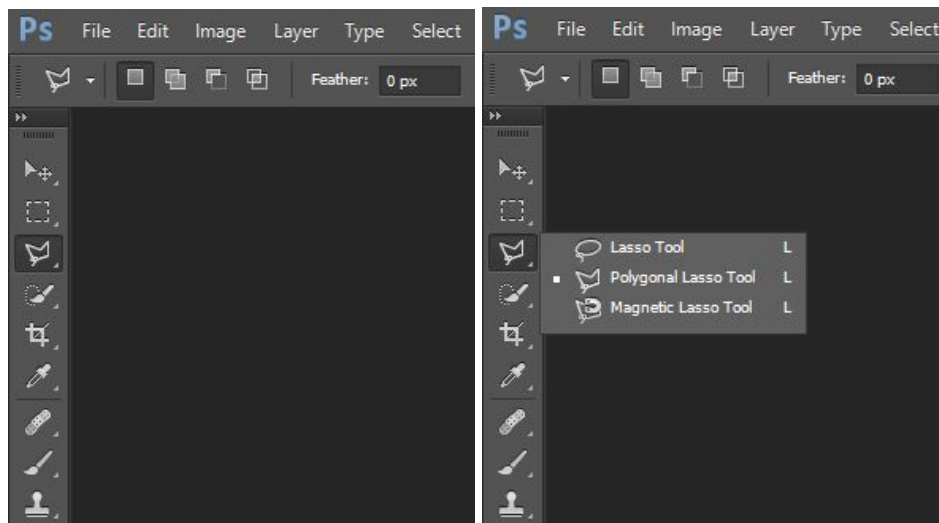


Exercise 6

Part 1: UI Patterns

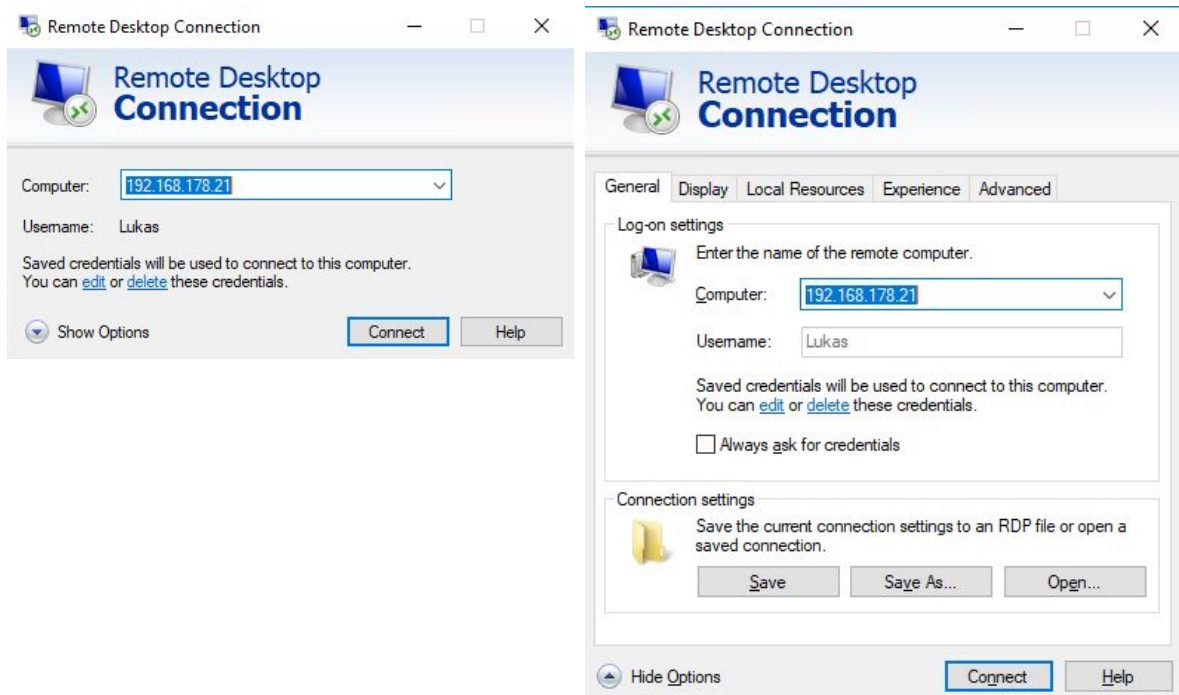
Extras on Demand in Photoshop

A left click on a tool symbol selects the tool. A right click on the same symbol reveals various different versions of the tool.



