

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css">
    <script src="jquery-3.1.0.js"></script>
    <title>To do list</title>
  </head>
  <body>
    <div id = "field"></div>
    <script type="text/javascript">

      var population = [];

      for(var i = 0; i < 20; i++){
        population.push(new Array());
        for(var j = 0; j < 1000; j++){
          population[i].push(randomDirection());
        }
      }

      function randomDirection(){
        return Math.floor(1 + Math.random() * 100 % 4);
      }

      for(var i = 0; i < 20; i++){
        population.fitness = fitness(population[i]);
      }

      function fitness(chromosome){
        var currentLocation = {
          x : 500,

```

```

        y : 500
    }

    for(var i = 0; i < 1000; i++){
        chromosome[i] === 1 ? currentLocation.y-- :
        chromosome[i] === 2 ? currentLocation.y++ :
        chromosome[i] === 3 ? currentLocation.x++ :
        chromosome[i] === 4 ? currentLocation.x-- :
        chromosome[i] = 0;
    }

    return Math.abs(800 - currentLocation.x) + Math.abs(800 -
currentLocation.y);

}

population = population.sort(function(first, second){
    return (first.fitness > second.fitness) - (second.fitness > first.fitness);
});

function mutation(chromosome){
    var result = [];
    for(var i = 0; i < chromosome.length; i++){
        if(Math.random() < 0.01){
            result.push(1 + Math.random() * 100 % 4);
        }else{
            result.push(chromosome[i]);
        }
    }
    return result;
}

```

```

function singlePointCrossover(chromosom1, chromosom2){
    var point = Math.floor(Math.random() * 600 + 200);

    var returnChoromosoms = [], returnChoromosom1 = [],
returnChoromosom2 = [];

    for(var i = 0; i < point; i++){
        returnChoromosom2.push(chromosom1[i]);
        returnChoromosom1.push(chromosom2[i]);
    }

    for(var i = point; i < 13; i++){
        returnChoromosom1.push(chromosom1[i]);
        returnChoromosom2.push(chromosom2[i]);
    }

    returnChoromosoms.push(returnChoromosom1);
    returnChoromosoms.push(returnChoromosom2);

    return returnChoromosoms;
}

var populationForEvolution = population;

while(populationForEvolution[0].fitness !== 0){
    var parents = populationForEvolution.slice(0, 10);
    var children = [];

    for(var i = 0; i < 5; i++){
        var childrenTemp =
singlePointCrossover(parents[Math.floor(Math.random() * 1000 % 10)],
parents[Math.floor(Math.random() * 1000 % 10)]);

        children.push(childrenTemp[0]);
    }
}

```

```

        children.push(childrenTemp[1]);
    }

    parents = parents.concat(children);
    console.log(parents[0].fitness);
    for(var i = 0; i < 20;i++){
        parents[i] = mutation(parents[i]);
        parents[i].fitness = fitness(parents[i]);
    }

    parents = parents.sort(function(first, second){
        return (first.fitness > second.fitness) - (second.fitness >
first.fitness);
    });
    console.log(parents[0].fitness);

    populationForEvolution = parents;
}

</script>

</body>
</html>

```