Introduction to Programming II 2021 - Final

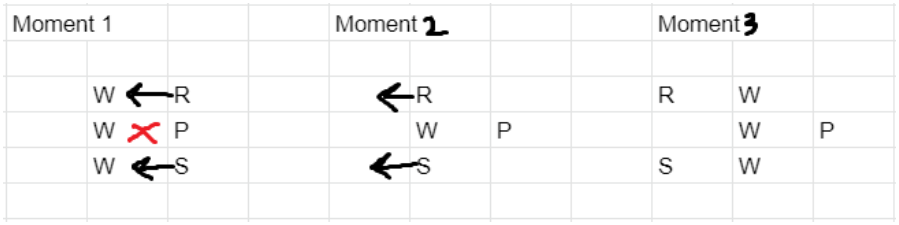
12:00 - 16:00 on 8th of May 2020

For this practical exam, you will extend Assignment 2, in which you have implemented a two-dimensional world where two players control units that battle to take each other's flag.

You are asked to choose and implement only one additional task requirement from a list. Also, implement ALL the technical requirements, which contain C++ techniques that you have seen during the course.

1. Task requirements (choose one: the first line of your code should state which one)

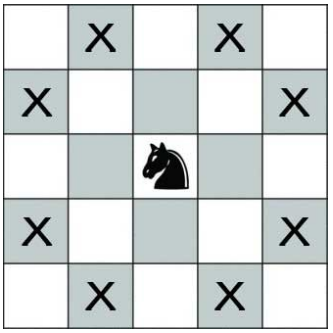
1. A random initial setup for the terrain and the units
    a. The units of one player should be grouped together, without partition
    b. A new type of terrain called lake ('L' character) composed of 7 squares, also grouped together without partition
    c. A new type of terrain called swamp ('W' character) composed of 7 squares, not necessarily grouped together
        i. Paper units cannot move through a swamp. Other units are allowed.
        ii. Example:



2. The parameter 'world' passed to each bot shows only objects (terrain or enemy units) within a 2 square distance. For example, this is what bot 1 would see (D stands for "dark"):

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
2	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
3	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
4	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
5	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
6	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
7	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
8	D	D	D	D	D	D	D	M							
9	D	D	D	D	D	D	D								
10	D	D	D	D	D	D	D			S	S	S	S	S	
11	D	D	D	D	D	D	D			P	P	P	P	P	
12	D	D	D	D	D	D	D			R	R	R	R	R	
13	D	D	D	D	D	D	D			S	S	S	S	S	
14	D	D	D	D	D	D	D			P	P	P	P	P	
15	D	D	D	D	D	D	D			R	R	R	R	R	F

3. Implement one supply center ('C' character) per bot which pops out one new unit of any type in the map every 2 seconds
    a. For this, there should be an empty cell orthogonally adjacent to the SC to place the new unit
    b. A supply center can be destroyed by any enemy unit. When this happens, the morale goes down and the player disbands any 5 units (which are afraid because they lost the SC)
    c. A supply center can move
    d. The maximum number of units (supply center and flag included) is limited to 40
4. One bomberman unit (B) can not only move like the others, but also decide to explode (instead of moving) and kill all units (including itself and its own player) in a range of 2 squares
    a. The flag is not destroyed
    b. The terrain is destroyed
    c. After the explosion, the cells are empty
    d. If player 0 moves its bomberman into player 1's bomberman, player 0 loses. And vice-versa.
        i. If two bomberman mutually move to the same empty position (nobody's fault) they automatically explode instead of bouncing back
5. Different types of movement
    a. Paper units can also move diagonally
    b. Rock units always move 2 cells instead of one
        i. If there are not two empty cells to move, it doesn't move
    c. Scissors move like knights in chess - L movement, being able to jump over other units



    d. Only once per game, the flag can swap its position with a Paper symbol

2. Assignment Requirements

a. Your entire code (to be submitted) should consist of **a single CPP file**
    i. While we recognize this is not a good practice, it will facilitate a more accurate and easier grading of tens of students
    ii. The name of the CPP file should have the form: **yourname_yoursurname.cpp**
b. The first line of your code should state the task requirement that you have chosen
c. Forgetting to identify the technical requirements using the standard **// ITEM 3.<x>. <Justification>** in the line before the implementation of the feature incurs in 0 points and no appeal

3. Technical requirements and grading schema (total of 30 points)

You have to demonstrate knowledge on each concept given to you through the course till this moment. In other words, your code should contain at least one justifiable use of each of the following concepts and requirements:

a. Implement the new task requirement (section 1) [15 points]
b. At least one class with a destructor and several constructors (default, with initializer-list, copy, move) (lecture 3) [1 point]
c. Implement operator =, +, and > (lecture 3) [1 point]
d. Static and dynamic typing (lecture 4) [2 points]
e. Virtual and pure virtual functions (lecture 4) [1 point]
f. A call to a constant member function (lecture 5) [1 point]
g. A non-copyable object (lecture 5) [1 point]
h. At least one explicit specialization (lecture 9) [2 points]

i. At least one use of decltype (lecture 9) [1 point]
j. One example of filtering a STL container using a predicate function (lecture 11) [2 points]
k. Your code should have STL containers and templates [1 point]

Two points are also given for the quality and readability of your code as perceived by the instructors. For example, naming conventions, indentation, consistency, comments before functions stating arguments, return and what the function does, quality of the visualization.

IMPORTANT: Add a one-line comment right before the part of the code in which you implemented each of the requested items (except 3.a). This will allow us to quickly search for your answers. Format of the comment:

```
// ITEM X.y: My justification of why I implemented the item here
// some code...
```

Example:

```
// ITEM 3.b: Bla bla bla…
```

Forgetting this will incur zero as the grade for the item, without appeal!

4. Notes

a. Plagiarism will incur zero as the grade of your midterm, independently of if you are the person who implemented the original code or made the copy.
b. This is an individual exam.