DSTB October 2021 Report

October 29, 2021

1 GSF Sigma DS Geo TB Report September 2021

1.1 Load Data

[ ]:

```
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
```

[ ]:

```
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

[ ]:

```
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

[ ]:

```
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

[ ]:

```
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
wamakima5004         2.442
Elros Halfelven      2.540
M1TTH                2.755
SloppySaberFlavor    3.211
Veristas             3.254
The Wall             3.525
Maxaron Lexilon      3.652
Wolfman314           4.126
Plucky Haydon        4.383
Chaunce              4.433
Name: pointsPerGP, dtype: float64
```

1.4.2 Lowest CM Waves Completed

[ ]:

```
if len(data) == 4:
    cm = data[3]['combatMissionWavesCompleted'] -␣
 ↪data[2]['combatMissionWavesCompleted']
else:
    cm = data[-1]['combatMissionWavesCompleted']
low_cm = cm.sort_values().head(n)
print(low_cm)
```

```
name
Larping Soccer Moms    0
M1TTH                  3
wamakima5004          4
Theflavorgreen         4
Masajj Vemtits         5
The Wall               5
joker                  6
Maxaron Lexilon        6
Veristas               6
PadawanTano            6
```

```
Name: combatMissionWavesCompleted, dtype: int64
```

1.4.3 Lowest TB Points

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

```
name
M1TTH                      12396386
wamakima5004               14881815
Veristas                   15384646
Elros Halfelven            15870337
The Wall                   18195861
Larping Soccer Moms        18984725
SloppySaberFlavor          19017906
BabyYodaHitta              22126078
Doomslug the Destroyer     23455347
PadawanTano                23871903
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
Promethean       8.422
ilekkund2        8.353
PadawanTano      8.113
TacoPizza        8.042
Loadage          7.690
Baxston Kane     7.674
Tommyboy85       7.634
MINI Stewabob    7.485
BabyYodaHitta    7.450
GANIC            7.395
Name: pointsPerGP, dtype: float64
```

1.5.2 Highest Combat Waves Completed

```
[ ]: high_cm = cm.sort_values(
         ascending=False).head(n)
     print(high_cm)
```

```
name
s o l o         17
Baxston Kane    15
```

```
ilekkund         15
Zlada14          14
Gryphix          14
Guntha Arbos     14
Loadage          14
Tommyboy85       13
ShootMeow        13
AKB              13
Name: combatMissionWavesCompleted, dtype: int64
```

1.5.3 Highest TB Points

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