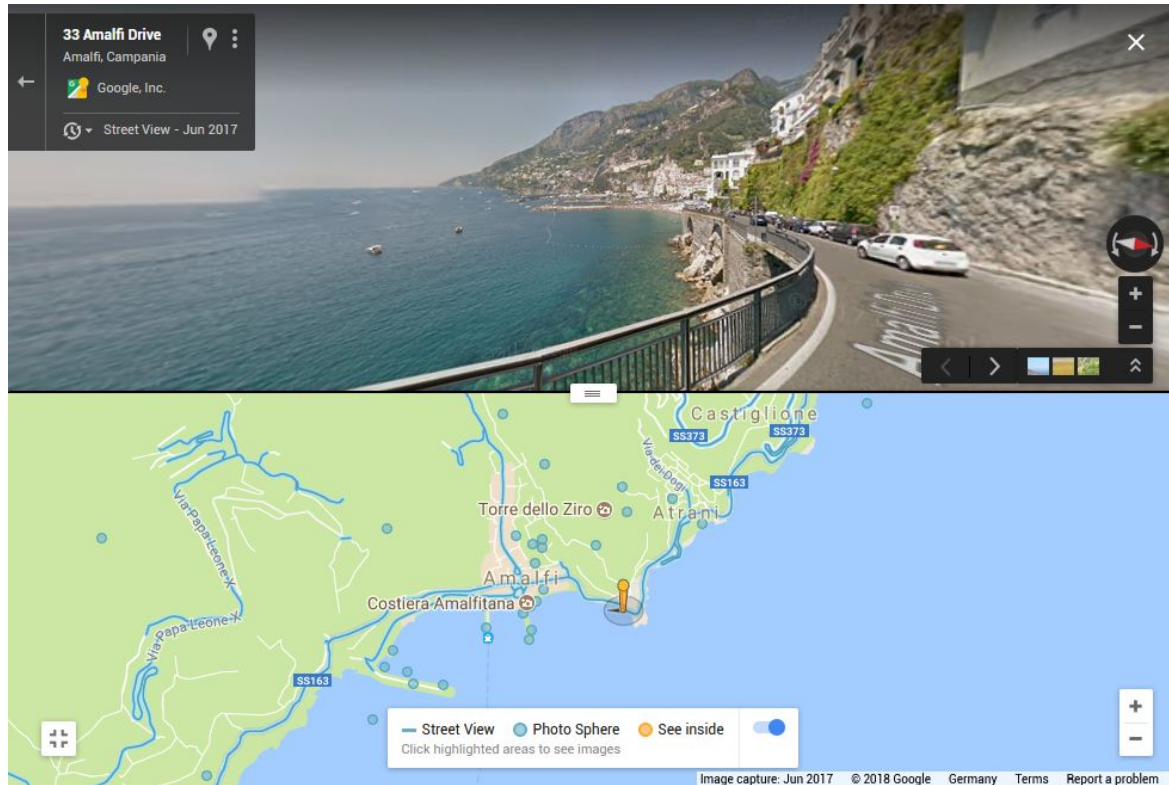
Extras on Demand in Windows' Remote Desktop

A left click on "Show Options" reveals more options!



