## 1 - Introduction

This project is based on the "*Cardinal Chains*" puzzle game, released in 2018 and developed by Daniel NORA.

It has to be realized by groups of maximum 2 students in 2 weeks, and is mostly based on what you have seen during this course but some parts may need a little research to complete. **One and only one** group of 3 is allowed **if and only if** the class has an odd number of students and all other groups have 2 members.

## 2 - Presentation

"*Cardinal Chains*" is a minimalist puzzle game centering on the concept of non-decreasing sequences. Each puzzle begins with a monochrome grid of numbers, plus a few colored cells marked with an "**x**". Starting at these cells, you must link up numbers in non-decreasing order until the whole grid is filled with color. The puzzles are carefully designed by hand, each with a unique solution. Difficulties and types of puzzles are varied, and there is no timer, nor a limited number of tries.

You have to implement a console version of this game in C, while having quite a lot of freedom in its layout. You will find a graphic version of this game, playable here, as well as a playthrough: here.

## 3 - Process

### 3.1 - Rules

In each level the player has a board with empty and full cells. The full cells are different numbers as well as starting points symbolized by "**x**". The goal is to go through **all** the numbers in ascending (or at least identical) order, from the starting points, by creating chains of directly adjacent numbers (no diagonals). Each cell can only be traveled once, so it is not possible to create intersections or overlaps. The levels can have one or more final cardinal chains, depending on the number of starting points.

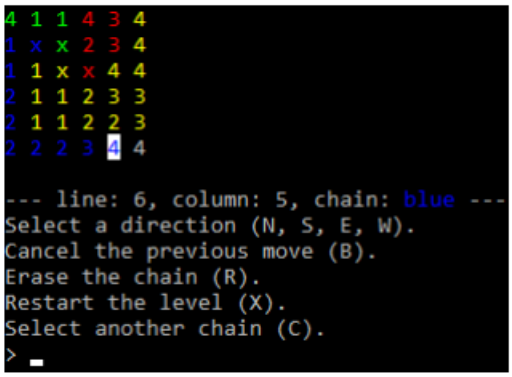
### 3.2 - Game

You will provide the player with a game experience that will automatically progress from one level to the next based on completion.

In each level, the player will have the possibility to grow his chains while respecting the different constraints of order, board or chain integrity. He will also have the possibility to cancel his moves one by one, to delete a chain or to start the level again.

The game must be cross-platform.

Here is an example of a design, you are free to realize the one you like or to define your own selection modalities. Be careful however to have a clear display.




### 3.3 - Levels

The initial game does not have an automatic generation, you will not set it up either and you are free to use existing levels. You have to implement about 30 levels with different difficulties and chains.

You will save them in individual files to offer the possibility to add new levels easily in the future.

## 4 - Delivery

Your rendering will be a ZIP containing your source codes, a user documentation and a technical documentation explaining your choices and/or implementation choices/details.

You will show your work and give a live demonstration during the oral sessions. Your PowerPoint presentation will illustrate the technical aspects.

## 5 - Grading scale

- Display and graphic rendering: 20 points
- User input management: 15 points
- Management of chains and possible actions: 25 points
- Rollback and erasement: 20 points
- Management of level files: 10 points
- Victory conditions and level advancement: 10 points

*This scale takes into consideration the quality of your code, your documentation and your ability to produce a stable and resilient game.*

Ajouter un travail