Contents

1. Graphical interface vs command line interface

If the program is run without any parametrs, then it automaticly start the graphical interface.

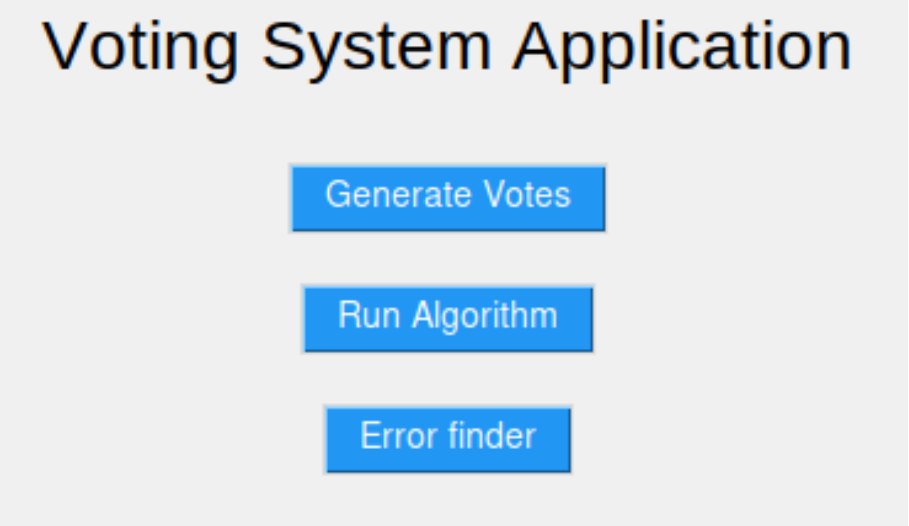
```
/bin/python ./vote_streams/src/main.py
```

More about that later. Otherwise if the program is run with any arguments it suggests that you didnt want to use graphical interface and its fully useable with command line.

```
/bin/python ./vote_streams/src/main.py -r ./TEST/soc/ ./TEST/TEST.txt stv
    1 0 2
```

