ARTIFICIAL INTELLIGENCE

ANGRY BIRDS – FINISHERS

Code Snippets for Submission-6

1. Implementation of **isConnected** method which returns true if two blocks are connected else returns false. It uses basic mathematical equation of a straight line of the form
   4 Corner points of one block are substituted in the 4 line equations of other block.

   Y – y1 = m (x – x1)

```
int xx, yy;
for (int i = 0; i < 4; i++) {
xx = Pts2[i].x;
yy = Pts2[i].y;
for (int j = 0; j < 4; j++) {
if ((ltop1.x - rtop1.x) * (yy - rtop1.y) == (ltop1.y - rtop1.y) * (xx - rtop1.x) ||
(ltop1.x - lbottom1.x) * (yy - lbottom1.y) == (ltop1.y - lbottom1.y) * (xx - lbottom1.x) ||
(rbottom1.x - lbottom1.x) * (yy - lbottom1.y) == (rbottom1.y - lbottom1.y) * (xx - lbottom1.x) ||
(rbottom1.x - rtop1.x) * (yy - rtop1.y) == (rbottom1.y - rtop1.y) * (xx - rtop1.x)) {
return true;
}
}
```

2. Method for determining sub-structures from the whole structure.
   ➔ Creates separate structures of type sub-structures and the blocks are divided accordingly of that type.

```
public void createSubStructures(){


for(int i=0;i<objects.size();i++){
ABObject o=objects.get(i);
if(i==0){
SubStructure ss=new SubStructure();
```

```
ss.add(o);
list.add(ss);
continue;
}
ArrayList<Integer> con=new ArrayList<Integer>();
for(int j=0;j<list.size();j++){
SubStructure ss1=list.get(j);
for(int k=0;k<ss1.obj.size();k++){
ABObject o1=ss1.obj.get(k);
//System.out.println(this.isconnected(o,o1));
if(this.isconnected(o,o1)){
con.add(j);
break;
}
}
}
if(con.size()==0){
SubStructure ss2=new SubStructure();
ss2.add(o);
list.add(ss2);
continue;
}
SubStructure temp=list.get(con.get(0));
for(int j=1;j<con.size();j++){
SubStructure temp1=list.get(con.get(j));
for(int k=0;k<temp1.obj.size();k++){
temp.add(temp1.obj.get(k));
}
}
for(int j=1;j<con.size();j++){
list.remove(con.get(j));
}
}
}
}
```