# Open3D GSOC/GSOD 2022 Coding Challenge

## Non-Disclosure Agreement

You agree to the following by participating in this coding challenge:

- You will complete the coding challenge by yourself and not rely on any other person or entity.
- You will not disclose any info about coding challenge, including solutions, to any other person or entity.

## Overview

In this coding challenge, you will find identically-colored connected components in a triangle mesh with Open3D. You will implement a function to return a list of identically-colored connected components. An identically-colored connected component consists of spatially connected vertices with the same color. In this question, a connected component is represented by a list of vertex indices.

## Example Input

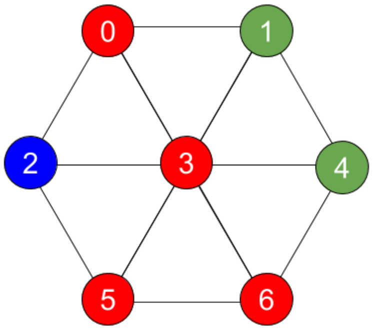

Figure 1: Example triangle mesh with colored vertices

Given the triangle mesh as show in Figure 1:

- There are 7 vertices: `[0, 1, 2, 3, 4, 5, 6]`,
- There are 6 triangles: `[(0, 2, 3), (0, 3, 1), (1, 3, 4), (2, 5, 3), (3, 5, 6), (3, 6, 4)]`.
- Each vertex has a color: `[red, green, blue, red, green, red, red]`.
- Expected output of `IdenticallyColoredConnectedComponents()`: `[[0, 3, 5, 6], [1, 4], [2]]`.

### Example C++ Solution

```cpp
#include "Open3D.h"

int main() {
    using namespace open3d;

    // Read triangle mesh "test_mesh.ply" as geometry::TriangleMesh
    ...

    // Then get the connected components
    auto connected_components = mesh.IdenticallyColoredConnectedComponents();

    // Print connected_components in the specified format
    ...

    return 0;
}
```

Listing 1: Save this to `examples/cpp/Solution.cpp`

### Example Python Solution

```python
import open3d as o3d

def main():
    # Read triangle mesh "test_mesh.ply" as o3d.geometry.TriangleMesh
    mesh = ...

    # Then get connected components
    connected_components = mesh.identically_colored_connected_components()

    # Print connected_components in the specified format
    ...

if __name__ == "__main__":
    main()
```

Listing 2: Save this to `examples/python/solution.py`

### Example Result

We write the results into `result.txt`:

- Each line in the text file represents one connected component.
- In each line, Verteices are sorted in ascending order.
- Lines are sorted in ascending order by the smallest vertex index in each list.

Example `result.txt`:

```
0 3 5 6
1 4
2
```

Listing 3: Save this to `examples/result.txt`

## Tasks

You're expected to:

1. Compile Open3D from source: http://www.open3d.org/docs/latest/compilation.html.
2. Write C++ function:
   `open3d::geometry::TriangleMesh::IdenticallyColoredConnectedComponents`.
3. Write Python binding:
   `open3d.geometry.TriangleMesh.identically_colored_connected_components`.
4. Write `examples/cpp/Solution.cpp` to read the input mesh `test_mesh.ply`, find identically-colored connected components. Change the build system so that an executable can be build.
5. Write `examples/python/solution.py` to read the input mesh `test_mesh.ply`, find identically-colored connected components and print results.
6. Output the result of task 3 or 4 (their results shall be the same) to `examples/result.txt`.
7. Write C++ and Python unit tests integrated with Open3D's unit test system.
8. Document your code, the algorithm used, how to build and run, and etc.

## Reference

Here are some links for your reference:

- Open3D repository: https://github.com/isl-org/Open3D.
- Open3D docs: http://www.open3d.org/docs.
- Pybind11: https://github.com/pybind/pybind11.
- Send an email to us.

## Submission

To submit your code:

1. Push the code to a **private** git repository, share the repository with us and ping us via email.
2. Include all the necessary files. The git history should indicate the files you changed.