DS TB November 2021 Report

November 28, 2021

1 GSF Sigma DS Geo TB Report November 2021

1.1 Load Data

```python
import pandas as pd
import math
file_paths = ['phase_1.csv','phase_2.csv','phase_3.csv','phase_4.csv']
data = []

for i in range(len(file_paths)):
        try:
                data.append(pd.read_csv(file_paths[i], index_col=0).
 ↪drop_duplicates().fillna(0))
        except:
                continue
```

```python
if len(data) >= 2:
    for i in range(len(data)-1,0,-1):
        data[i]['territoryPointsContributed'] =␣
 ↪data[i]['territoryPointsContributed'] - \
        data[i-1]['territoryPointsContributed']

tot = pd.Series([0]*len(data[-1]), index=data[-1].index)
for phase in data:
    phase['pointsPerGP'] = round(phase['territoryPointsContributed'] /
                                 (phase['shipGP']+phase['characterGP']), 3)
    tot += phase['pointsPerGP']
avgPointsPerGP = round(tot/len(data), 3)
```

1.2 Input

1.2.1 Sandbagging

```python
tbType = 'DS'

sbag_1_top = False
sbag_1_mid = False
sbag_1_bottom = False
```

```
sbag_2_top = False
sbag_2_mid = False
sbag_2_bottom = False

sbag_3_top = False
sbag_3_mid = False
sbag_3_bottom = False
```

1.2.2 Shards and Stars

Wat shards: 42

Phase 1	Phase 2	Phase 3	Phase 4
x	3	3	3
3	3	3	1
3	3	3	3

1.3 Calculations

1.3.1 TB Points per CM

LS GEO TB

```python
if (tbType == 'LS'):
    p1_ships_1 = [0,523900]
    p1_ships_2 = [0,0]
    p2_ships_2 = [0,0]
    p3_ships_2 = [0,0]
    p4_ships_2 = [0,0]

    if(sbag_1_top):
        p2_ships_1 = p1_ships_1
    else:
        p2_ships_1 = [0,900000]
    if(sbag_2_top):
        p3_ships_1 = p2_ships_1
    else:
        p3_ships_1 = [0,1800000]
    if(sbag_3_top):
        p4_ships_1 = p3_ships_1
    else:
        p4_ships_1 = [0,2750000]

    p1_ground_1 = [0,403000,573500,840000,1155000]
    p1_ground_2 = [0,403000,573500,840000,1155000]

    if(sbag_1_bottom):
        p2_ground_1 = p1_ground_1
```

```python
            p2_ground_2 = p1_ground_2
        else:
            p2_ground_1 = [0,434000,704000,1014750,1377000]
            p2_ground_2 = [0,434000,704000,1014750,1377000]
        if(sbag_2_bottom):
            p3_ground_1 = p2_ground_1
            p3_ground_2 = p2_ground_2
        else:
            p3_ground_1 = [0,464000,775500,1105000,1627500]
            p3_ground_2 = [0,464000,775500,1105000,1627500]
        if(sbag_3_bottom):
            p4_ground_1 = p3_ground_1
            p4_ground_2 = p3_ground_2
        else:
            p4_ground_1 = [0,511500,867000,1242500,1837500]
            p4_ground_2 = [0,511500,867000,1242500,1837500]

        p1_ground_3 = [0,403000,573500,840000,1155000]
        p1_ground_4 = [0,523900,745550,1092000,1501500]
        p1_ground_5 = [0,0,0,0,0]

        if(sbag_1_mid):
            p2_ground_3 = p1_ground_3
            p2_ground_4 = p1_ground_4
            p2_ground_5 = p1_ground_5
        else:
            p2_ground_3 = [0,434000,704000,1014750,1377000]
            p2_ground_4 = [0,434000,704000,1014750,1377000]
            p2_ground_5 = [0,564200,915200,1319175,1790100]
        if(sbag_2_mid):
            p3_ground_3 = p2_ground_3
            p3_ground_4 = p2_ground_4
            p3_ground_5 = p2_ground_5
        else:
            p3_ground_3 = [0,464000,775500,1105000,1627500]
            p3_ground_4 = [0,464000,775500,1105000,1627500]
            p3_ground_5 = [0,0,0,0,0]
        if(sbag_3_mid):
            p4_ground_3 = p3_ground_3
            p4_ground_4 = p3_ground_4
            p4_ground_5 = p3_ground_5
        else:
            p4_ground_3 = [0,511500,867000,1242500,1837500]
            p4_ground_4 = [0,664950,1127100,1615250,2388750]
            p4_ground_5 = [0,867000,1837500,0,0]
```