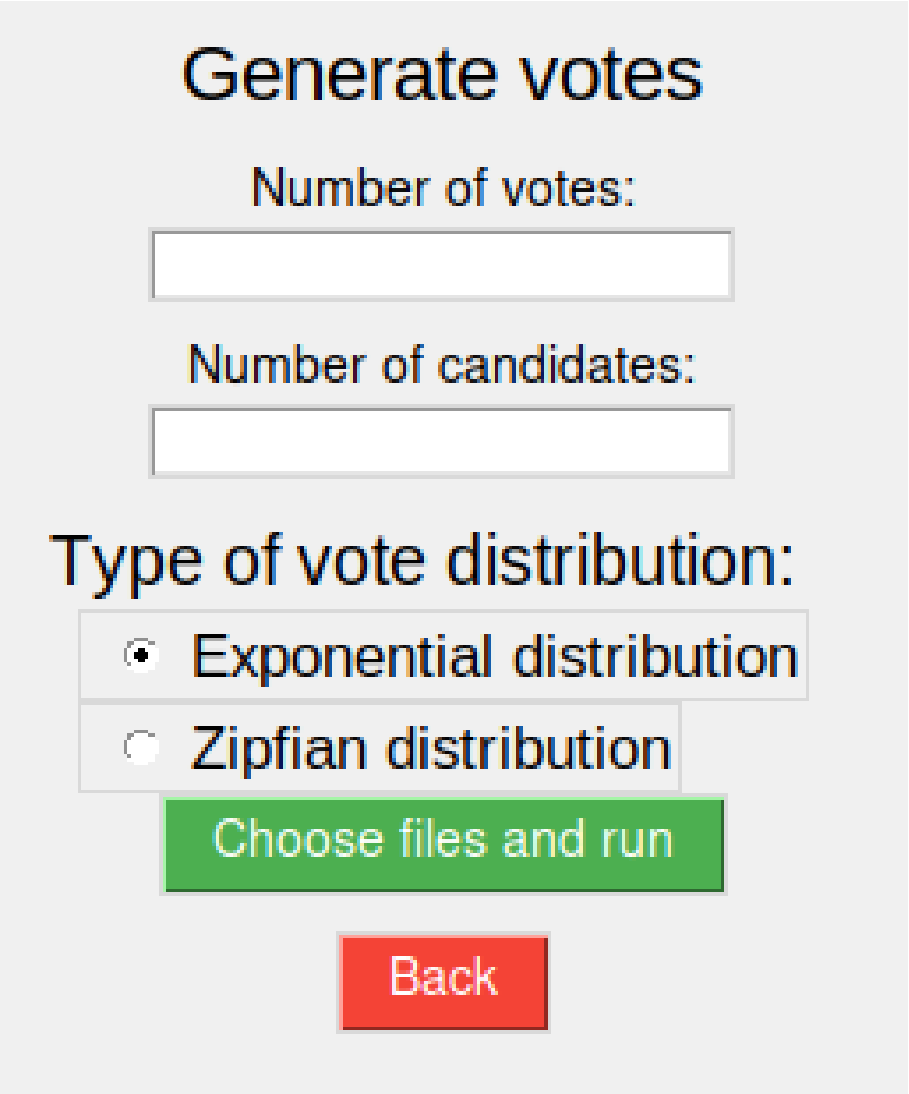
2. Graphical interface

If you open the graphical interface, this window will pop up:

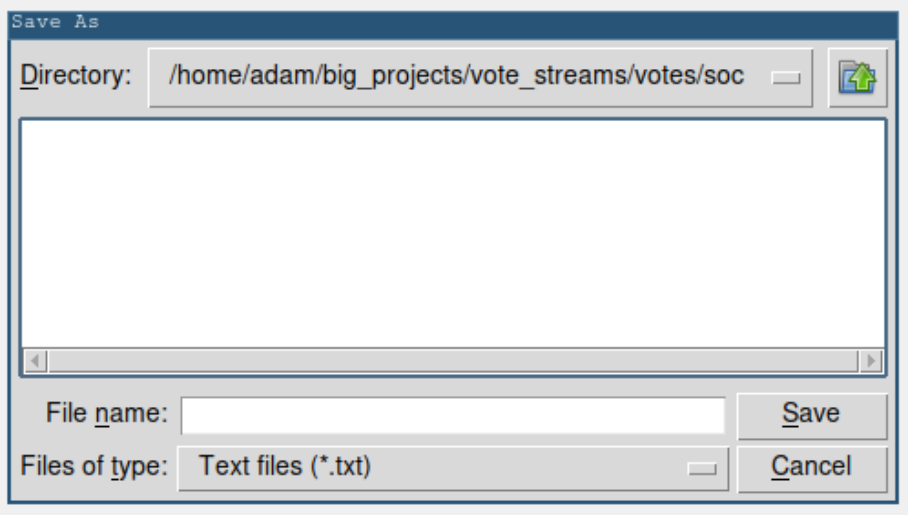


2.1 Generate votes

If you clicked in the main menu on the generate votes this window pop up:



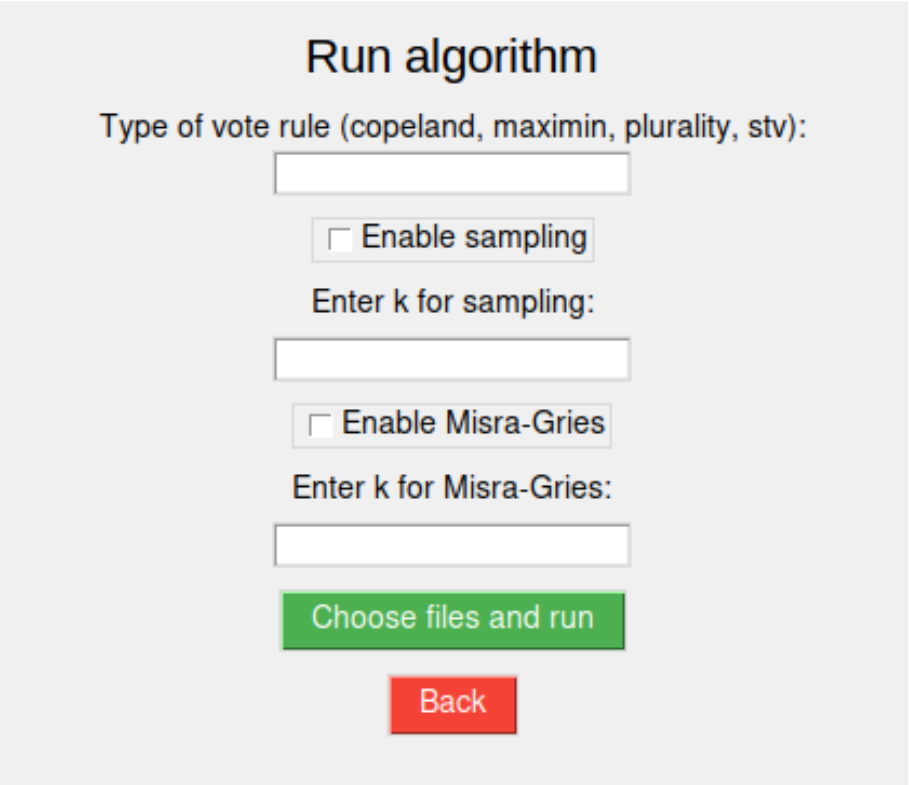
Clicking back will lead you back into main menu. If you put in all the informations and click "Choose files and run" the program will ask where it should save the generated vote file. That looks like this.



