Course Real-Time Rendering
Winter term 2019/20

Lab Class **January 20-21, 2020**

Collection of Exercises 5

Solutions due February 05, 2020

In this lab class you will verify if a given triangulation satisfies the circle criterion. Though we recommend to use C++, it is also possible to use JAVA as programming language. We suggest Visual Studio Community C++ 2019 (for Windows, it is free), Eclipse (Windows/Linux/Mac), Sublime (Windows/Linux/Mac) or build tools like *make*, *cmake*, *javac* as programming environment.

Submission and Grading

The exercise can be solved as a group of one or two students. We encourage you to thoroughly solve the exercise on your own. Plagiarism can lead to an exclusion from the course. If you have any problems or questions don't hesitate to ask your professor Rhadamés Carmona (rhadames.elias.carmona.suju@uni-weimar.de, Bauhausstr. 9a, third floor, guest room).

The solution for this exercise have to be submitted until **February 05, 2020, 11:59 pm** to our **moodle** website. The solutions will be evaluated by your professor using own test cases. Granted points will be proportional to the number of correct answers to problem instances. The grading will be announced on moodle.

For grading, your implementations will be tested on the following online compilers:

- C++ will be tested on: http://cpp.sh/
- JAVA will be tested on: https://www.jdoodle.com/online-java-compiler/

For submission, you have to implement your solutions as two .cpp or .java files (i.e. `exercise51.cpp` and `exercise52.cpp` or `exercise51.java` and `exercise52.java`). The first line(s) must include your name(s) as comment in order to document authorship(s).

As an example, if you are a group of two students and submit in C++, a submission file should be structured as follows:

```
// author: <forename> <family name>
```

```
// author: <forename> <family name>

#include <iostream>

// function definitions can be placed here

int main(){
/*
your implementation starts here
*/
return 0;
}
```

As another example, if you are a single student and submit in JAVA, a submission file should be structured as follows:

```
// author: <forename> <family name>

public class MyClass {
    public static void main(String args[]) {
/*
your implementation starts here
*/
    }
}
```

We strongly recommend to test your code on these online compilers before submission. In particular, the output format of your implementations should be exactly as it is specified in the exercise. For grading, the output will be compared line by line with an online software (e.g. https://text-compare.com/). In order to obtain full grade, the output generated by your implementation has to be identical with the expected output specified in the exercise.

Exercise 5.1

[10 points]

Implement a brute force approach to verify the circle criterion for a given triangulation. In this case, for each triangle, verify if each of the other points are outside the circumcircle.

Input The input starts with a single text line containing two integer numbers n and m, with $1 <= n, m <= 1000$. The next n lines contain the coordinates x and y (floating point) of each input point separated by one blank space. Each of the next m lines contains three 0-based point indices of one triangle. Assume there are no degenerated triangles.

Output Print `YES` if the triangulation satisfies the circle criterion. Otherwise print `NO`.

Sample input

```
4 2
3 1
1 0
3 0
2 1
1 2 3
2 0 3
```

Sample output

```
YES
```

Exercise 5.2

[10 points]

Use a simplified version of DCEL to accelerate the circle criterion. Adjacent faces (and thus, adjacent vertices) of each triangle can be accessed in constant time. Therefore, only few points are commonly tested per triangle instead of n points. You can assume that the input is a valid triangulation (non-crossing lines).