We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 10.

Parts of Q01 were chosen to be marked for different style criteria. Thus, it is possible that other parts might have style problems that we did not address. There were also design recipe elements that we chose to give feedback on and not deduct for. You should still improve on these, since we may decide to deduct for them on a later assignment. Please review the posted solutions and style guide to help resolve any questions you may have. If that is insufficient, please raise your questions in 1-1 consulting hours.

**Question 1 (puzzle.rkt)**

**Purpose, Contract and Code Complexity**

- Many students did not explicitly reference the function parameters by name in the purpose. As a reminder, this means that the parameters that appear in your function header should be referenced by exact name in the purpose. For example: "...takes in a list of numbers [lon] and ..."

- Many students did not adequately describe the full behaviour of the function in their purpose. For example, it should be mentioned that `find-unfilled` produces `(-1 -1)` on finding no unfilled spots, and `neighbours` produces `empty`.

- Similar to above, many students did not include the edge cases in their contract. For example, the produced value of `find-unfilled` needs to represent a coordinate or our signal value: `find-unfilled : ... -> (anyof (list Nat Nat) (list -1 -1))`

- Many students used the wrong types or the wrong number of types in their contracts. For example, using `(listof Nat Nat)` instead of `(list Nat Nat)`.

- Many students forgot requirements for `try`. Recall that we must specify that there is at least one unfilled spot, and that the Nat arguments are the Row and Column of that spot, respectively.

- Many students did not adequately utilize helpers for `fill-spots`. The complexity of this function means that in many cases, the use of multiple helpers is required for the function to be readable.

- Many students used `equal?` when better comparators were available (`char=?`).

---

1