Introduction to Programming II 2021 - Assignment 2

Deadline: 23:59 on 7th of May 2020

In this assignment, you will implement a two-dimensional world in which two players control units that battle to take each other's flag. Regarding C++ techniques, you will have the chance of using containers, smart pointers and other features seen in the last weeks.

1. Task requirements

1. Each player (player 0 and player 1) controls a set of units:
    a. Moveable units: 10 scissors, 10 rocks, 10 papers
    b. Fixed object: flag
2. The world is composed of a 15x15 maze
    a. Initial setup:

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

    b. M symbols represent mountains, which can neither move nor be moved by players
    c. Player 0 controls the small letters, while player 1 controls the capital letters
    d. Only one unit can occupy one position in the maze at a time. Overlaps should be resolved by the game controller
3. Once per second the game controller (main method) does the following:
    a. Requests the action (movement) to each player
        i. If the player takes more than 400 ms, it loses
        ii. Example of movement: Player 0's first movement could be [6,6] → [7,6]
    b. You should implement some strategy for both players to move towards the opponent's flag
4. Movement rules
    a. A player can only move its own units (rock, paper or scissors). A flag cannot be moved.
    b. When requested by the game controller (once per second), the player can move only one unit at a time (rock, paper or scissors) to an orthogonally adjacent position within the limits of the world

c. Moving towards a mountain is an illegal move, so is moving to a position occupied by himself
d. Any unit can take the flag by moving to its position. The player is then declared the winner.
e. If two similar units (ex: s and S) move to the same position, they bounce back and both actions are cancelled. This is not an illegal move.
f. A player immediately loses if attempted to make an illegal move
g. There is killing. Examples of killing using the figure below: player 0's movement is [8,5] → [8,6] and player 1 is [9,6] → [8,6]

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

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. There should be a comment before every single method explaining what it does. Use the standard: https://developer.lsst.io/cpp/api-docs.html
c. Forgetting to identify the items using the standard **// ITEM 3.<x>. <Justification>** in the line before the implementation of the feature incurs in 0 points and no appeal

3. Grading schema (total of 15 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 all the task requirements (section 1) - 5 points
b. In the class World, the containers should have the appropriate type of smart pointers to dynamically allocated objects - 3 points
   i. You might need to create your own classes
   ii. Don't forget to use templates
   iii. Write comments in the code justifying your choices
c. One of the players should move randomly, while the other should contain some basic strategy - 3 points
d. Implement a unique game feature that no other colleague has done. For example, showing score, probability of victory, saving states of the game into a file, etc. [2 points]

Two points are also given for the quality and readability of your code as perceived by the instructors. For example, naming conventions, indentation, consistence, 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 assignment.