```java
 1 package core;
 2
 3 import java.awt.Component;
11
12 public class ParmenidianTruthManager {
13
14     private ModelManager modelManager;
15     private ExportManager exportManager;
16
17     public ParmenidianTruthManager(){
18
19
20         modelManager = new ModelManager();
21         exportManager = new ExportManager();
22
23     }
24
25     public void clear(){
26
27         modelManager.clear();
28     }
29
30     public String getTargetFolder(){
31
32         return modelManager.getTargetFolder();
33
34     }
35
36     public void stopConvergence(){
37
38         modelManager.stopConvergence();
39
40     }
41
42     public void saveVertexCoordinates(String projectIni) throws IOException{
43
44         modelManager.saveVertexCoordinates(projectIni);
45
46     }
47
48     public void setTransformingMode(){
49
50         modelManager.setTransformingMode();
51
52     }
53
54     public void setPickingMode(){
55
56         modelManager.setPickingMode();
57
58     }
59
60     public void visualize(VisualizationViewer< String, String> vv,String
   projectIni,String targetFolder,int edgeType) throws IOException {
61
62         modelManager.visualize(vv,projectIni, targetFolder, edgeType);
63     }
64
65     public Component loadProject(String sql,String xml,String graphml, double
   frameX,double frameY,double scaleX,double scaleY,double centerX,double
   centerY,String targetFolder,int edgeType) throws Exception{
66
67         return modelManager.loadProject( sql, xml, graphml,  frameX, frameY,
   scaleX, scaleY, centerX, centerY, targetFolder, edgeType);
```

```
 68
 69     }
 70
 71     public void createTransitions(File selectedFile) throws Exception{
 72
 73         exportManager.createTransitions(selectedFile);
 74
 75     }
 76
 77     public void createPowerPointPresentation(ArrayList<String> FileNames,String
    targetFolder,String projectName) throws FileNotFoundException, IOException{
 78
 79         exportManager.createPowerPointPresentation(FileNames, targetFolder,
    projectName);
 80
 81     }
 82
 83     public void createVideo(File file) throws IOException{
 84
 85         exportManager.createVideo(file);
 86     }
 87
 88
 89
 90     public Component refresh(double forceMult, int repulsionRange) {
 91
 92         return modelManager.refresh(forceMult,repulsionRange);
 93     }
 94
 95     public void calculateMetrics(String targetFolder,ArrayList<Metric_Enums>
    metrics) throws FileNotFoundException {
 96
 97
 98         for(int i=0;i<metrics.size();i++)
 99             if(metrics.get(i)==Metric_Enums.VERTEX_IN_DEGREE)
100                 modelManager.generateVertexInDegreeReport(targetFolder);
101             else if (metrics.get(i)==Metric_Enums.VERTEX_OUT_DEGREE)
102                 modelManager.generateVertexOutDegreeReport(targetFolder);
103             else if (metrics.get(i)==Metric_Enums.VERTEX_DEGREE)
104                 modelManager.generateVertexDegreeReport(targetFolder);
105             else if (metrics.get(i)==Metric_Enums.VERTEX_BETWEENNESS)
106                 modelManager.generateVertexBetweennessReport( targetFolder);
107             else if (metrics.get(i)==Metric_Enums.CLUSTERING_COEFFICIENT)
108                 modelManager.generateClusteringCoefficientReport( targetFolder);
109             else if (metrics.get(i)==Metric_Enums.EDGE_BETWEENNESS)
110                 modelManager.generateEdgeBetweennessReport(targetFolder);
111             else if (metrics.get(i)==Metric_Enums.GRAPH_DIAMETER)
112                 modelManager.generateGraphDiameterReport(targetFolder);
113             else if (metrics.get(i)==Metric_Enums.NUMBER_OF_VERTICES)
114                 modelManager.generateVertexCountReport(targetFolder);
115             else if (metrics.get(i)==Metric_Enums.NUMBER_OF_EDGES)
116                 modelManager.generateEdgeCountReport(targetFolder);
117             else if (metrics.get(i)==Metric_Enums.NUMBER_OF_CONNECTED_COMPONENTS)
118                 modelManager.generateConnectedComponentsCountReport(targetFolder);
119             else if(metrics.get(i)==Metric_Enums.NUMBER_OF_VERTICES_IN_GCC)
120                 modelManager.generateVertexCountReportForGCC(targetFolder);
121             else if(metrics.get(i)==Metric_Enums.NUMBER_OF_EDGES_IN_GCC)
122                 modelManager.generateEdgeCountReportForGCC(targetFolder);
123
124 //      modelManager.getArticulationVertices();
125
126
127
128     }
```

```
129
130
131
132
133 }
134
```