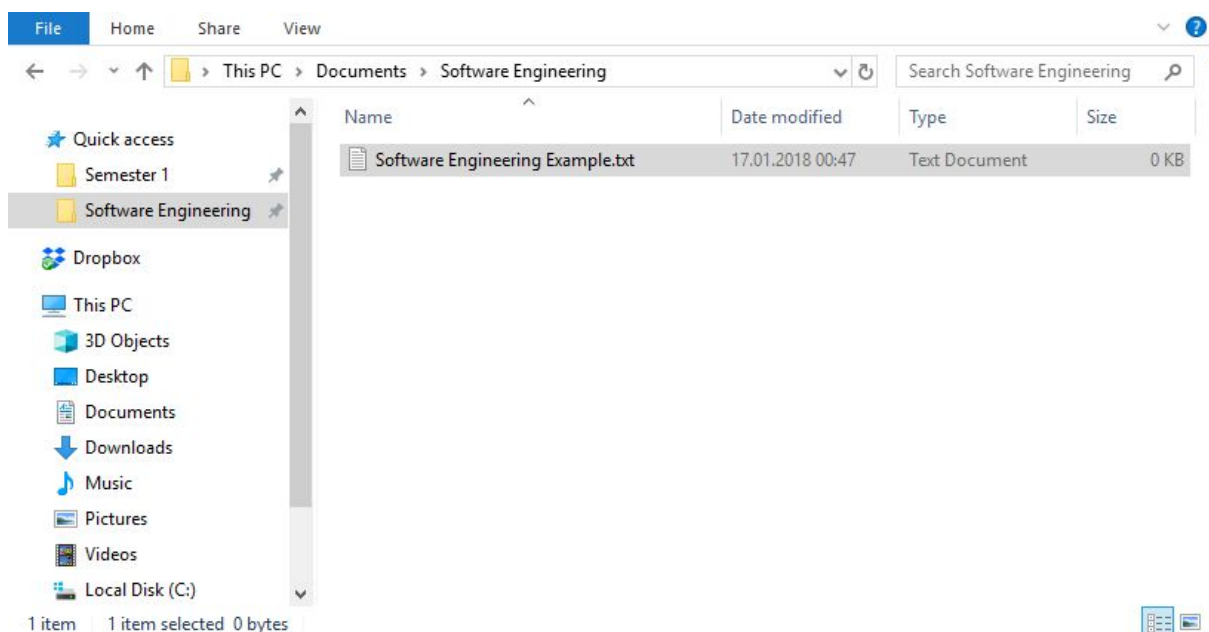
Two-Panel Selector in Google Street View

The lower panel shows the map including selectable locations and the user's position. The upper panel shows the Street View at the selected position.



Two Panel Selector in Windows Explorer

The left panel shows a folder overview. The right panel shows the content of a selected directory.



Part 3: Debugging

For debugging, we used the internal debugger of "[IntelliJ IDEA](#)". This allowed us to set breakpoints and watch variables during runtime. The mentioned line numbers reference the original DebugMe.java file which was provided.

1. Bug: Nullpointer exception

We used breakpoints to find out where the nullpointer exception occurs. We found out that in line 52 the marker is set to null, and in line 54 marker is set to marker.next(). null cannot have a .next() so the exception is thrown. We simply commented out line 52: `//marker = null;`

2. Bug: Head item in list cannot be deleted

Again, we used breakpoints and watched variables of items, especially of marker and temp, to follow the program at runtime. When the program tries to remove the head item in the list, it creates a new head item with the same value (line 42). Instead, the next item should be set as the head item. We changed line 42 to the following:

```
head = marker.next();
```

3. Bug: Program tries to print empty list -> nullpointer exception

Looking at the values of the marker item, we saw that the print method tries to print marker.value() even if marker == null. We changed `do {...} while (...)` in line 72 - 75 to `while (...) {...}`. This way, it first checks whether marker exists.

4. Bug: find method only looks at the second item in the list

We wondered, what would happen if we search for another value than 3 with the find method. We entered 2 and saw that the program got stuck. Using breakpoints, we saw that that program got stuck in the while loop of line 62 - 65. We also saw, that the program does not start searching at the head item in the list, but at the second one. Therefore we changed line 61 to the following:

```
Item<T> marker = head;
```

The while loop would also not iterate through the entire list, so we changed the loop to the following:

```
while (marker != null) {  
    if (marker.value() == value) {  
        return marker;  
    }  
    marker = marker.next();  
}
```

We also included the fixed DebugMe.java file in our Github commit.