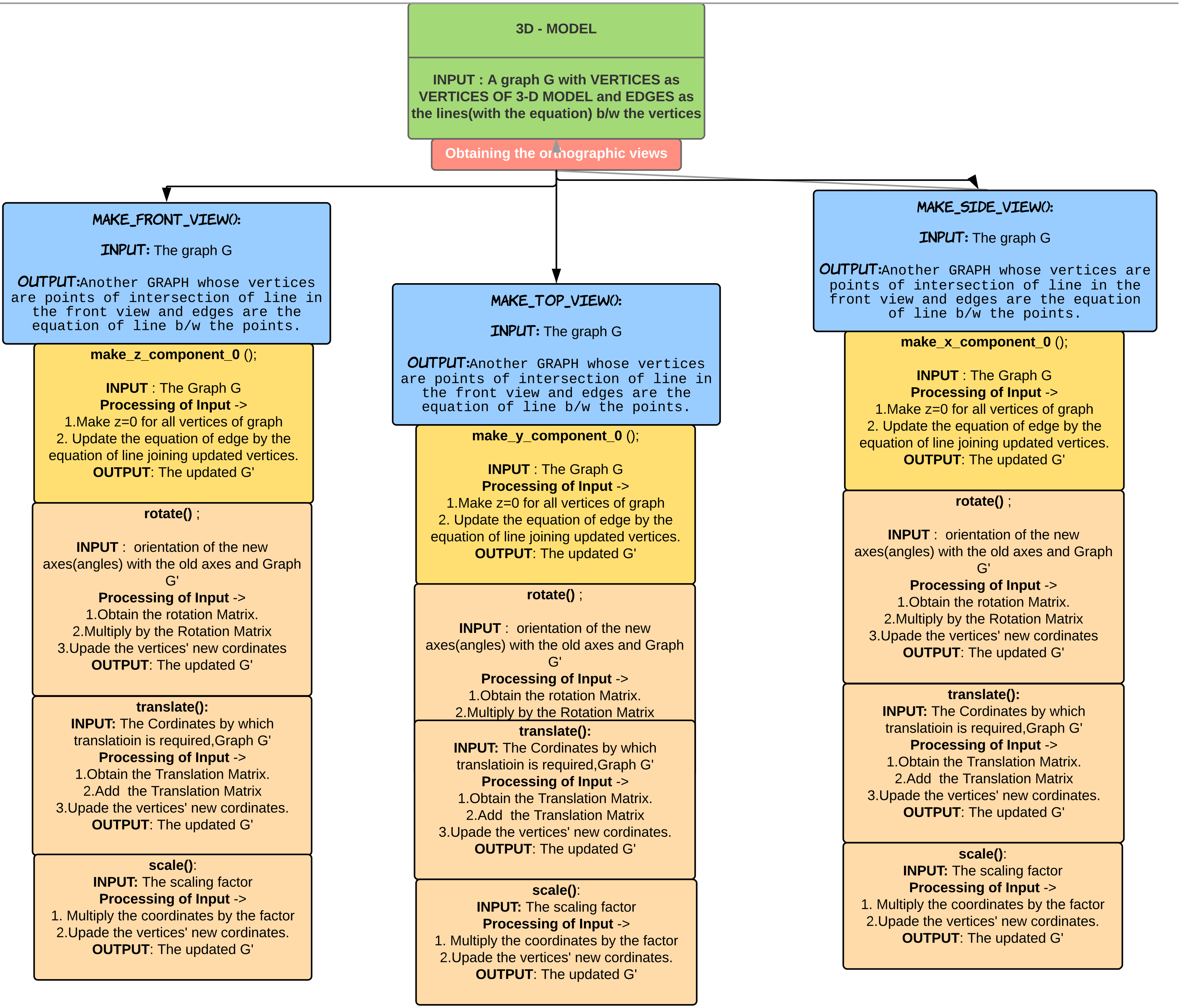

3D - MODEL

INPUT : A graph G with VERTICES as VERTICES OF 3-D MODEL and EDGES as the lines(with the equation) b/w the vertices

Obtaining the orthographic views

MAKE_FRONT_VIEW():

INPUT: The graph G

OUTPUT: Another GRAPH whose vertices are points of intersection of line in the front view and edges are the equation of line b/w the points.

make_z_component_0 ();

INPUT : The Graph G
Processing of Input ->
1.Make z=0 for all vertices of graph
2. Update the equation of edge by the equation of line joining updated vertices.
OUTPUT: The updated G'

rotate() ;

INPUT : orientation of the new axes(angles) with the old axes and Graph G'
Processing of Input ->
1.Obtain the rotation Matrix.
2.Multiply by the Rotation Matrix
3.Upade the vertices' new cordinates
OUTPUT: The updated G'

translate():
INPUT: The Cordinates by which translatioin is required,Graph G'
Processing of Input ->
1.Obtain the Translation Matrix.
2.Add the Translation Matrix
3.Upade the vertices' new cordinates.
OUTPUT: The updated G'

scale():
INPUT: The scaling factor
Processing of Input ->
1. Multiply the coordinates by the factor
2.Upade the vertices' new cordinates.
OUTPUT: The updated G'

MAKE_TOP_VIEW():

INPUT: The graph G

OUTPUT: Another GRAPH whose vertices are points of intersection of line in the front view and edges are the equation of line b/w the points.

make_y_component_0 ();

INPUT : The Graph G
Processing of Input ->
1.Make z=0 for all vertices of graph
2. Update the equation of edge by the equation of line joining updated vertices.
OUTPUT: The updated G'

rotate() ;

INPUT : orientation of the new axes(angles) with the old axes and Graph G'
Processing of Input ->
1.Obtain the rotation Matrix.
2.Multiply by the Rotation Matrix

translate():
INPUT: The Cordinates by which translatioin is required,Graph G'
Processing of Input ->
1.Obtain the Translation Matrix.
2.Add the Translation Matrix
3.Upade the vertices' new cordinates.
OUTPUT: The updated G'

scale():
INPUT: The scaling factor
Processing of Input ->
1. Multiply the coordinates by the factor
2.Upade the vertices' new cordinates.
OUTPUT: The updated G'

MAKE_SIDE_VIEW():

INPUT: The graph G

OUTPUT: Another GRAPH whose vertices are points of intersection of line in the front view and edges are the equation of line b/w the points.

make_x_component_0 ();

INPUT : The Graph G
Processing of Input ->
1.Make z=0 for all vertices of graph
2. Update the equation of edge by the equation of line joining updated vertices.
OUTPUT: The updated G'

rotate() ;

INPUT : orientation of the new axes(angles) with the old axes and Graph G'
Processing of Input ->
1.Obtain the rotation Matrix.
2.Multiply by the Rotation Matrix
3.Upade the vertices' new cordinates
OUTPUT: The updated G'

translate():
INPUT: The Cordinates by which translatioin is required,Graph G'
Processing of Input ->
1.Obtain the Translation Matrix.
2.Add the Translation Matrix
3.Upade the vertices' new cordinates.
OUTPUT: The updated G'

scale():
INPUT: The scaling factor
Processing of Input ->
1. Multiply the coordinates by the factor
2.Upade the vertices' new cordinates.
OUTPUT: The updated G'