DS GEO TB

```python
if (tbType == 'DS'):

    p1_ships_1 = [0,0]
    p1_ships_2 = [0,0]
    p2_ships_1 = [0,825000]
    p2_ships_2 = [0,1072500]

    if(sbag_2_top):
        p3_ships_1 = p2_ships_1
        p3_ships_2 = p2_ships_2
    else:
        p3_ships_1 = [0,1665000]
        p3_ships_2 = [0,2164500]
    if(sbag_3_top):
        p4_ships_1 = p3_ships_1
        p4_ships_2 = p3_ships_2
    else:
        p4_ships_1 = [0,2750000]
        p4_ships_2 = [0,0]

    p1_ground_1 = [0,187500,297500,500000,792000]
    p1_ground_2 = [0,187500,297500,500000,792000]

    if(sbag_1_bottom):
        p2_ground_1 = p1_ground_1
        p2_ground_2 = p1_ground_2
    else:
        p2_ground_1 = [0,270000,420000,708000,1080000]
        p2_ground_2 = [0,270000,420000,708000,1080000]
    if(sbag_2_bottom):
        p3_ground_1 = p2_ground_1
        p3_ground_2 = p2_ground_2
    else:
        p3_ground_1 = [0,336000,540000,910000,1352000]
        p3_ground_2 = [0,336000,540000,910000,1352000]
    if(sbag_3_bottom):
        p4_ground_1 = p3_ground_1
        p4_ground_2 = p3_ground_2
    else:
        p4_ground_1 = [0,405000,675000,1038500,1564000]
        p4_ground_2 = [0,405000,675000,1038500,1564000]

    p1_ground_3 = [0,187500,297500,500000,792000]
    p1_ground_4 = [0,187500,297500,500000,792000]
    p1_ground_5 = [0,0,0,0,0]

    if(sbag_1_mid):
```

```
            p2_ground_3 = p1_ground_3
            p2_ground_4 = p1_ground_4
            p2_ground_5 = p1_ground_5
        else:
            p2_ground_3 = [0,270000,420000,708000,1080000]
            p2_ground_4 = [0,270000,420000,708000,1080000]
            p2_ground_5 = [0,351000,546000,920400,1404000]
        if(sbag_2_mid):
            p3_ground_3 = p2_ground_3
            p3_ground_4 = p2_ground_4
            p3_ground_5 = p2_ground_5
        else:
            p3_ground_3 = [0,336000,540000,910000,1352000]
            p3_ground_4 = [0,336000,540000,910000,1352000]
            p3_ground_5 = [0,0,0,0,0]
        if(sbag_3_mid):
            p4_ground_3 = p3_ground_3
            p4_ground_4 = p3_ground_4
            p4_ground_5 = p3_ground_5
        else:
            p4_ground_3 = [0,405000,675000,1038500,1564000]
            p4_ground_4 = [0,405000,675000,1038500,1564000]
            p4_ground_5 = [0,1350050,2033200,0,0]
```

