Union of convex polygons algorithm

This algorithm takes a set of overlapping convex polygons and then returns the union of all of these polygons. Creating a convex polygon ensures that there are no long and thin artifacts pointing inside the polygon.

- description of result of union
- sharp corners are highly unlikely

```
Polygon create_random_polygon (int max_number_of_points, int number_of_polygons) {
  List convex_polygons;
  for (int i = 0; i < number_of_polygons; ++i) {
    convex_polygons.push_back(create_random_convex_polygon(max_number_of_points));
  }
  return union_of(convex_polygons);
}
```

A random convex polygon can be easily created by calculating the convex hull of a set of randomly generated points.

- ommit the details of the calculation

Using this simple approach, the resulting convex polygon will have a random amount of vertices since some of the generated points may lie on the interior of the convex hull. Thus, only a maximum number of vertices can be specified for that polygon.

```
Polygon create_random_convex_polygon (int max_number_of_points) {
  List vertices = create_random_points(max_number_of_points);
  Polygon polygon = create_convex_hull(vertices);
  return polygon;
}
```

- Polygons created by this algorithm will look too regular
- All polygons lay on top of each other.
- By adding more and more points the result will lead to a circle-like shape.
- Assuming the generated random points are constructed uniformly on a disk
- TODO insert an image

Shifting generated polygons

- To avoid this problem, after generating a polygon it will be shifted.
- The shift is based on the position of the previous polygon and an additional vector with a random direction and a fixed length.
- Therefore, a new parameter needs to be introduced, that is the fixed length of the vector.
- If the length is smaller the resulting polygon will be more round.
- The convex polygons will be closer toghether and overlap a lot.
- Increasing the distance will lead to bigger gaps.
- It also increases the risk of a polygon being completely disjunct from the other polygons.
- TODO find a way to deal with those cases

```
Polygon create_random_polygon (int max_number_of_points, int number_of_polygons,
float vector_length) {
  List convex_polygons;
  for (int i = 0; i < number_of_polygons; ++i) {
    Polygon polygon = create_random_convex_polygon(max_number_of_points)
    polygon.shift(create_random_vector(vector_length))
    convex_polygons.push_back(polygon);
  }
```

```
  return union_of(convex_polygons);
}

Vector create_random_vector(float vector_length) {
  TODO
}
```