The file with generated votes will apear there.

2.2 Run algorithm

If you clicked run algorithm this will apear:



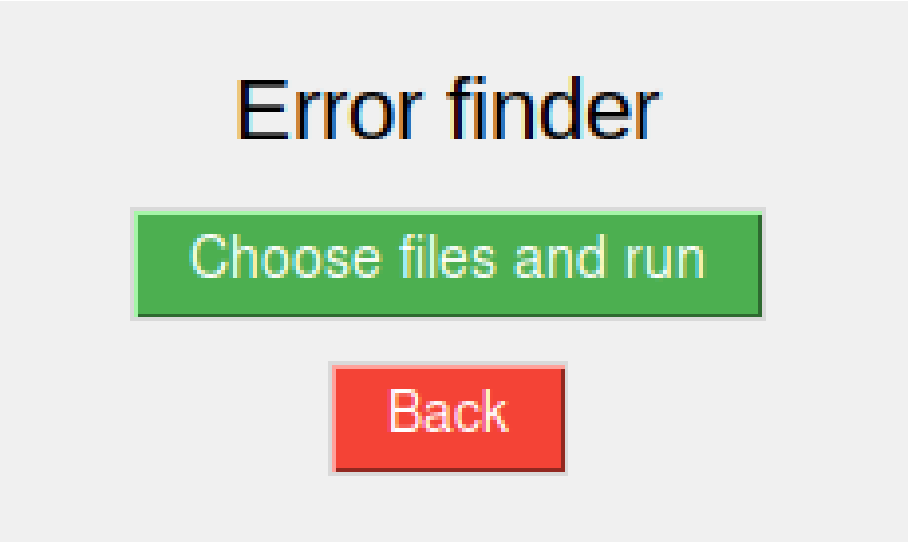
In the "Type of vote rule" you have these options:

- STV
- Plurality
- Minimax
- Copeland

The capitalization of the letters in that doesnt metter. Then you choose if you want to have enabled sampling or Misra Gries and put there the k for tham. In sampling k is that every k-th vote will be choosen, for $k = 3$ it will be number of votes // k. In Misra Gries it is the classical parametr of Misra Gries algorithm. Make sure you dont choose both sampling and Misra Gries at the same time, then the program wont let you further. Then you click "Choose files and run" same dialog as in Generate votes will appear for choosing the files, except now you can choose more files as an input. And then save file same as before. And the program will generate the winners.

2.3 Error finder

If clicked the Error finder the only thing that will apear is this:



Again just click "Choose files and run" and the program will ask you to put in the 2 files and 1 save file, choose them and the rpogram will find the error between them.

3. Command line interface

When run with wrong number of arguments the program will output the help messege which is:

```
-g/generate
-- Generate votes with given distribution and save them in file you choose.

-r/run
-- Run algorithm to determinate winner from some votes saved in vote file.

-e/error
-- Find what is error in some aproximated votes, that is you give it twofiles and it find out by some metric how far they are from each other.

-gi/graphical
-- Start graphical interface

-h/help
-- Print larger help messege.
```

It basicly works the same as Graphical interface but when -r is used than just one parametr of k is set, thats because we know that not both sampling and Misra Gries can be used so we use just one. Also when -r is used also one vote file can be given (then you can run it again on different file). If you give it just the argument (like "-e/error") another help messege will pop up with new help that guides you thru (in this example) -e arguments. Like this:

- -g/generate [save_path] [num_votes] [num_candidates] [generate_distribution].
- -r/run [load_path] [save_path] [vote_type] [sampling_enable] [misra_enable] [k]
  - for the [sampling_enable] [misra_enable] write 1 for true and 0 for false.
  - k is used for either divide the number of votes when sampling for example k = 2 is "half of the votes" or for misra gries as a parametr.
  - Optionaly the [sampling_enable] [misra_enable] [k] can be ignored if you dont plan using them.
- -e/error [load_path1] [load_path2] [save_path]
  - [load_path1] is the pathe that will be transformed into [load_path2].