1.3.2 CM Points

```
[ ]: global ground_missions
     ground_missions = {}
     ground_missions[1] = [p1_ground_1, p1_ground_2,
                           p1_ground_3, p1_ground_4, p1_ground_5]
     ground_missions[2] = [p2_ground_1, p2_ground_2,
                           p2_ground_3, p2_ground_4, p2_ground_5]
     ground_missions[3] = [p3_ground_1, p3_ground_2,
                           p3_ground_3, p3_ground_4, p3_ground_5]
     ground_missions[4] = [p4_ground_1, p4_ground_2,
                           p4_ground_3, p4_ground_4, p4_ground_5]

     global ship_missions
     ship_missions = {}
     ship_missions[1] = [p1_ships_1, p1_ships_2]
     ship_missions[2] = [p2_ships_1, p2_ships_2]
     ship_missions[3] = [p3_ships_1, p3_ships_2]
     ship_missions[4] = [p4_ships_1, p4_ships_2]

     global max_ground
     max_ground = {}
```

```
max_ground[1] =␣
 ↪p1_ground_1[4]+p1_ground_2[4]+p1_ground_3[4]+p1_ground_4[4]+p1_ground_5[4]
max_ground[2] = p2_ground_1[4]+p2_ground_2[4] + \
    p2_ground_3[4]+p2_ground_4[4]+p2_ground_5[4]
max_ground[3] =␣
 ↪p3_ground_1[4]+p3_ground_2[4]+p3_ground_3[4]+p3_ground_4[4]+p3_ground_5[4]
max_ground[4] = p4_ground_1[4]+p4_ground_2[4] + \
    p4_ground_3[4]+p4_ground_4[4]+p4_ground_5[2]

global max_ships
max_ships = {}
max_ships[1] = p1_ships_1[1]+p1_ships_2[1]
max_ships[2] = p2_ships_1[1]+p2_ships_2[1]
max_ships[3] = p3_ships_1[1]+p3_ships_2[1]
max_ships[4] = p4_ships_1[1]+p4_ships_2[1]
```

1.4 Low Performers

1.4.1 Lowest TB Points per GP

```
[ ]: n = 10
     low_ppg = data[-1]['pointsPerGP'].sort_values().head(n)
     print(low_ppg)
```

```
name
SloppySaberFlavor     0.912
Neeb                  0.916
Maxaron Lexilon       0.953
M1TTH                 1.022
Zhil Axfow            1.182
Exeel                 1.215
Baxston Kane          1.266
Argarax               1.455
Dark Penguin          1.592
EvilCoyote2011        1.650
Name: pointsPerGP, dtype: float64
```

1.4.2 Lowest CM Waves Completed

```
[ ]: if len(data) >= 2:
         for i in range(len(data)-1, 0, -1):
             data[i]['combatMissionWavesCompleted'] =␣
     ↪data[i]['combatMissionWavesCompleted'] - \
                 data[i-1]['combatMissionWavesCompleted']

     low_cm = data[-1]['combatMissionWavesCompleted'].sort_values().head(n)
     print(low_cm)
```

```
name
EvilCoyote2011       5
Neeb                 5
SloppySaberFlavor    5
M1TTH                6
BabyYodaHitta        6
Higgs                6
Theflavorgreen       6
Maxaron Lexilon      7
Dark Penguin         7
wamakima5004         7
Name: combatMissionWavesCompleted, dtype: int64
```

1.4.3 Lowest TB Points

```
[ ]:   low_tb = data[-1]['territoryPointsContributed'].sort_values().head(n)
       print(low_tb)
```

```
name
M1TTH                4595551
Neeb                 5411443
SloppySaberFlavor    5501819
BabyYodaHitta        6023473
Maxaron Lexilon      6373836
Larping Soccer Moms  6953759
Dark Penguin         6998884
Zhil Axfow           7560266
Exeel                7993018
Argarax              8221167
Name: territoryPointsContributed, dtype: int64
```

1.5 Top Performers

1.5.1 Highest TB Points per GP

```
[ ]:   high_ppg = data[-1]['pointsPerGP'].sort_values(ascending = False).head(n)
       print(high_ppg)
```

```
name
MINI xipokemastrix   2.415
MINI Stewabob        2.406
TacoPizza            2.371
joker                2.339
Loadage              2.317
MarkKenoburger       2.271
ilekkund2            2.244
Obi Won Sebroni      2.225
Plucky Haydon        2.199
Zlada14              2.110
```

```
Name: pointsPerGP, dtype: float64
```

1.5.2 Highest Combat Waves Completed

```python
high_cm = data[-1]['combatMissionWavesCompleted'].sort_values(
    ascending=False).head(n)
print(high_cm)
```

```
name
s o l o          17
ilekkund         16
OttoVonGens      15
TacoPizza        14
Gryphix          14
Plucky Haydon    14
Guntha Arbos     14
Loadage          14
Tommyboy85       13
joker            13
Name: combatMissionWavesCompleted, dtype: int64
```

