DSTB_July_2021_report

July 31, 2021

1 GSF Sigma LS Geo TB Report July 2021

1.1 Load Data

[18]:
```
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
```

[19]:
```
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

[20]:
```
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: 40

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

1.3 Calculations

1.3.1 TB Points per CM

LS GEO TB

[21]:
```
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

[22]:
```
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

[23]:
```
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

[24]:
```
n = 10
low_ppg = data[-1]['pointsPerGP'].sort_values().head(n)
print(low_ppg)
```

```
name
SloppySaberFlavor    2.735
Hirano               3.112
LGuy 21              4.322
Zlada14              4.360
MINI Stewabob        4.424
MINI BigFish         4.753
Maxaron Lexilon      4.839
ShootMeow            4.915
Philo Beddoe         4.943
Promethean           5.241
Name: pointsPerGP, dtype: float64
```

1.4.2 Lowest CM Waves Completed

[25]:
```
low_cm = data[-1]['combatMissionWavesCompleted'].sort_values().head(n)
print(low_cm)
```

```
name
SloppySaberFlavor    15
MINI Stewabob        18
Theflavorgreen       26
Obi Won Sebroni      33
Hirano               35
M1TTH                35
Zhil Axfow           36
BabyYodaHitta        36
MINI BigFish         37
LGuy 21              40
Name: combatMissionWavesCompleted, dtype: int64
```

1.4.3 Lowest TB Points

[26]:
```
low_tb = data[-1]['territoryPointsContributed'].sort_values().head(n)
print(low_tb)
```

```
name
MINI Stewabob          14044588
SloppySaberFlavor      15407590
Obi Won Sebroni        18079465
BabyYodaHitta          22257431
Hirano                 22630277
M1TTH                  23868420
Zlada14                25667850
MINI xipokemastrix     25762803
LGuy 21                25826513
The Wall               26534149
Name: territoryPointsContributed, dtype: int64
```

1.5 Top Performers

1.5.1 Highest TB Points per GP

[27]:
```
high_ppg = data[-1]['pointsPerGP'].sort_values(ascending = False).head(n)
print(high_ppg)
```

```
name
Larping Soccer Moms       10.318
Dark Penguin               9.085
Loadage                    8.813
Argarax                    8.503
BabyYodaHitta              8.185
TacoPizza                  8.130
ilekkund2                  8.114
Doomslug the Destroyer     7.893
GANIC                      7.809
Flywire                    7.332
Name: pointsPerGP, dtype: float64
```

1.5.2 Highest Combat Waves Completed

[28]:
```
high_cm = data[-1]['combatMissionWavesCompleted'].sort_values(
    ascending=False).head(n)
print(high_cm)
```

```
name
Loadage              70
Masajj Vemtits       68
s o l o              68
OttoVonGens          67
```

```
MINI xIPokemastrIx    67
Tomer Nola            66
MagnusV               65
Gryphix               65
Argarax               64
AKB                   63
Name: combatMissionWavesCompleted, dtype: int64
```

1.5.3 Highest TB Points

[29]:
```
high_tb = data[-1]['territoryPointsContributed'].sort_values(
    ascending=False).head(n)
print(high_tb)
```

```
name
s o l o               49484004
Loadage               47634423
OttoVonGens           45400419
Gryphix               45237351
MINI xIPokemastrIx    43169002
Heywood Jablowme      42941562
MagnusV               42503395
TacoPizza             40696827
Flywire               39421721
Masajj Vemtits        39339520
Name: territoryPointsContributed, dtype: int64
```

1.6 Guild Performance

[30]:
```
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
```

[31]:
```
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

[32]:
```
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  70.0%     0%
Phase 2  74.0%  33.0%
Phase 3  76.0%  35.0%
Phase 4  38.0%  74.0%
```

1.6.2 Average TB Points per GP

[33]:
```
print(avgPointsPerGP)
```

```
6.14
```

1.6.3 Guild TB Points and TB Points per GP

[34]:
```
data[-1].loc[:,['territoryPointsContributed','pointsPerGP']].
↪sort_values(by=['territoryPointsContributed'], ascending=False)
```

[34]:

name	territoryPointsContributed	pointsPerGP
s o l o	49484004	6.721
Loadage	47634423	8.813
OttoVonGens	45400419	6.663
Gryphix	45237351	6.893
MINI xIPokemastrIx	43169002	5.489
Heywood Jablowme	42941562	6.366
MagnusV	42503395	6.832
TacoPizza	40696827	8.130
Flywire	39421721	7.332
Masajj Vemtits	39339520	6.815
Wolfman314	39319029	6.444
Tomer Nola	39247804	6.560
Elladan Halfelven	37501129	5.874
ilekkund	37254380	5.323
Argarax	37083681	8.503
Dark Penguin	37040710	9.085
Guntha Arbos	37027995	6.219
ilekkund2	36450205	8.114
Kypomm	36135149	5.853
Higgs	36126392	6.886
AKB	35942285	6.366
GANIC	35892458	7.809
ONE	35065572	5.738
Chaunce	34985938	5.635
Exeel	34621197	5.689
Larping Soccer Moms	34036676	10.318
Baxston Kane	33369687	5.265
Elros Halfelven	32879240	5.380
Zhil Axfow	32720100	5.413
Neeb	32585472	5.794
MINI BigFish	32524691	4.753
wamakima5004	32122233	5.392
MINICalens	32101606	5.470
Maxaron Lexilon	30357909	4.839
Agave	30184399	5.291

ShootMeow	29477634	4.915
Doomslug the Destroyer	28344232	7.893
Promethean	27810922	5.241
Philo Beddoe	27384074	4.943
Theflavorgreen	27112042	5.473
The Wall	26534149	5.599
LGuy 21	25826513	4.322
MINI xipokemastrix	25762803	6.529
Zlada14	25667850	4.360
M1TTH	23868420	5.308
Hirano	22630277	3.112
BabyYodaHitta	22257431	8.185
Obi Won Sebroni	18079465	5.887
SloppySaberFlavor	15407590	2.735
MINI Stewabob	14044588	4.424