

How to use `wagc_points.py`

Reinhardt Messerschmidt

2006-10-22

1 Introduction

1.1 History

The calculation of SAGA ranks and WAGC points used to be done with a Visual Basic program maintained by Paul Edwards. In September 2005, SAGA officially recognised the ranks calculated by the script run on the SAGC website after its algorithm was adjusted to be the same as the Visual Basic program's algorithm. However, the script did not calculate WAGC points, and so the Python script `wagc_points.py` was written to fill this gap.

1.2 Objective

See the *WAGC points system* page on the SAGC wiki for what `wagc_points.py` tries to do.

2 Input

`wagc_points.py` requires two files as input: a parameter file and a data file.

2.1 Parameter file

The parameter file should have the following format:

```
<Annual depreciation factor>
*
<Representative's depreciation cap>
*
<Membership points>
*
<SA Championship points for 1st place>
<SA Championship points for 2nd place>
<...>
<SA Championship points for the rest>
*
```

<Number of Internet Championship tournaments taken into account>
 <Internet Championship points for 1st place>
 <Internet Championship points for 2nd place>
 <...>
 <Internet Championship points for the rest>
 *
 <Participation points cap>
 <Participation points cap per opponent>
 <Participation points for a tournament game>
 <Participation points for a club game>
 <Participation points for an internet game>

Note that the lengths of the lists of SA Championship and Internet Championship points are not fixed — the lists can be made longer or shorter as needed.

2.2 Data file

The data file should have the following format:

<Calender year for which WAGC points are being calculated>
 <Opening balances file>
 <Representatives file>
 <Membership file>
 <SA Championship file>
 <Internet Championship file>
 <Games file>

2.2.1 Opening balances file

wagc_points.py's output from the previous calendar year.

2.2.2 Representatives file

The representatives file should have a line for each representative in the following format:

<Name>, <Sponsorship level>, <Number of times>, <Position>, <Number of participants>

Sponsorship level is a number between 0 and 100, based on the level of sponsorship of the tournament. Also, the code assumes no player appears more than once in the representatives file.

2.2.3 Membership file

The members for the current calendar year, one name per line.

2.2.4 SA Championship file

The SA Championship file should have the following format:

```
<Name of 1st placed qualifier>[, <Name of 1st placed qualifier>[...]]  
<Name of 2nd placed qualifier>[, <Name of 2nd placed qualifier>[...]]  
<...>  
*  
<Names of participants in SA Open and qualifying tournaments, one name per  
line>
```

Square brackets indicate optional arguments that are only used if there is a tie for a position. It is not necessary to exclude the names of qualifiers from the names below the *, or to have only one instance of a name below the * — `wagc_points.py` handles any repetition.

Note that this list should only contain SAGA members. Otherwise, points are allocated (but not awarded) to players who are not SAGA members, disadvantaging SAGA members placed below them.

2.2.5 Internet Championship file

The Internet Championship file should have a group of lines for each internet tournament, in the following format:

```
<Name of 1st placed player>[, <Name of 1st placed player>[...]]  
<Name of 2nd placed player>[, <Name of 2nd placed player>[...]]  
<...>  
*
```

Square brackets indicate optional arguments that are only used if there is a tie for a position.

Note that this list should only contain SAGA members. Otherwise, points are allocated (but not awarded) to players who are not SAGA members, disadvantaging SAGA members placed below them.

2.2.6 Games file

The SAGC ranking system script generates a file with information on all the logged games. Executing the command

```
python make_input.py <Calender year for which WAGC points are being cal-  
culated> <SAGC games file>
```

creates a file called `games.txt`, which should be used as the games file. `make_input.py` does the following to the SAGC games file:

- Removes all free games and games not in the calendar year for which WAGC points are being calculated.
- Changes the *date* format from *dd/mm/yyyy* to *yyyymmdd*.
- Changes the *game type* format from *0.5|1|1.5* to *i|c|t*.
- Changes the *result* format from *W|B* to *w|b*.
- Removes the *effective handicap* and *comments* columns.
- Changes the order of the remaining columns.
- Sorts the games by date.

3 Execution

The following command should be used:

```
python wagg_points.py <parameter file> <data file>
```

4 Output

`wagg_points.py` creates a file called `wagg_points.txt` giving a summary of the WAGC points standings. It also creates a directory called `recordsheets` containing a detailed report for each player.