Co-funded by the European Union

Team Building Game

Script Instructions

by Jarosław Jablonka, PhD, University of Bielsko-Biala, POLAND

The document contains instructions on how to run a file (python script) that performs the necessary calculations to create teams based on the findings made by the game participants.

It describes the files needed, the purpose of each file, the reports generated by the script, and what can be changed in the code.

1. Required Files:

a) **File `participants.csv`** : This is the primary input file that contains a list of participants along with their roles and competencies. It must be structured correctly to ensure the game functions properly.

Structure of `participants.csv` :

- Column `Name` : The name of the participant.
- Column `Manager` : Indicates whether the participant is a manager (`YES`) or a candidate (`NO`).
- Columns `Comp1` to `Comp5` : The 5 competencies chosen by each participant in order from most important (weight 5) to least important (weight 1).

b) **Python Script**: This script contains all the functions that handle the game logic, such as assigning candidates to projects, calculating scores, and generating reports.

2. How the Files Work and Instructions

a) **`participants.csv`** : This file contains the input data necessary for the game to function. It includes all participants, their roles, and the competencies they have chosen. The order of competencies for each participant should be from the most important (weight 5) to the least important (weight 1).

Ensure that this file is correctly structured before running the script.

b) **Python Script**: The script performs all the stages of the game, from assigning candidates to projects to generating the reports. It will:

- Load the data from the `participants.csv` file.
- Randomly assign projects to managers.
- Calculate the suitability of candidates based on the competencies selected.

- Assign candidates to projects based on their scores.
- Generate reports detailing project assignments, manager choices, and candidate outcomes.

3. Reports Generated by the Script

The script generates a report in `.txt` or `.pdf` format that includes:

- A list of assigned projects and managers.
- Manager preferences (competencies without weights, in order of importance).
- Candidate scores and suitability for each project.
- Information on candidates who were not selected due to tie-breaking.
- The project order randomly generated in each round.

4. Elements in the Code You Can Modify

a) **CSV Modifications**: You can update the `participants.csv` file depending on the participants involved. It is crucial to maintain the correct structure of this file.

b) **Competency Weights**: In the current version, competency weights are set from 5 (most important) to 1 (least important). You can adjust this system either in the CSV file or by modifying the scoring functions in the script.

c) **Report Generation**: The script currently generates reports in `.txt` or `.pdf` formats. You can modify the script to generate reports in different formats if needed.

5. Running the Script in Different Environments

This Python script can be executed in various Python environments, including Anaconda, Spyder, Google Colab, and Visual Studio Code. Below are the instructions for running the script in each environment, as well as the necessary additional modules.

a) **Anaconda**:

1. Open Anaconda Navigator and create a new environment if necessary.
2. Install the required modules by running the following commands in the terminal:

   ```
   pip install pandas
   pip install reportlab
   ```
3. Run the script in Spyder or JupyterLab, both of which are available through Anaconda Navigator.

b) **Spyder**:

1. Open Spyder (you can install it via Anaconda if necessary).
2. Make sure that the required modules (`pandas`, `reportlab`) are installed using the following command in the terminal:

```
pip install pandas reportlab
```

3. Run the script in Spyder’s editor by pressing the green 'Run' button.

c) **Google Colab**:

1. Open Google Colab in your browser.
2. Upload the script and the `participants.csv` file to Colab.
3. Install the necessary modules by running:

```
!pip install pandas reportlab
```

4. Run the script in Colab by pressing the 'Run' button next to each code cell.

d) **Visual Studio Code (VS Code)**:

1. Open Visual Studio Code and install the Python extension if necessary.
2. Install the required modules by running the following commands in VS Code’s terminal:

```
pip install pandas reportlab
```

3. Run the script by opening the `.py` file in VS Code and pressing 'F5' to start debugging.