Getting started:

1)Installation
>git clone https://github.com/wdklotz/simulinac XXX
Use the command above to clone the latest version into the directory XXX.

2)run the sample
make shure you use pyhton 3
cd to XXX
>python simu.py [<input-file>]

3) if it works - bingo!

4)Input (*.yml)
default sample input-file is
„XXX/yml/25_09_2017_versuche_70_200MeV.yml“ This input file is loaded with lots of comments which explain how to set up input for simu.py. Be aware: its syntax is YAML which is very picky with indentation!
edit *filepath* at the end of simu.py if you want other default input files.

5)Features
the new version is loaded with new features:

5.1)flags:
most of them are commented. if so they are set to their default values which are shown in brackets {....}. if you want to activate uncomment! most important are:

map: **if** False the linear T3D mappings are used for cavities **else** nonlinear mapping through the cavities is used
sigma: **if** True beam size is calculated from sigma matrix formalism **else** from twiss functions
express: **if** True quadrupoles are replaced by fast implementations of thin-lense matrices **else** the thin-lens is calculated as triplet product D*KICK*D

5.2)sections:
two sections are defined in the sample LE and HE like

- [&<section alias><space><name>,&<section alias><space><name>,….]

only HE is used in the sample.

5.3)parameters:
physics and lattice parameter definitions like:

- <name>: &<parameter alias><space><value> | <value> | *<parameter alias>

some parameter names (see the sample input) are recognized by the program and used as starting conditions. they are: *aperture*, *betax_i*, *alfax_i*, *emitx_i*, *betay_i*, *alfay_i*, *emity_i*, *sigmaz_i*, *dp2p_i*, *frequency* and *Tkin*. others are of free choice to be used in element definitions

5.4)elements:
definition of elements

1st line - <ID>: &<element alias> #any ID you like
2nd line - type: <CLASS> #defines the lattice element
attribute 1 - <key>: *<parameter alias> | <value>
:

attribute N - <key>: *<parameter alias> | <value>
all attribute keys with exception **- sec**: *<section alias> are mandatory and depend on the element CLASS

5.4.1)element CLASS definitions available

I	NOP unit	
D	Drift	
QF	thick F quadrupole with **slices** option	
QD	thick D quadrupole with **slices** option	
RFG	RF gap with **mapping** options (ref. T3D and A.Shishlo)	
TTFG	TimeTransitionFactorGap (ref. A.Shishlo) **using superfish data**	
RFC	RF cavity as triple product D*RFG*D	
GAP	Simple zero length RF-gap w/o (s,dp/p) (ref. Dr.Tiede & T.Wrangler)	
M	Marker	
SD	T3D sector dipole in x-plane	
RD	T3D rectangular dipole x-plane	
WD	T3D dipole wedge x-plane	
QFth	thin F quadrupole	(only available as QF slice)
QDth	thin D quadrupole	(only available as QD slice)
QFthx	thin express F quadrupole	(only available as QF slice)
QDthx	thin express D quadrupole	(only available as QD slice)

5.4.2)CLASS „RFG“ has a **mapping** attribute which can be either „base“ or „simple“. „base“ is a simplified non-linear map assuming constant time-transition factors. „simple“ is a linear map and gives same result as T3D matrix. Both options are only active if flag **map** is activated.

5.4.3)CLASS „QF“ and „QD“ have a **slices** attribute which can take values [1,…,N]. **if** N=1 the thick T3D matrix is used **elif** N>1 the quadrupole is cut into N slices of thin-lens quadrupoles either as D*KICK*D or „express“

5.4.4)CLASS „TTFG“ has a **SFdata** attribute which has to be set to the name of the file containing the field profile data from superfish

6)segments:
a segment is defined as
- <seg name>: #any name you like
  - *<element alias>
  - *<element alias>
  :

7)lattice:
1st line - title: <any text> # **must be present as 1st line!**
lines below - [N, <seg name>,<seg name>,…,<seg name>]
lines below - [N, <seg name>,<seg name>,…,<seg name>]
:
number N expands the segment list behind N times