CSE3117/CSE3216 SIMULATION AND MODELLING PROJECT CONTENTS

Analyze and report your projects according to the following conditions.

1. Simulation Program/Tool/Compiler/Framework:
   pygame (python library module)

2. Simulation/Programing Language:
   Python (programming language)

3. Scenario of the application
   the program places determined range of gas molecules at their location
   and calculates the next behaviour of every gases action (like collision,heat transmission or acceleration etc.) compared to previous behaviours with monte carlo simulation.

4. Defining the goals of the project
   program helps us to understand of how gases act on different states like how elastic collision occur or when there is heat transfer,how gases acceleration change. dumps the outputs.

5. System definition
   a. Entities
      gas molecule(s)
      system (display screen)
      closed subsystem(lines; box)
      gases initial acceleration,temperature,location etc.

   b. Attributes
      #some important attributes

      gas molecule(object)
         -mass
         -temperature
         -acceleration
         -velocity
         -radius of molecule
         -color
         -x location
         -y location

      line(object)
         -x1, y1 location (start point)
         -x2 ,y2 location (end point)
         -color

   c. Activities

gas molecules moves calculates
gas molecules acceleration reverses when bounces off the corner
distancea calculates between all molecules

d. Events

collision occurs between two molecules
heat transfer occurs between two molecules
acceleration change occurs between two molecule (elastic collision)
molecule hits the corners

e. State variables

-all molecules locations
-checks if molecules hit anything according to locations
-average temperature of sample molecules (kelvin)
-instantenous heat transfers
-average entropy changes of molecules (distance)
-elapsed time (seconds)

6. System behavior

7. How it works

a. Formulation

-Molecules positioning random location on determined range.
-Then its acceleration calculates according the formula:
acceleration= sqroot(3*R*T/m) * [0,..,1] and results assings to velocity

i.

b. Methods

move()
object_collides()
mass_distribuiton_entropy()
generateMolecules_at_special_point()
calc_average_T()
register()

c. Algorithms

object_collides() -> checks if ;firstly, if molecule hits any line or reached to screens end, secondly a molecules distance to any molecule on given lists.

İf distance of molecule is smaller than sum of itself and other molecules radius,assume as they collided;
swap molecules acceleration,

calculate average temperature difference and return, increase/decrease temperature of 2 molecules.

if molecule hits line(assume as wall);
reverse the molecules acceleration vector.(bounce)

molecule: move() -> increases the molecules x,y positions by acceleration vector, respectively