1.5.3 Highest TB Points

```python
high_tb = data[-1]['territoryPointsContributed'].sort_values(
    ascending=False).head(n)
print(high_tb)
```

```
name
s o l o           16819750
ilekkund          15727362
OttoVonGens       15101128
Loadage           14426781
Gryphix           14199109
Plucky Haydon     13519169
Tommyboy85        13441711
ShootMeow         13360319
Zlada14           12853463
Masajj Vemtits    12778610
Name: territoryPointsContributed, dtype: int64
```

1.6 Guild Performance

```python
def toPoints(points,waves):
    i = 0
    point_value = pd.Series([0]*len(waves),index = waves.index)
    while(i<len(waves)):
        point_value[i] = points[waves.iloc[i].astype('int64')]
        i+=1
```

```
      return point_value
```

```python
def percents(df,phase):
        if(not (("Ch 5") in df.columns)):
                df['Ch 5'] = [0]*len(df)

        if(not (("Fl 1") in df.columns)):
                df['Fl 1'] = [0]*len(df)

        if(not (("Fl 2") in df.columns)):
                df['Fl 2'] = [0]*len(df)

        df['ground'] = toPoints(ground_missions[phase][0], df['Ch␣
 ↪1'])+toPoints(ground_missions[phase][1], df['Ch␣
 ↪2'])+toPoints(ground_missions[phase][2], df['Ch␣
 ↪3'])+toPoints(ground_missions[phase][3], df['Ch␣
 ↪4'])+toPoints(ground_missions[phase][4], df['Ch 5'])

        df['ship'] = toPoints(ship_missions[phase][0], df['Fl 1'])+ \
           toPoints(ship_missions[phase][1], df['Fl 2'])

        df['ground_perc'] = round((df['ground']/max_ground[phase])*100, 0)

        df['ship_perc'] = round((df['ship']/max_ships[phase])*100, 0)

        ground_perc = df['ground_perc'].mean()
        ship_perc = df['ship_perc'].mean()

        return (ground_perc,ship_perc)
```

1.6.1 Percent of Combat Mission Points per Phase

```python
perc = list()
for i in range(len(data)):
        (ground_perc, ship_perc) = percents(data[i], i + 1)
        if math.isnan(ship_perc):
                ship_perc = 0
        if math.isnan(ground_perc):
                ground_perc = 0

        perc.append([str(round(ground_perc,2)) + '%', str(round(ship_perc,2)) +␣
  ↪'%'])

for i in range(len(data),4):
        perc.append(['NA','NA'])

perc_points = pd.DataFrame(perc, index=[
```

```
                          'Phase 1', 'Phase 2', 'Phase 3', 'Phase 4'],␣
 ↪columns=['Ground', 'Ships'])
print(perc_points)
```

```
        Ground   Ships
Phase 1  63.08%      0%
Phase 2   69.4%  31.42%
Phase 3  76.38%  36.82%
Phase 4  42.48%   76.0%
```

