DSTB August 2021 Report

August 30, 2021

1 GSF Sigma LS Geo TB Report August 2021

1.1 Load Data

[35]:
```
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
```

[36]:
```
data[-1]['pointsPerGP'] = round(data[-1]['territoryPointsContributed'] /
                                (data[-1]['shipGP']+data[-1]['characterGP']), 3)
avgPointsPerGP = round(data[-1]['pointsPerGP'].mean(), 3)
```

1.2 Input

1.2.1 Sandbagging

[37]:
```
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: 34

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

1.3 Calculations

1.3.1 TB Points per CM

LS GEO TB

[38]:
```
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
        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

[39]:
```
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

[40]:
```
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

[41]:
```
n = 10
low_ppg = data[-1]['pointsPerGP'].sort_values().head(n)
print(low_ppg)
```

```
name
SloppySaberFlavor    1.995
Heywood Jablowme     2.134
Hirano               2.606
Agave                3.570
KingPete             4.230
LGuy 21              4.476
Flywire              4.486
Theflavorgreen       4.508
Elros Halfelven      4.797
Masajj Vemtits       4.851
Name: pointsPerGP, dtype: float64
```

1.4.2 Lowest CM Waves Completed

[42]:
```
low_cm = data[-1]['combatMissionWavesCompleted'].sort_values().head(n)
print(low_cm)
```

```
name
SloppySaberFlavor         18
Theflavorgreen            19
Hirano                    21
Heywood Jablowme          23
Agave                     30
Flywire                   33
M1TTH                     36
KingPete                  36
Doomslug the Destroyer    37
BabyYodaHitta             37
Name: combatMissionWavesCompleted, dtype: int64
```

1.4.3 Lowest TB Points

[43]:
```
low_tb = data[-1]['territoryPointsContributed'].sort_values().head(n)
print(low_tb)
```

```
name
SloppySaberFlavor    11388927
Heywood Jablowme     14618505
Hirano               19216346
Agave                20591334
KingPete             21270880
Theflavorgreen       22886582
BabyYodaHitta        23434391
Obi Won Sebroni      23532501
MINI Stewabob        24361420
Flywire              24399598
Name: territoryPointsContributed, dtype: int64
```

1.5 Top Performers

1.5.1 Highest TB Points per GP

[44]:
```
high_ppg = data[-1]['pointsPerGP'].sort_values(ascending = False).head(n)
print(high_ppg)
```

```
name
Larping Soccer Moms    9.516
PadawanTano            9.341
BabyYodaHitta          8.472
ilekkund2              8.422
MINI xipokemastrix     8.209
Loadage                8.202
GANIC                  7.556
MINI Stewabob          7.493
OttoVonGens            7.444
Zlada14                7.442
Name: pointsPerGP, dtype: float64
```

1.5.2 Highest Combat Waves Completed

[45]:
```
high_cm = data[-1]['combatMissionWavesCompleted'].sort_values(
    ascending=False).head(n)
print(high_cm)
```

```
name
OttoVonGens      69
Baxston Kane     69
ilekkund         68
Loadage          68
s o l o          67
```

```
Plucky Haydon     65
ShootMeow         64
ilekkund2         64
Guntha Arbos      63
TacoPizza         62
Name: combatMissionWavesCompleted, dtype: int64
```

1.5.3 Highest TB Points

[46]:
```
high_tb = data[-1]['territoryPointsContributed'].sort_values(
    ascending=False).head(n)
print(high_tb)
```

```
name
s o l o           52362053
OttoVonGens       51424402
ilekkund          51046947
Loadage           45067125
ShootMeow         44746265
Baxston Kane      44702893
Zlada14           44271050
Plucky Haydon     43666329
Gryphix           41109289
Philo Beddoe      40768598
Name: territoryPointsContributed, dtype: int64
```

1.6 Guild Performance

[47]:
```
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
```

[48]:
```
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'])

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

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

        return (ground_perc,ship_perc)
```

1.6.1 Percent of Combat Mission Points per Phase

[49]:
```
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(ground_perc) + '%', str(ship_perc) + '%'])

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  76.0%     0%
Phase 2  74.0%  17.0%
Phase 3  72.0%  39.0%
Phase 4  38.0%  72.0%
```

1.6.2 Average TB Points per GP

[50]:
```
print(avgPointsPerGP)
```

```
6.203
```

1.6.3 Guild TB Points and TB Points per GP

[51]:
```
data[-1].loc[:,['territoryPointsContributed','pointsPerGP']].
↪sort_values(by=['territoryPointsContributed'], ascending=False)
```

[51]:

name	territoryPointsContributed	pointsPerGP
s o l o	52362053	6.870
OttoVonGens	51424402	7.444
ilekkund	51046947	7.165
Loadage	45067125	8.202
ShootMeow	44746265	7.241
Baxston Kane	44702893	6.954
Zlada14	44271050	7.442
Plucky Haydon	43666329	7.333
Gryphix	41109289	6.238
Philo Beddoe	40768598	7.142
AKB	39958760	6.958
ilekkund2	38891091	8.422
Exeel	38406127	6.200
Higgs	38301723	7.131
Guntha Arbos	37887299	6.149
TacoPizza	37136044	7.310
Neeb	36847057	6.430
Chaunce	36243620	5.724
GANIC	35364152	7.556
Ben8cv	35336174	5.776
Maxaron Lexilon	35249322	5.534
Zhil Axfow	34857647	5.690
Wolfman314	34260983	5.547
ONE	33992617	5.522
Kypomm	33870963	5.348
Promethean	33393405	6.211
MINI xipokemastrix	33321226	8.209
wamakima5004	32884290	5.457
Elladan Halfelven	32831527	5.112
Argarax	32036684	7.004
Larping Soccer Moms	31948446	9.516
The Wall	30673173	6.247
MINICalens	30649867	5.164
Dark Penguin	30380580	7.334
Elros Halfelven	29429693	4.797
Masajj Vemtits	28344385	4.851
LGuy 21	27625048	4.476
Doomslug the Destroyer	26729589	6.935
M1TTH	25264141	5.616
PadawanTano	25180353	9.341

Flywire	24399598	4.486
MINI Stewabob	24361420	7.493
Obi Won Sebroni	23532501	7.073
BabyYodaHitta	23434391	8.472
Theflavorgreen	22886582	4.508
KingPete	21270880	4.230
Agave	20591334	3.570
Hirano	19216346	2.606
Heywood Jablowme	14618505	2.134
SloppySaberFlavor	11388927	1.995