

Assignment Two

Student Name: Quin'darius Ali Lyles-Woods

Student ID: 000980796

Assignment Number: Two

Due Date: July 11th 2021

Signature: Quin'darius Ali Lyles-Woods

How to find Project

Github

<https://github.com/quinwoods/Artificial-Intelligence-.git>

Personal Website

<https://quindarius.com:3006>

Code

index.html

```
<html>
<head>
  <link rel="stylesheet" href="./css/style.css" />
  <link rel="stylesheet" href="./css/bootstrap.min.css" />

  <script type="text/javascript" src="./lib/jquery-1.7.2.min.js"></script>
  <script type="text/javascript" src="./src/bob.json"></script>
  <script type="text/javascript" src="./src/main.js"></script>
  <script type="text/javascript" src="./src/algorithm.js"></script>
  <script type="text/javascript" src="./src/utils.js"></script>
  <script type="text/javascript" src="./src/data.js"></script>
</head>
<body>
  <div class="container">
    <header>
      <h1 id="title">
        Fastest way across the United States using the Genetic Algorithm
      </h1>
    </header>
    <div class="row-fluid">
      <div id="control_buttons" class="btn-group row">
        <button id="start_btn" class="btn">Start/Restart</button>
```

```

        <button id="stop_btn" class="btn">Stop/Continue</button>
    </div>
    <div class="row">
        <span id="status" class="show-grid"></span>
    </div>
    <canvas id="canvas" width="400" height="200" style="border: 1px solid">
        your browser sucks
    </canvas>
</div>
</div>
</body>
</html>

```

main.js

```

var canvas, ctx;
var WIDTH, HEIGHT;
var points = [];
var running;
var canvasMinX, canvasMinY;
var doPreciseMutate;

var POPULATION_SIZE;
var ELITE_RATE;
var CROSSOVER_PROBABILITY;
var MUTATION_PROBABILITY;
var OX_CROSSOVER_RATE;
var UNCHANGED_GENS;

var mutationTimes;
var dis;
var bestValue, best;
var currentGeneration;
var currentBest;
var population;
var values;
var fitnessValues;
var roulette;

$(function () {
    init();
    initData();
    points = citydata;

    $("#start_btn").click(function () {
        if (points.length >= 3) {
            initData();

```

```

        GAIInitialize();
        running = true;
    } else {
        alert("add some more points to the map!");
    }
});
$("#clear_btn").click(function () {
    running === false;
    initData();
    points = new Array();
});
$("#stop_btn").click(function () {
    if (running === false && currentGeneration !== 0) {
        if (best.length !== points.length) {
            initData();
            GAIInitialize();
        }
        running = true;
    } else {
        running = false;
    }
});
function init() {
    ctx = $("#canvas")[0].getContext("2d");
    WIDTH = $("#canvas").width();
    HEIGHT = $("#canvas").height();
    setInterval(draw, 10);
    init_mouse();
}
function init_mouse() {
    $("#canvas").click(function (evt) {
        if (!running) {
            canvasMinX = $("#canvas").offset().left;
            canvasMinY = $("#canvas").offset().top;
            $("#status").text("");

            x = evt.pageX - canvasMinX;
            y = evt.pageY - canvasMinY;
            points.push(new Point(x, y));
        }
    });
}
function initData() {
    running = false;
    POPULATION_SIZE = 30;
}

```

```

    ELITE_RATE = 0.3;
    CROSSOVER_PROBABILITY = 0.9;
    MUTATION_PROBABILITY = 0.01;
    //OX_CROSSOVER_RATE = 0.05;
    UNCHANGED_GENS = 0;
    mutationTimes = 0;
    doPreciseMutate = true;

    bestValue = undefined;
    best = [];
    currentGeneration = 0;
    currentBest;
    population = []; //new Array(POPULATION_SIZE);
    values = new Array(POPULATION_SIZE);
    fitnessValues = new Array(POPULATION_SIZE);
    roulette = new Array(POPULATION_SIZE);
}

function drawCircle(point) {
    ctx.fillStyle = "#000";
    ctx.beginPath();
    ctx.arc(point.x * 3, point.y * 3, 3, 0, Math.PI * 2, true);
    ctx.closePath();
    ctx.fill();
}

function drawLines(array) {
    ctx.strokeStyle = "#f00";
    ctx.lineWidth = 1;
    ctx.beginPath();

    ctx.moveTo(points[array[0]].x * 3, points[array[0]].y * 3);
    for (var i = 1; i < array.length; i++) {
        ctx.lineTo(points[array[i]].x * 3, points[array[i]].y * 3);
    }
    ctx.lineTo(points[array[0]].x * 3, points[array[0]].y * 3);

    ctx.stroke();
    ctx.closePath();
}

function draw() {
    if (running) {
        GANextGeneration();
        $("#status").text(
            "There are " +
            points.length +
            " cities in the map, " +

```

```

        "the " +
        currentGeneration +
        "th generation with " +
        mutationTimes +
        " times of mutation. best value: " +
        ~~bestValue
    );
} else {
    $("#status").text("There are " + points.length + " Cities in the map. ");
}
clearCanvas();
if (points.length > 0) {
    for (var i = 0; i < points.length; i++) {
        drawCircle(points[i]);
    }
    if (best.length === points.length) {
        drawLines(best);
    }
}
}
function clearCanvas() {
    ctx.clearRect(0, 0, WIDTH, HEIGHT);
}

```