

LSTB September 2021 Report

September 14, 2021

1 GSF Sigma LS Geo TB Report September 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 = True

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

1.2.2 Shards and Stars

KAM shards: 30

Phase 4	Phase 3	Phase 2	Phase 1
x	3	2	2
x	x	3	3
x	0	2	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
EvilCoyote2011      3.034
Zhil Axflow        3.107
ONE                 3.201
JJ Bolte           3.568
BabyYodaHitta      3.855
SloppySaberFlavor  4.088
Theflavorgreen     4.216
Ben8cv             4.263
Zlada14            4.295
Masajj Vemtits     4.296
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
ONE                2
Obi Won Sebroni    5
Neeb               7
BabyYodaHitta      7
joker              8
Theflavorgreen     8
Zhil Axflow        10
M1TTH              11
Kypomm             11
Doomslug the Destroyer 12
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      10977172
Obi Won Sebroni    15888507
EvilCoyote2011     17872116
PadawanTano        18864680
Zhil Axflow        19285418
ONE                19786595
JJ Bolte           19924758
Doomslug the Destroyer 19960409
M1TTH              21298798
Theflavorgreen     21572901
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
Elros Halfelven    7.637
MINI Stewabob      7.632
ilekkund2          7.597
s o l o            7.294
Loadage            7.255
ilekkund           7.068
Larping Soccer Moms 7.037
Argarax            6.994
PadawanTano        6.826
Gryphix            6.805
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
s o l o      67
Elros Halfelven 59
ilekkund      58
OttoVonGens    53
Chaunce       52
```

Gryphix	48
Heywood Jablowme	46
Loadage	46
Agave	46
ilekkund2	42

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	56044950
ilekkund	50422906
Elros Halfelven	47277540
Heywood Jablowme	45391216
Gryphix	45010214
OttoVonGens	42873878
Chaunce	41700646
Loadage	40288463
AKB	39112885
LGuy 21	38788898

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'])

    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()

    # 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	33.3%	32.0%
Phase 2	42.9%	52.0%

Phase 3 40.26% 4.0%
Phase 4 31.88% 44.0%

1.6.2 Percent of CM Completed by Player

```
[19]: 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	0.0	0.0	8.0	0.0	24.0	100.0	11.0	0.0
Larping Soccer Moms	34.0	100.0	25.0	100.0	52.0	0.0	31.0	0.0
Zhil Axfow	20.0	0.0	24.0	0.0	17.0	0.0	0.0	0.0
Elros Halfelven	62.0	100.0	87.0	100.0	94.0	0.0	79.0	100.0
Plucky Haydon	70.0	100.0	0.0	0.0	40.0	0.0	74.0	100.0
Philo Beddoe	0.0	0.0	19.0	100.0	24.0	0.0	29.0	0.0
Loadage	68.0	100.0	66.0	100.0	62.0	0.0	50.0	100.0
BabyYodaHitta	12.0	0.0	13.0	0.0	17.0	0.0	0.0	0.0
s o l o	100.0	100.0	87.0	100.0	94.0	0.0	100.0	100.0
MINI xipokemastrix	31.0	0.0	20.0	0.0	23.0	0.0	18.0	100.0
TacoPizza	45.0	100.0	51.0	100.0	40.0	0.0	27.0	0.0
GANIC	62.0	0.0	72.0	100.0	56.0	0.0	44.0	0.0
Ben8cv	50.0	100.0	0.0	0.0	67.0	0.0	42.0	0.0
Heywood Jablowme	94.0	100.0	79.0	100.0	40.0	0.0	35.0	100.0
Obi Won Sebroni	8.0	0.0	8.0	0.0	17.0	0.0	0.0	0.0
Masajj Vemtits	47.0	0.0	32.0	100.0	0.0	0.0	11.0	0.0
AKB	59.0	100.0	68.0	0.0	51.0	0.0	55.0	100.0
Zlada14	0.0	0.0	60.0	0.0	0.0	0.0	52.0	100.0
Agave	77.0	100.0	58.0	100.0	78.0	100.0	50.0	0.0
ONE	0.0	0.0	0.0	0.0	7.0	0.0	0.0	0.0
LGuy 21	70.0	100.0	75.0	100.0	30.0	0.0	50.0	100.0
Theflavorgreen	0.0	0.0	13.0	0.0	19.0	0.0	7.0	0.0
OttoVonGens	43.0	100.0	87.0	100.0	100.0	0.0	54.0	100.0
Argarax	55.0	0.0	80.0	100.0	55.0	0.0	23.0	100.0
MINI Stewabob	0.0	0.0	0.0	0.0	40.0	0.0	42.0	0.0
Baxston Kane	77.0	0.0	49.0	0.0	56.0	0.0	57.0	0.0
Neeb	0.0	0.0	0.0	0.0	14.0	0.0	11.0	0.0
PadawanTano	0.0	0.0	33.0	100.0	24.0	0.0	16.0	100.0
Promethean	0.0	0.0	64.0	0.0	40.0	0.0	64.0	100.0
EvilCoyote2011	0.0	0.0	13.0	100.0	17.0	0.0	0.0	0.0
Exeel	70.0	100.0	44.0	100.0	23.0	0.0	23.0	100.0
Guntha Arbos	0.0	0.0	36.0	0.0	14.0	0.0	19.0	0.0

