

Initial_Game_Balance

December 5, 2021

```
[2]: import numpy as np
import matplotlib.pyplot as plt

[9]: # data
levels = np.array([1,2,3,4,5,6,7,8])
total_health = np.array([20,25,30,35,40,45,50,55])
thing = np.array([7.639320218310988,
9.549150272888735,
11.458980327466481,
13.368810382044229,
15.278640436621975,
17.188470491199723,
19.09830054577747,
21.008130600355216])

plt.xlabel("Level")
plt.plot(levels, total_health, label="Total health")
plt.plot(levels, thing, label="Theoretical gold reward")

def progression(ratio):
    # init
    pu = 0
    gold = 0
    dps_per_lvl = []
    # simulate
    for i in range(len(levels)):

        dps = 0.8 * 2 * (pu+1) * 2
        dps_per_lvl.append(dps)

        gold += total_health[i] * ratio

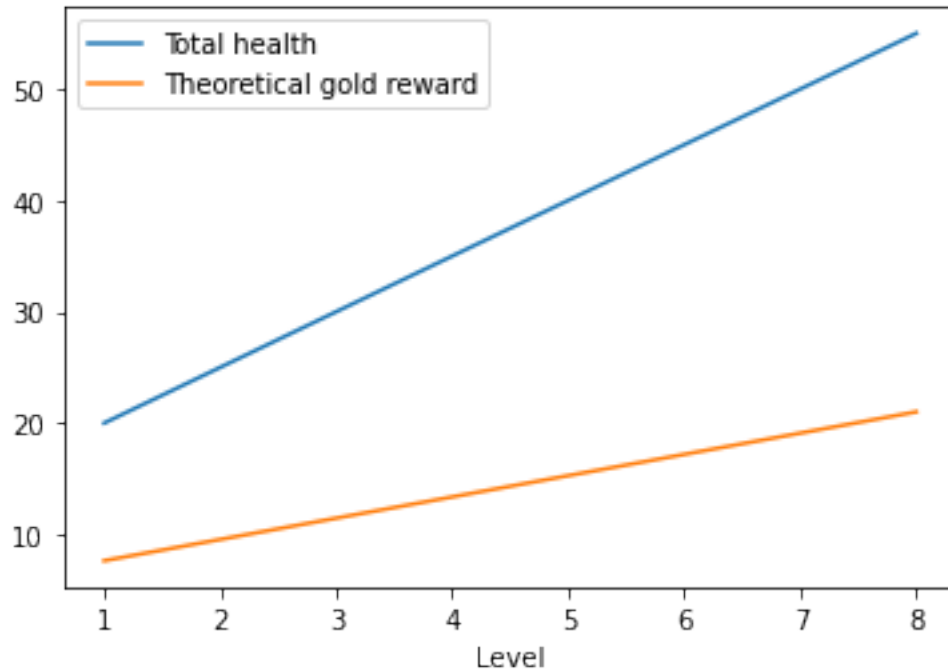
    npu = gold // 5
    gold -= npu * 5
    pu += npu
```

```

return dps_per_lvl

#plt.plot(damage_dealt, label="dmg")
plt.legend()
plt.show()

```



```

[6]: def objective(reward):
    damage_dealt = progression(reward)
    return sum(pow(total_health-damage_dealt,2))

domain = np.arange(0, 3, 0.1)
objective_landscape = [objective(r) for r in domain]
print(objective_landscape)
print(objective(1.0))

plt.xlabel("Reward Modifier")
plt.plot(domain, objective_landscape, label="obj")
plt.show()

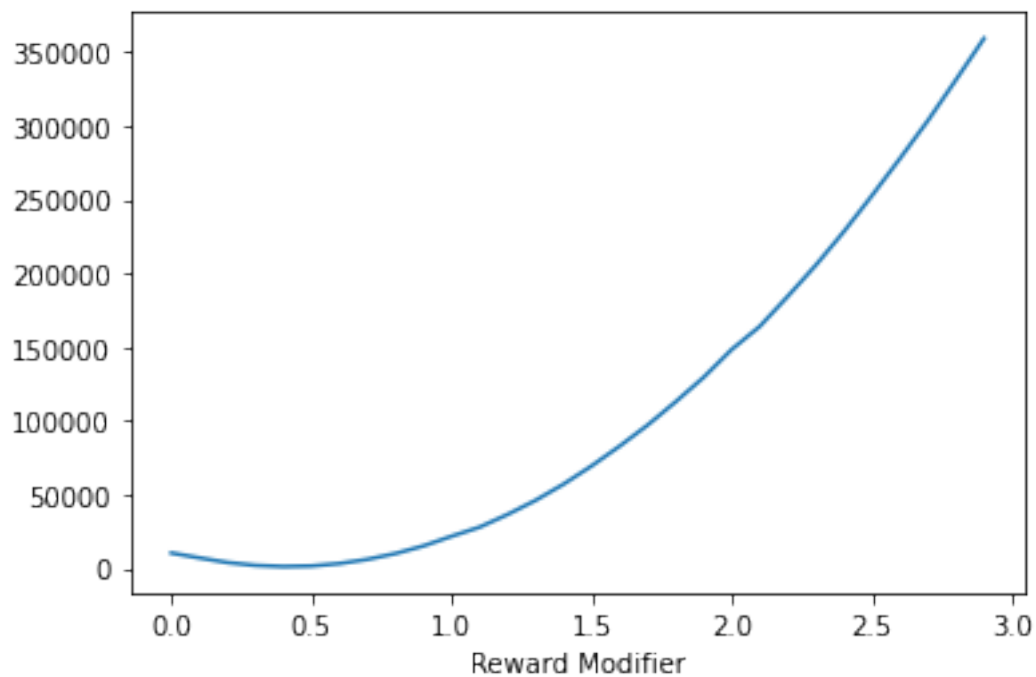
```

```

[10461.919999999998, 7159.52, 4018.3999999999996, 2067.6800000000003, 1302.24,
1665.76, 3316.9600000000001, 6164.960000000001, 10103.52, 15265.76, 21805.28,
27964.640000000003, 36579.040000000001, 46158.560000000001, 57148.64,
69492.960000000002, 82899.680000000002, 97196.0, 112972.00000000001,
129582.56000000003, 148410.08000000002, 164031.20000000007, 184401.12,
205510.88, 228256.48000000004, 252581.60000000003, 277743.84000000001,

```

303488.48000000001, 331101.92000000004, 359160.80000000005]
21805.28



```
[7]: from scipy import optimize

# return sum(pow(enemy_health-damage_dealt,2))

result = optimize.minimize_scalar(objective) #, method='brent', bounds=[0,10])
print(result.success) # check if solver was successful
print(result.x)
list(map(lambda y : result.x * y, total_health))
```

True
0.3819660109155494

```
[7]: [7.639320218310988,
      9.549150272888735,
      11.458980327466481,
      13.368810382044229,
      15.278640436621975,
      17.188470491199723,
      19.09830054577747,
      21.008130600355216]
```

```
[ ]:
```