Mario Vega

Individual/Pair Project Class and Method Descriptions for Mario

Pokemon:

Creates a pokemon and holds its characteristics [name, level] and calculated stat groups [Physical Stats, Attack Stats, Defense Stats]. The pokemon's raw stats [hp, speed, attack, special attack, defense, special defense] are passed to the constructor and used to calculate the stat groups. Should ensure level passed to constructor is no larger than MAX_LEVEL [100].

Methods:

---

calculate_stat_groups():

Physical:

(hp + speed) / GROUP_TOTAL

Attack:

(attack + sp_attack) / GROUP_TOTAL

Defense:

(defense + sp_defense) / GROUP_TOTAL

Multiple each by:

level / MAX_LEVEL

ex:

Physical * (level / MAX_LEVEL)

Save each in its associated key in the dictionary for stat_groups

---

calculate_total_stat():

Add up all calculated stat groups into one large total

ex:

total_stat = stat_groups[‘Physical’] + stat_groups[‘Attack’] +

stat_groups[‘Defense’]

---

setup():

calculate_stat_groups()

calculate_total_stat()

---

getters():

You know, like get the stuff, eh?

Trainer:

Creates trainer who has multiple pokemon [A max of 6] and holds their characteristics[name, skill_level]. Should ensure skill_level passed to constructor is no larger than MAX_SKILL [10].

Methods:

---

add_pokemon(pokemon):

Check if pokemon_team list length is less than MAX_POKEMON [6],

if it isn't:

Raise error with message explaining it couldn't be added

append pokemon to pokemon_team list

---

calculate_team_stat_groups():

Add each stat group for all pokemon together

Total_Physical:

poke1_physical + … + poken_physical

Total_Attack:

poke1_attack + … + poken_attack

Total_Defense:

poke1_defense + … + poken_defense

Divide each by:

MAX_SKILL - skill_level

ex:

Total_Physical / (MAX_SKILL - skill_level)

Save each in its associated key in the dictionary for team_stat_groups

---

calculate_team_mvp():

Loop through pokemon_team list:

Get total_stat from pokemon

Multiply each by:

(Randomly Generated Number) /MVP_VARIABILITY

ex:

poke1_total * (random_num / MVP_VARIABILITY)

Get the largest result, set the name of that pokemon as the mvp and return it

---