke2)
```

```
Mario.calculate_team_stat_groups()
```

```
Jhibby = Trainer('JHibbs420BlazeGodAbsoluteSteezWizard, ...[yes]...')
```

```
Jhibby.add_pokemon(poke3)
```

```
Jhibby.add_pokemon(poke4)
```

```
Jhibby.calculate_team_stat_groups()
```

```
Third = Trainer('Third_rate_trainer_no_one_cares_about', ...[.]...)
```

```
Third.add_pokemon(poke5)
```

```
Third.add_pokemon(poke6)
```

```
Third.calculate_team_stat_groups()
```

```
arena = Arena('The Pwn Zone')
```

```
arena.set_challenger(Mario)
```

```
arena.add_defender(Jhibby)
```

```
arena.add_defender(Third)
```

```
for battle in range(arena.get_defenders):
```

```
    arena.battle_next_defender()                # This method initiates the Battle class and
                                                # calls all methods for the battle
```

```
    if arena.get_status() == False:              # Challenger lost
```

```
        arena.exit_arena(False)
```

arena.exit_arena(True)

Challenger won