VG100
Introduction to Engineering

Homework 2

Michele & Manuel — UM-JI (Summer 2020)

- Download the homework support materials
- Always include comments in the code
- Before starting think of the program structure
- Keep testing and improving your code
- Write a single `README` file per assignment
- When altering code list your changes in a `changelog`
- Archive the files (`*.{zip|tar}`) and upload on Canvas

Ex. 1 — *Starting with Elm*

1. Reading.
   a) Read documentation, guides, examples related to tuples, records, and lists in Elm.
   b) Follow the Elm guide, starting from the beginning until the end of the section on types.
2. Exercising.
   a) Explain the difference between the three concepts and write short Elm programs highlighting how and when to use them.
   b) For lists test various more advanced functions such as `Map` or `IndexedMap`.
   c) For tuples, explain why there is a limitation to three components. Is it a limitation or a "feature". Explain.
   d) For records show how keep some record fields while changing some others.

Ex. 2 — *Tetris*

In this exercise we want to alter the Tetris game. Make sure to document all the details about the changes in the README file, and include a one line summary for each change in the changelog file.

1. Adjust the scoring system. No guidelines[1], simply decide on a new system, detail it in the README file, and implement it.
2. Allow the tetriminos to randomly change color.
3. Investigate how full lines, i.e. lines without any hole, are detected and cleared.
   a) Which files and functions are needed for detecting and detecting lines?
   b) What are `Nothing` and `Just` and how to use them? How do they help keeping the code clear and bug free?
   c) Adjust the code such that each time a line is full, not only that line is cleared but also the line below it, if it exists.

[1] Your changes should make it clear that you fully understand how the points are counted.