LSTB August 2021 Report

August 16, 2021

1 GSF Sigma LS Geo TB Report August 2021

1.1 Load Data

[1]:
```
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
```

[2]:
```
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

[3]:
```
tbType = 'LS'

sbag_1_top = False
sbag_1_mid = True
sbag_1_bottom = False

sbag_2_top = False
sbag_2_mid = False
sbag_2_bottom = False

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

1.2.2 Shards and Stars

KAM shards: 25

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

1.3 Calculations

1.3.1 TB Points per CM

LS GEO TB

[4]:
```
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

[5]:
```
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

[6]:
```
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

[7]:
```
n = 10
low_ppg = data[-1]['pointsPerGP'].sort_values().head(n)
print(low_ppg)
```

```
name
Zhil Axfow            2.665
Kypomm                2.737
ONE                   2.962
Masajj Vemtits        2.978
Higgs                 2.983
Guntha Arbos          3.209
Obi Won Sebroni       3.215
BabyYodaHitta         3.216
Philo Beddoe          3.411
MINI xipokemastrix    3.440
Name: pointsPerGP, dtype: float64
```

1.4.2 Lowest CM Waves Completed

[8]:
```
low_cm = data[-1]['combatMissionWavesCompleted'].sort_values().head(n)
print(low_cm)
```

```
name
Obi Won Sebroni           1
Kypomm                    3
Guntha Arbos              3
Wolfman                   5
Higgs                     6
MINI xipokemastrix        6
ONE                       6
BabyYodaHitta             6
Doomslug the Destroyer    7
Dark Penguin              8
Name: combatMissionWavesCompleted, dtype: int64
```

1.4.3 Lowest TB Points

[9]:
```
low_tb = data[-1]['territoryPointsContributed'].sort_values().head(n)
print(low_tb)
```

```
name
BabyYodaHitta               8848657
Obi Won Sebroni            10070186
Larping Soccer Moms        12975504
MINI xipokemastrix         13749442
Doomslug the Destroyer     13872651
Dark Penguin               14965261
Higgs                      15874988
Zhil Axfow                 16186639
Argarax                    16513575
M1TTH                      16559743
Name: territoryPointsContributed, dtype: int64
```

1.5 Top Performers

1.5.1 Highest TB Points per GP

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

```
name
MINI UK PedroMontenegro    5.819
MINI Stewabob              5.620
Elros Halfelven            5.480
s o l o                    5.191
Loadage                    5.029
Zlada14                    4.983
MINICalens                 4.877
OttoVonGens                4.723
Promethean                 4.677
Exeel                      4.618
Name: pointsPerGP, dtype: float64
```

1.5.2 Highest Combat Waves Completed

[11]:
```
high_cm = data[-1]['combatMissionWavesCompleted'].sort_values(
    ascending=False).head(n)
print(high_cm)
```

```
name
MINI UK PedroMontenegro    48
s o l o                    43
ilekkund                   40
Elros Halfelven            39
OttoVonGens                38
```

```
Zlada14                    38
Exeel                      36
Gryphix                    35
Chaunce                    34
Promethean                 33
Name: combatMissionWavesCompleted, dtype: int64
```

1.5.3 Highest TB Points

[12]:
```
high_tb = data[-1]['territoryPointsContributed'].sort_values(
    ascending=False).head(n)
print(high_tb)
```

```
name
s o l o                    38785014
MINI UK PedroMontenegro    38348230
Elros Halfelven            33557762
OttoVonGens                32485233
ilekkund                   31920072
Zlada14                    29518498
Baxston Kane               29457868
MINICalens                 28829787
Chaunce                    28635149
Exeel                      28391679
Name: territoryPointsContributed, dtype: int64
```

1.6 Guild Performance

[13]:
```
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
```

[14]:
```
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

[15]:
```
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  39.0%  42.0%
Phase 2  43.0%  58.0%
Phase 3  42.0%  58.0%
Phase 4  28.0%  26.0%
```

1.6.2 Average TB Points per GP

[16]:
```
print(avgPointsPerGP)
```

```
4.068
```

1.6.3 Guild TB Points and TB Points per GP

[17]:
```
data[-1].loc[:,['territoryPointsContributed','pointsPerGP']].
↪sort_values(by=['territoryPointsContributed'], ascending=False)
```

[17]:

name	territoryPointsContributed	pointsPerGP
s o l o	38785014	5.191
MINI UK PedroMontenegro	38348230	5.819
Elros Halfelven	33557762	5.480
OttoVonGens	32485233	4.723
ilekkund	31920072	4.560
Zlada14	29518498	4.983
Baxston Kane	29457868	4.598
MINICalens	28829787	4.877
Chaunce	28635149	4.546
Exeel	28391679	4.618
Heywood Jablowme	28193416	4.154
Loadage	27428043	5.029
Wolfman314	27304394	4.433
ShootMeow	26768790	4.372
Gryphix	26613843	4.042
Maxaron Lexilon	26416348	4.177
wamakima5004	26222319	4.366
AKB	25934654	4.537
LGuy 21	25895386	4.218
Hirano	25812787	3.506
Promethean	25108160	4.677
Elladan Halfelven	24894039	3.886
Ben8cv	24237931	3.987
Flywire	24084128	4.463
SloppySaberFlavor	22835397	4.023
Agave	22687380	3.955
Neeb	20859891	3.690
ilekkund2	20791415	4.599
GANIC	20548796	4.426
Philo Beddoe	19308341	3.411
Guntha Arbos	19265480	3.209
Theflavorgreen	19037266	3.811
KingPete	18764934	3.772
ONE	18120737	2.962
MINI Stewabob	18014616	5.620
Wolfman	17860027	3.918
TacoPizza	17415480	3.461
Masajj Vemtits	17306988	2.978
Kypomm	17242440	2.737
The Wall	16672360	3.455

M1TTH	16559743	3.683
Argarax	16513575	3.670
Zhil Axfow	16186639	2.665
Higgs	15874988	2.983
Dark Penguin	14965261	3.652
Doomslug the Destroyer	13872651	3.691
MINI xipokemastrix	13749442	3.440
Larping Soccer Moms	12975504	3.908
Obi Won Sebroni	10070186	3.215
BabyYodaHitta	8848657	3.216