```
name
s o l o          57223818
ilekkund         54236038
Baxston Kane     50933704
Tommyboy85       49270404
OttoVonGens      47294508
ShootMeow        45768419
Guntha Arbos     45413678
Promethean       45302087
Loadage          45190497
ONE              43343297
Name: territoryPointsContributed, dtype: int64
```

1.6 Guild Performance

[ ]:
```
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
```

[ ]:
```
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

```
[ ]: 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   73.0%      0%
Phase 2   71.7%   27.7%
Phase 3  73.74%  37.42%
Phase 4  40.08%   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
OttoVonGens              91.0  NaN  100.0   57.0  100.0  100.0  46.0  100.0
TacoPizza               100.0  NaN  100.0   57.0   92.0   43.0  52.0  100.0
Zlada14                  91.0  NaN  100.0   43.0  100.0    0.0  58.0  100.0
GANIC                    72.0  NaN   85.0    0.0  100.0   57.0  43.0  100.0
Heywood Jablowme         91.0  NaN   94.0    0.0   84.0    0.0  48.0  100.0
Obi Won Sebroni          75.0  NaN   57.0    0.0   62.0   57.0  26.0  100.0
Calens                   91.0  NaN  100.0    0.0  100.0    0.0  29.0    0.0
Promethean              100.0  NaN  100.0   43.0  100.0  100.0  49.0  100.0
Revanche Gilder          91.0  NaN   81.0    0.0   75.0   57.0  40.0  100.0
Guntha Arbos             84.0  NaN  100.0    0.0  100.0   43.0  58.0  100.0
Loadage                  91.0  NaN  100.0   43.0   92.0  100.0  72.0  100.0
Argarax                  91.0  NaN  100.0   57.0  100.0    0.0  50.0  100.0
Baxston Kane            100.0  NaN  100.0   57.0  100.0  100.0  65.0  100.0
SloppySaberFlavor         0.0  NaN   87.0    0.0    0.0    0.0  24.0  100.0
ONE                      57.0  NaN   81.0   57.0   77.0  100.0  30.0  100.0
Tommyboy85              100.0  NaN  100.0   57.0  100.0  100.0  54.0  100.0
s o l o                 100.0  NaN  100.0  100.0  100.0  100.0  75.0  100.0
Maxaron Lexilon          66.0  NaN   94.0    0.0   50.0    0.0  25.0    0.0
Chaunce                 100.0  NaN   62.0    0.0   92.0    0.0  38.0  100.0
Neeb                     50.0  NaN   81.0   57.0   62.0   43.0  34.0    0.0
Zhil Axfow               50.0  NaN   62.0   43.0  100.0   57.0  38.0  100.0
Plucky Haydon             0.0  NaN   81.0    0.0   50.0   57.0  41.0  100.0
Philo Beddoe             91.0  NaN  100.0    0.0   75.0   57.0  52.0  100.0
Higgs                    72.0  NaN   74.0   57.0   84.0   43.0  26.0  100.0
The Wall                 72.0  NaN    0.0    0.0    0.0    0.0  21.0    0.0
Wolfman314              100.0  NaN   25.0   43.0   50.0    0.0  30.0  100.0
AKB                     100.0  NaN   94.0    0.0  100.0   57.0  52.0  100.0
JustinAlexander11       100.0  NaN  100.0    0.0   84.0    0.0  38.0  100.0
EvilCoyote2011           91.0  NaN  100.0    0.0   75.0    0.0  51.0  100.0
Theflavorgreen           50.0  NaN   59.0    0.0   34.0    0.0  18.0    0.0
Veristas                 91.0  NaN    0.0    0.0   75.0    0.0  25.0    0.0
PadawanTano              66.0  NaN   72.0    0.0   69.0    0.0  24.0  100.0
Doomslug the Destroyer   25.0  NaN   50.0    0.0   42.0   43.0  26.0  100.0
Larping Soccer Moms      91.0  NaN   88.0    0.0   77.0   57.0   0.0    0.0
Flywire                  75.0  NaN    0.0    0.0   75.0    0.0  55.0  100.0
MINI xipokemastrix       91.0  NaN   85.0    0.0  100.0    0.0  49.0    0.0
```

```
ShootMeow               100.0  NaN  100.0    0.0  100.0  100.0  52.0  100.0
wamakima5004            100.0  NaN   87.0    0.0    0.0    0.0  19.0    0.0
Gryphix                 100.0  NaN    0.0  100.0   34.0    0.0  58.0  100.0
Agave                    82.0  NaN  100.0  100.0  100.0    0.0  52.0    0.0
MINI Stewabob            25.0  NaN    0.0    0.0   67.0   57.0  27.0  100.0
Elros Halfelven          50.0  NaN    0.0    0.0    0.0    0.0  52.0  100.0
ilekkund                100.0  NaN  100.0  100.0  100.0  100.0  63.0  100.0
ilekkund2               100.0  NaN  100.0  100.0   59.0  100.0  43.0  100.0
Exeel                     0.0  NaN   62.0    0.0  100.0    0.0  38.0  100.0
BabyYodaHitta            41.0  NaN    0.0    0.0   50.0    0.0  41.0  100.0
joker                     0.0  NaN   62.0   57.0  100.0    0.0  27.0    0.0
Masajj Vemtits            0.0  NaN  100.0   57.0   92.0  100.0  19.0  100.0
```

1.6.3 Average TB Points per GP

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

```
5.975
```

1.6.4 Guild TB Points and TB Points per GP

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

```
[ ]:                         territoryPointsContributed  pointsPerGP
      name
      s o l o                                   57223818        7.254
      ilekkund                                  54236038        7.341
      Baxston Kane                              50933704        7.674
      Tommyboy85                                49270404        7.634
      OttoVonGens                               47294508        6.726
      ShootMeow                                 45768419        7.073
      Guntha Arbos                              45413678        7.230
      Promethean                                45302087        8.422
      Loadage                                   45190497        7.690
      ONE                                       43343297        6.842
      TacoPizza                                 42425288        8.042
      Heywood Jablowme                          42198547        6.005
      AKB                                       41952972        7.034
      ilekkund2                                 41672017        8.353
      Philo Beddoe                              39984821        6.832
      Argarax                                   39347708        7.239
      Zlada14                                   38532893        6.348
      JustinAlexander11                         37601707        6.580
      GANIC                                     36638349        7.395
      Agave                                     36337285        6.098
      EvilCoyote2011                            36233840        6.051
      Higgs                                     35637798        6.410
```

Zhil Axfow	35551195	5.642
Revanche Gilder	35472788	6.183
Neeb	33133954	5.686
Gryphix	31925030	4.732
Masajj Vemtits	31781656	5.290
Calens	31696855	5.172
Exeel	31339817	4.935
MINI xipokemastrix	30868312	7.186
Flywire	30103223	5.438
Chaunce	28928672	4.433
Obi Won Sebroni	28076537	7.291
Dark Penguin	27237460	6.385
Theflavorgreen	27222557	5.200
MINI Stewabob	26605322	7.485
Plucky Haydon	26524240	4.383
Wolfman314	25979349	4.126
joker	25957566	5.005
Maxaron Lexilon	24054433	3.652
PadawanTano	23871903	8.113
Doomslug the Destroyer	23455347	5.459
BabyYodaHitta	22126078	7.450
SloppySaberFlavor	19017906	3.211
Larping Soccer Moms	18984725	5.518
The Wall	18195861	3.525
Elros Halfelven	15870337	2.540
Veristas	15384646	3.254
wamakima5004	14881815	2.442
M1TTH	12396386	2.755