1.6.2 Percent of CM Completed by Player

```
[ ]: player_perc = pd.DataFrame()
for i in range(len(data)):
    ground_name = 'P' + str(i+1) + 'G'
    ship_name = 'P' + str(i+1) + 'S'
    player_perc[ground_name] = data[i]['ground_perc']
    player_perc[ship_name] = data[i]['ship_perc']
print(player_perc)
```

```
                         P1G  P1S    P2G    P2S    P3G    P3S   P4G    P4S
name
Doomslug the Destroyer   50.0  NaN   50.0   57.0   85.0    0.0  29.0  100.0
Larping Soccer Moms      91.0  NaN   94.0  100.0   92.0   57.0  41.0    0.0
Zhil Axfow                0.0  NaN   81.0    0.0   84.0    0.0  41.0    0.0
Elros Halfelven         100.0  NaN   94.0    0.0   77.0    0.0  34.0  100.0
Guntha Arbos             84.0  NaN  100.0    0.0  100.0   57.0  63.0    0.0
Plucky Haydon             0.0  NaN    0.0    0.0  100.0   57.0  58.0  100.0
Tommyboy85               91.0  NaN  100.0  100.0  100.0    0.0  52.0  100.0
Philo Beddoe            100.0  NaN   81.0   43.0   92.0  100.0  48.0  100.0
Loadage                 100.0  NaN  100.0  100.0  100.0  100.0  68.0  100.0
s o l o                 100.0  NaN  100.0  100.0  100.0  100.0  75.0  100.0
ilekkund                100.0  NaN  100.0    0.0  100.0    0.0  69.0  100.0
Higgs                    91.0  NaN   81.0   43.0   75.0  100.0  23.0  100.0
ilekkund2                91.0  NaN    0.0    0.0  100.0    0.0  46.0  100.0
MINI xipokemastrix        0.0  NaN   92.0    0.0   75.0    0.0  44.0  100.0
Flywire                  50.0  NaN   75.0    0.0  100.0    0.0  40.0  100.0
TacoPizza                91.0  NaN  100.0   57.0  100.0   57.0  58.0  100.0
ShootMeow                91.0  NaN  100.0  100.0  100.0  100.0  51.0  100.0
Heywood Jablowme         82.0  NaN   94.0   57.0   92.0    0.0  37.0  100.0
Dark Penguin             91.0  NaN  100.0    0.0    0.0    0.0  31.0    0.0
Obi Won Sebroni          82.0  NaN   50.0   57.0   77.0   57.0  30.0  100.0
joker                     0.0  NaN  100.0    0.0  100.0    0.0  55.0  100.0
Masajj Vemtits           50.0  NaN  100.0    0.0  100.0  100.0  50.0  100.0
Gryphix                   0.0  NaN  100.0   43.0    0.0    0.0  58.0  100.0
AKB                     100.0  NaN  100.0    0.0    0.0    0.0  48.0  100.0
Agave                    25.0  NaN  100.0  100.0  100.0    0.0  33.0  100.0
Theflavorgreen           57.0  NaN   64.0    0.0   48.0   57.0  21.0  100.0
```

```
Argarax               100.0  NaN  100.0    0.0   50.0    0.0  38.0  100.0
Baxston Kane           91.0  NaN  100.0   57.0  100.0    0.0  57.0    0.0
Neeb                   57.0  NaN   82.0   57.0   77.0  100.0  23.0    0.0
JustinAlexander11      91.0  NaN  100.0    0.0   92.0   57.0  40.0  100.0
Promethean             91.0  NaN   94.0    0.0  100.0    0.0  51.0    0.0
EvilCoyote2011         91.0  NaN  100.0    0.0  100.0  100.0  17.0  100.0
Calens                 66.0  NaN    0.0    0.0   92.0    0.0  44.0  100.0
Wolfman314            100.0  NaN   25.0    0.0  100.0  100.0  30.0  100.0
GNAY                   82.0  NaN    0.0  100.0  100.0   57.0  38.0  100.0
Revanche Gilder        82.0  NaN   94.0  100.0  100.0  100.0  44.0  100.0
MarkKenoburger         91.0  NaN  100.0    0.0   92.0    0.0  40.0  100.0
ONE                    66.0  NaN   70.0   43.0   75.0  100.0  44.0  100.0
OttoVonGens           100.0  NaN  100.0  100.0  100.0    0.0  65.0  100.0
MINI Stewabob          91.0  NaN   62.0    0.0   67.0   57.0  32.0  100.0
Maxaron Lexilon        34.0  NaN    0.0    0.0   50.0    0.0  31.0    0.0
Sultan2309             91.0  NaN    0.0    0.0   67.0   57.0  58.0    0.0
Exeel                   0.0  NaN    0.0    0.0    0.0    0.0  44.0    0.0
BabyYodaHitta          50.0  NaN    0.0    0.0   42.0   57.0  24.0  100.0
The Wall                0.0  NaN    0.0    0.0   84.0   57.0  30.0  100.0
SloppySaberFlavor       0.0  NaN    0.0    0.0    0.0    0.0  24.0    0.0
wamakima5004            0.0  NaN   87.0    0.0   84.0   57.0  29.0  100.0
Chaunce                 0.0  NaN  100.0  100.0  100.0    0.0  43.0  100.0
Zlada14                 0.0  NaN  100.0    0.0    0.0    0.0  50.0  100.0
```

