

```

1  package ch.hevs.gdx2d.lunar.physics;
2
3  import com.badlogic.gdx.math.Vector2;
4
5  import ch.hevs.gdx2d.lunar.main.PolygonWorking;
6
7  public class Ground {
8
9      private Vector2[] polyPoints = new Vector2[Constants.SCALE];
10     private PolygonWorking groundPoly;
11
12     public Ground() {
13         for (int i = 0; i < polyPoints.length; i++) {
14             if (i == 0) {
15                 polyPoints[i] = new Vector2(0, 100);
16             } else if (i == polyPoints.length - 1) {
17                 polyPoints[i] = new Vector2(800, 100);
18             } else if (i == Constants.FLAT_ZONE + 1) {
19                 polyPoints[i] = new Vector2(800 / polyPoints.length * i,
20                 polyPoints[i - 1].y);
21             } else if (polyPoints[i - 1].y <= Constants.MIN_ALTITUDE) {
22                 polyPoints[i] = new Vector2(800 / polyPoints.length * i,
23                 (float) (polyPoints[i - 1].y + Math.random() *
24                 Constants.MAX_INCLINE * 2));
25             } else {
26                 polyPoints[i] = new Vector2(800 / polyPoints.length * i, (float)
27                 (polyPoints[i - 1].y
28                 + Math.random() * Constants.MAX_INCLINE * 2 -
29                 Constants.MAX_INCLINE));
30             }
31         }
32         groundPoly = new PolygonWorking(polyPoints);
33     }
34
35     public PolygonWorking getPolygon() {
36         return groundPoly;
37     }
38
39     public Vector2 getPolyPoint(int i) {
40         if (i >= polyPoints.length) {
41             i = 0;
42         }
43         return polyPoints[i];
44     }
45 }

```