
Team 12

| | | |
|-------------------|------|-----------|
| Martijn Debeuf | (1h) | (270 LOC) |
| Frédéric Blondeel | (2h) | (700 LOC) |

Demo: <https://penocw.cs.kotnet.kuleuven.be:8012>

Files reviewed: delaunayTriangulation.js/Triangle.js ; Island.js ; PermutationConverter.js

1 Delaunay Triangulation

1.1 delaunayTriangulation.js

Er werd gekozen om een nieuwere syntax van javascript te gebruiken, de ";" is achterwege gelaten. Best wordt hier een algemene keuze in gemaakt die dan doorheen het hele project gebruikt wordt. In andere klassen is de oude syntax nog gehanteerd.

- In de statische functie "triangulation" wordt ook nog een lijst dezelfde naam gegeven als de functie. Beter wordt een andere naam gekozen om verwarring te vermijden.
- Test functies werden onderaan het document geplaatst, mooier zou zijn moesten deze in aparte file staan.

1.2 Triangle.js

- De if-statement lijn 19 kan vereenvoudigd worden door te schrijven:

```
return edge1.includes(edge2[0]) && edge1.includes(edge2[1])
```

2 Advanced Screendetection

- De findCorners() functie is zeer lang (ongeveer 200 lijnen code) en zou best opgesplitst worden in kleinere meer leesbare delen. De functie is ook redelijk complexe en zonder extra uitleg is deze lastig te begrijpen.
- Net zoals findCorners() is cleanCorners() moeilijk te snappen. Documentatie is bij deze 2 functies essentieel net zoals een elegantere manier om deze te schrijven.
- Er is geen geschreven of geteste reden waarom in inRangeOf() de waarde "10" gebruikt.
- Elke functie heeft een grote nood aan extra uitleg in de vorm van documentatie. De code is zeer abstract en lang dus een korte uitleg is vereist.
- PermutationConverter.js is leesbare gemakkelijke code maar documentatie mankeert.