# NSS ENGINE:

* Starting UI: sim->startUI();
* Setting up parameters: sim->setParameters();
* Declaring timestep: sim->setNoOfSteps();
* Performing timesteps: sim->run();
* Print status: sim-> printStat();
* Export status: sim->exportStat();
* Import status: sim->importStat();

# Simulation Engine:

* Engine Instance: sim
* startUI();
* newWorld(); - generate a new world based on parameters.
  + Set Simulation Parameters
    - setSeed();
    - setSize();
    - setorganismCount();
    - setFoodCount();
    - setGlobalReach();
  + Print World Parameters: world->printParams();
  + Initialize world
    - world->spawnOrganisms();
    - world->spawnFood();
* run():
  + Forloop with timestep();
* timestep();
  + world->calculateOFDistances();
  + world->calculateODDistances();
  + world->organismVisionReach();
  + world->OOInteractions();
  + world->OFInteractions();
  + world->organismDecisionAction();
  + world->runMiscTasks();
  + world->cleanup();
* int seed;
* float sizeX,sizeY;
* intOrganismCount;
* intFoodCount;
* floatGlobalReach;

# World:

* World Instance: world
* getDistance(Organism a, Organism b);
* getDistance(Organism a, Food b);
* printParams();
  + No. of organisms
  + No.
* printStats();
  + No. of deaths
  + No. of births
  + No. of food eaten
  + No of fights
* printOrganismStats();
* randomizeOrganismPosition(); //integrate in sim->spawnOrganism();
* randomizeFoodPosition(); //integrate in sim->spawnOrganism();
* calculateOODistance();
  + Calculate all OODistances and store in a vector
* calculateOFDistance();
  + Calculate all OFDistances and store in a vector
* oraganismVisionReach();
  + For each organism call the organism->visonReachCheck();
* organismDecision();
  + Based on items in vision, put organism into state.
  + If organism energy < 50 organism->state.isHungry = true;
  + If organism can see mate & energy > 25 organism->state.wantMate = true;
  + Chase mate. – regardless of enemy / food.
  + If organism can see enemy -> organism->estimateStrenght()
  + If estimated strength + distance -> flee / chase.
* OOInteractions():
  + If Organism within reach
  + Mate
    - Produce offspring
    - State-canMate = false -Do not know
  + Fight
* OFInteractions();
  + If food within reach consume food
* randomFoodTick();
  + Randomly generate food.

# Organism