Flywire	0.0	0.0	66.0	100.0	40.0	0.0	23.0	100.0
Kypomm	0.0	0.0	13.0	100.0	23.0	0.0	0.0	0.0
Dark Penguin	27.0	0.0	25.0	100.0	42.0	0.0	0.0	0.0
Gryphix	40.0	0.0	79.0	100.0	62.0	0.0	92.0	100.0
JJ Bolte	0.0	0.0	49.0	0.0	0.0	0.0	38.0	0.0
Higgs	23.0	0.0	29.0	0.0	17.0	0.0	13.0	0.0
MINICalens	79.0	0.0	72.0	0.0	61.0	0.0	24.0	0.0
M1TTH	0.0	0.0	28.0	100.0	0.0	0.0	0.0	100.0
Wolfman314	0.0	0.0	16.0	100.0	40.0	0.0	23.0	100.0
The Wall	0.0	0.0	58.0	0.0	67.0	0.0	45.0	0.0
ilekkund	87.0	0.0	87.0	100.0	89.0	0.0	73.0	100.0
ilekkund2	55.0	100.0	54.0	100.0	57.0	0.0	57.0	100.0
joker	0.0	0.0	7.0	0.0	12.0	0.0	13.0	100.0
wamakima5004	79.0	100.0	79.0	100.0	57.0	0.0	7.0	100.0
SloppySaberFlavor	0.0	0.0	60.0	0.0	46.0	0.0	0.0	0.0
Chaunce	76.0	100.0	62.0	100.0	94.0	0.0	64.0	0.0
ShootMeow	45.0	0.0	33.0	0.0	29.0	0.0	23.0	0.0
Maxaron Lexilon	0.0	0.0	57.0	0.0	42.0	0.0	25.0	0.0

1.6.3 Average TB Points per GP

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

5.541

1.6.4 Guild TB Points and TB Points per GP

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

```
[18]:
```

	territoryPointsContributed	pointsPerGP
name		
s o l o	56044950	7.294
ilekkund	50422906	7.068
Elros Halfelven	47277540	7.637
Heywood Jablowme	45391216	6.617
Gryphix	45010214	6.805
OttoVonGens	42873878	6.190
Chaunce	41700646	6.523
Loadage	40288463	7.255
AKB	39112885	6.746
LGuy 21	38788898	6.250
Baxston Kane	38702842	5.949
Agave	38336782	6.628
Plucky Haydon	38091441	6.386
MINICalens	37587235	6.282
Promethean	36045775	6.704
ilekkund2	35540078	7.597

Wolfman314	35107264	5.665
Flywire	34116998	6.250
Argarax	33410935	6.994
Maxaron Lexilon	33396719	5.219
GANIC	31833006	6.738
Philo Beddoe	31394788	5.467
wamakima5004	30843878	5.100
Kypomm	29728263	4.667
Exeel	29381548	4.713
ShootMeow	29187927	4.663
TacoPizza	28555314	5.595
Guntha Arbos	27584506	4.453
Ben8cv	26169178	4.263
Higgs	26064589	4.828
Zlada14	25859647	4.295
Neeb	25846457	4.504
The Wall	25772133	5.190
MINI Stewabob	25297263	7.632
Masajj Vemtits	25163429	4.296
Larping Soccer Moms	23761181	7.037
SloppySaberFlavor	23508273	4.088
joker	22860932	4.603
Dark Penguin	22722656	5.393
MINI xipokemastrix	22048508	5.358
Theflavorgreen	21572901	4.216
MITTH	21298798	4.734
Doomslug the Destroyer	19960409	4.947
JJ Bolte	19924758	3.568
ONE	19786595	3.201
Zhil Axflow	19285418	3.107
PadawanTano	18864680	6.826
EvilCoyote2011	17872116	3.034
Obi Won Sebroni	15888507	4.597
BabyYodaHitta	10977172	3.855