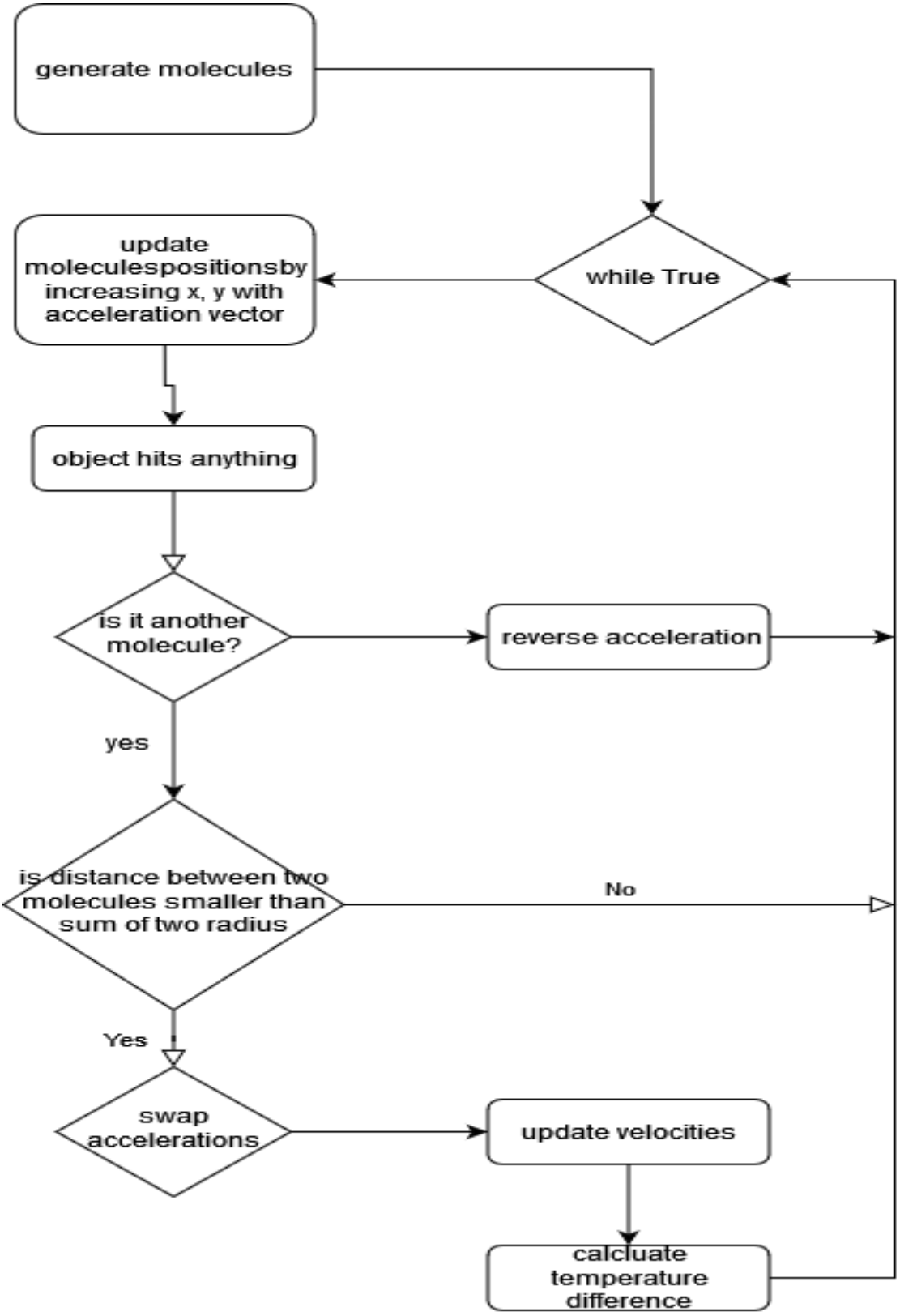
mass_distribuiton_entropy() -> takes entropy from return_entropy() method ,then interpret that value as display object.

generateMolecules_at_special_point() -> places arbitrary randomly accelerated molecules at given special position,color,radius,temperature.

return_entropy() -> sums the x-axis of all molecules on a given list, then returns it

calc_average_T() -> calculates average temperature of molecules on given list.
Register() -> dumps number of molecules,temperature,entropy,time to the file.

8. System model


generate molecules
while True
update moleculespositionsby increasing x, y with acceleration vector
object hits anything
is it another molecule?
reverse acceleration
yes
is distance between two molecules smaller than sum of two radius
No
Yes
swap accelerations
update velocities
calcluate temperature difference

9. Validation and Verification

used 2 sample molecules(temperature is different) on different initial starting sides.

used 2 different molecules(temperature and mass is different) on different initial starting sides.

used 4 different molecules (temperature and mass is different) on different initial starting points.

I verified that simulations output commonly successful on different states.

But according my researches over gases behaviour, its not validate for real-life systems behavior. Real-life systems have much more different entities.

10. Project results(opinions)

According to my researches, i found that the motion of gas molecules is not act on a straight line,rather it is orbital. And collision between the molecules do not as we think. Two molecules applies each other electromagnetic force and their motion vector changes before they touch each others. And there is many different external variables on real-life that affect gases motion.

As a result; system is not much efficient to explain real-life gas behavior but its possible outcomes helps us to determine what will happen according to previous state. But on the conservative real-life systems ,for example pressure cooker, its outouts satisfying enough

its results can be used for lecturer over simple physics on conservative system,Thus learner can interpret possible behavior of gas over simple-solid physics.