1.6.3 Average TB Points per GP

```
[ ]: print(avgPointsPerGP)
```

```
name
GNAY                 1.547
ShootMeow            1.900
M1TTH                  NaN
Wolfman314           1.267
ilekkund2            1.363
Agave                1.543
Exeel                0.326
ilekkund             1.652
MINI xipokemastrix   1.450
Promethean           1.676
Obi Won Sebroni      1.997
Guntha Arbos         1.647
Higgs                1.816
Loadage              2.030
s o l o              1.749
Theflavorgreen       1.505
AKB                  1.295
Baxston Kane         1.411
Chaunce              1.478
```

```
Calens                   1.283
Flywire                  1.560
Argarax                  1.201
wamakima5004             1.219
Maxaron Lexilon          0.408
Elros Halfelven          1.609
Masajj Vemtits           1.748
BabyYodaHitta            1.778
Neeb                     1.540
SloppySaberFlavor        0.552
Heywood Jablowme         1.482
Philo Beddoe             1.895
TacoPizza                2.053
Zlada14                  0.915
Zhil Axfow               0.826
ONE                      1.716
Dark Penguin             1.180
Plucky Haydon            1.225
Revanche Gilder          1.819
Tommyboy85               1.756
Sultan2309               1.586
Larping Soccer Moms      2.371
OttoVonGens              1.638
Gryphix                  1.018
JustinAlexander11        1.714
Doomslug the Destroyer   1.431
The Wall                 1.074
joker                    1.438
EvilCoyote2011           1.696
MINI Stewabob            1.982
MarkKenoburger           1.600
dtype: float64
```

1.6.4 Guild TB Points and TB Points per GP

```python
data[-1].loc[:,['territoryPointsContributed','pointsPerGP']].
 ↪sort_values(by=['territoryPointsContributed'], ascending=False)
```

name	territoryPointsContributed	pointsPerGP
s o l o	16819750	2.094
ilekkund	15727362	2.106
OttoVonGens	15101128	2.093
Loadage	14426781	2.317
Gryphix	14199109	2.080
Plucky Haydon	13519169	2.199
Tommyboy85	13441711	2.036

ShootMeow	13360319	2.018
Zlada14	12853463	2.110
Masajj Vemtits	12778610	2.101
Heywood Jablowme	12776119	1.779
TacoPizza	12749155	2.371
Chaunce	12739577	1.924
ONE	12659903	1.964
AKB	12550114	2.065
Calens	12503492	1.988
Philo Beddoe	12469022	2.083
joker	12405558	2.339
Revanche Gilder	12293014	2.019
JustinAlexander11	11793020	1.970
Elros Halfelven	11660958	1.840
Guntha Arbos	11570765	1.816
Agave	11511672	1.848
Flywire	11430033	2.033
Wolfman314	11413093	1.795
ilekkund2	11403122	2.244
GNAY	11296827	2.014
wamakima5004	11142984	1.786
MINI xipokemastrix	10604074	2.415
MarkKenoburger	10376727	2.271
The Wall	10353644	1.953
Higgs	10089285	1.780
EvilCoyote2011	10081543	1.650
Promethean	9932351	1.735
Theflavorgreen	9754136	1.783
Sultan2309	9676056	2.006
Doomslug the Destroyer	9389521	2.100
Obi Won Sebroni	9180297	2.225
MINI Stewabob	8853820	2.406
Baxston Kane	8627834	1.266
Argarax	8221167	1.455
Exeel	7993018	1.215
Zhil Axfow	7560266	1.182
Dark Penguin	6998884	1.592
Larping Soccer Moms	6953759	1.972
Maxaron Lexilon	6373836	0.953
BabyYodaHitta	6023473	1.973
SloppySaberFlavor	5501819	0.912
Neeb	5411443	0.916
M1TTH	4595551	1.022