**WSP Packages**

* wsp: top level control loop
  + \_\_init\_\_.py: defines the logging format, using the Vinay Sajip logging pkg
* telescope: everything to control the telescope
  + pw1000.py
    - has the PWI4 class from PW
* control:
  + observingMode.py
    - has sections for each observing mode
* power
  + power.py
    - commands and classes to query and command subsystem power
* status
  + status\_struct.py
    - builds the data frame for the system data logger
    - writes the dirfile to store the data
* watchdog
  + watchdog.py
* config
  + PDUi.ini
    - ini config file which has the information about the power strip IP and plugs
  + iptables.ini
    - List of all the IP addresses of the critical systems

**systemControl.py**

Order of operations:

1. Initialize control class
   1. Get config file
   2. Get base directory
   3. Set observing mode from wsp command line input
2. create and all class objects (ie make instances of each class)
   1. telescope = telescope.PWI4()
   2. pdu1 = power.PDU()
   3. dome = housekeeping.site.dome()
   4. camera1 = camera.camera()
3. Initialize the systems (could be part of above function like in Minerva)
   1. telescope.initialize()
   2. pdu1.sendStatus([initial status])
   3. etc.

**weather**

* separate weather rules if dome is open vs dome is closed
* if it’s rained in the last hour don’t open
  + track a time of last precipitation
* if it’s snowed in the last 24 hours, don’t open
  + track last precipitation and temp < 1C