Comprehensive Exercise Report

Team LAZREG-POILVE-LASNIER

240AEB015 LASNIER Aurélien

240AEB032 LAZREG Mehdi

240AEB011 POILVE Alexandre

[**Requirements/Analysis 2**](#_heading=)

[Journal 2](#_heading=)

[Software Requirements 3](#_heading=)

[**Black-Box Testing 4**](#_heading=)

[Journal 4](#_heading=)

[Black-box Test Cases 5](#_heading=)

[**Design 6**](#_heading=)

[Journal 6](#_heading=)

[Software Design 7](#_heading=)

[**Implementation 8**](#_heading=)

[Journal 8](#_heading=)

[Implementation Details 9](#_heading=)

[**Testing 10**](#_heading=)

[Journal 10](#_heading=)

[Testing Details 11](#_heading=)

[**Presentation 12**](#_heading=)

[Preparation 12](#_heading=)

# Requirements/Analysis

Week 2

## Journal

The following prompts are meant to aid your thought process as you complete the requirements/analysis portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* After reading the client’s brief (possibly incomplete description), write one sentence that describes the project (expected software) and list the already known requirements.
  + <<Insert one sentence description>>
  + **The software will be a 2 player’s turn-based strategy video game where the first player who aligns 4 discs of his color in the grid wins.**
    - <<Insert known requirements from client description, add more bullets as needed>>
    - **2 player’s game.**
    - **Create a grid.**
    - **Create 2 types of discs (red and yellow)**
    - **Each player gets 21 discs.**
    - **Player wins by aligning 4 discs of his color.**
    - **Selection of starting player**
* After reading the client’s brief (possibly incomplete description), what questions do you have for the client? Are there any pieces that are unclear? After you have a list of questions, raise your hand and ask the client (your instructor) the questions; make sure to document his/her answers.
  + <<Insert your questions and your instructor’s answers>>
  + **What size for the grid?**
* Does the project cover topics you are unfamiliar with? If so, look up the topics and list your references.
  + <<Insert answer>>**No**
* Describe the users of this software (e.g., small child, high school teacher who is taking attendance).
  + <<Insert answers>>**For kids ages 6 and up, so it must be easy to use.**
* Describe how each user would interact with the software.
  + <<Insert answer>>**At each turn the user will have the option to choose a column to place his disc.**
* What features must the software have? What should the users be able to do?
  + <<Insert answer>>
  + **the software needs to detect the lowest empty location on the selected column.**
  + **It also needs to detect when 4 discs of the same color are aligned to stop the game and show the winner.**
  + **It needs to ask both players if they want to start again**
  + **User should be able to quit if they want**
* Other notes:
  + <<Insert notes>>

## Software Requirements

<<Use your notes from above to complete this section of the formal documentation by writing a detailed description of the project, including a paragraph overview of the project followed by a list of requirements (see lecture for format of requirements). You may also choose to include user stories.>>

The main goal of our project is to develop a software for a client. Here, it will be a Connect 4 game. The concept is simple, it’s a 2 player’s turn based strategy game, where the goal is to connect 4 discs of the same color. To do that, each player throws their disc from the top of the same grid and then the disc goes down to the lowest position possible. This game is playable from 6 years old to 99. Since it’s originally a physical game, the objective here is to transform it into a digital game. Also, during the realization of the project, the client might add some options to it so it needs to be flexible.

To achieve this project, a list of requirement is necessary, first, the functional ones :

* First, it needs to be playable by the client on his own computer.
* Since it’s a 2 player game, the users need to be able to play their turn individually.
* Players should be able to choose who starts or be able to choose a random mode.
* The program must display a grid so the players know where they can play.
* 2 colors of discs must be created and must appear on the grid after being played.
* The software needs to know where the lowest position on each column is and also needs to know when 4 discs are aligned.
* The software should add an option to quit the game or restart a new one.

For the non-functional ones :

* It need to be accessible for young kids since it can be played at 6 years old
* The software must run fast, without lags or long loading time.
* It should be safe to play, meaning that the computer shouldn’t be damaged by the program.
* The game needs to be clear and well constructed.
* It also needs to respect the rules of the game described above.

# Black-Box Testing

Instructions: Week 4

## Journal

***Remember:*** Black box tests should only be based on your requirements and should work independent of design.

The following prompts are meant to aid your thought process as you complete the black box testing portion of this exercise. Please review your list of requirements and respond to each of the prompts below. Feel free to add additional notes.

* What does input for the software look like (e.g., what type of data, how many pieces of data)?
  + <<Insert answer>>
* What does output for the software look like (e.g., what type of data, how many pieces of data)?
  + <<Insert answer>>
* What equivalence classes can the input be broken into?
  + <<Insert answer>>
* What boundary values exist for the input?
  + <<Insert answer>>
* Are there other cases that must be tested to test all requirements?
  + <<Insert answer>>
* Other notes:
  + <<Insert notes>>

## Black-box Test Cases

Use your notes from above to complete the black-box test plan section of the formal documentation by writing black box test cases (other than actual results since no program currently exists). Remember to test each equivalence class, boundary value, and requirement.

| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Design

Instructions: Week 6

## Journal

***Remember:*** You still will not be writing code at this point in the process.

The following prompts are meant to aid your thought process as you complete the design portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* List the nouns from your requirements/analysis documentation.
  + <<Insert answer>>
* Which nouns potentially may represent a class in your design?
  + <<Insert answer>>
* Which nouns potentially may represent attributes/fields in your design? Also list the class each attribute/field would be a part of.
  + <<Insert answer>>
* Now that you have a list of possible classes, consider different design options (***lists of classes and attributes***) along with the pros and cons of each. We often do not come up with the best design on our first attempt. Also consider whether any needed classes are missing. These two design options should not be GUI vs. non-GUI; instead you need to include the classes and attributes for each design. Reminder: Each design must include at least two classes that define object types.
  + <<List at least two design options with pros and cons of each>>
* Which design do you plan to use? Explain why you have chosen this design.
* List the verbs from your requirements/analysis documentation.
  + <<Insert answer>>
* Which verbs potentially may represent a method in your design? Also list the class each method would be part of.
  + <<Insert answer>>
* Other notes:
  + <<Insert notes>>

## Software Design

<<Use your notes from above to complete this section of the formal documentation by planning the classes, methods, and fields that will used in the software. Your design should include UML class diagrams along with method headers. ***Prior to starting the formal documentation, you should show your answers to the above prompts to your instructor.****>>*

# Implementation

Instructions: Week 8

## Journal

The following prompts are meant to aid your thought process as you complete the implementation portion of this exercise. Please respond to each of the prompt below and feel free to add additional notes.

* What programming concepts from the course will you need to implement your design? Briefly explain how each will be used during implementation.
  + <<Insert answer>>
* Other notes:
  + <<Insert notes>>

## Implementation Details

<<Use your notes from above to write code and complete this section of the formal documentation with a README for the user that explains how he/she will interact with the system.>>

# Testing

Instructions: Week 10

## Journal

The following prompts are meant to aid your thought process as you complete the testing portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

* Have you changed any requirements since you completed the black box test plan? If so, list changes below and update your black-box test plan appropriately.
  + <<Insert answer>>
* List the classes of your implementation. For each class, list equivalence classes, boundary values, and paths through code that you should test.
  + <<Insert class>>
    - <<Insert needed tests>>
  + <<Insert class and tests for each class>>
* Other notes:
  + <<Insert notes>>

## 

## 

## Testing Details

<<Use your notes from above to write your test programs and complete this section of the formal documentation by creating a list of your test programs along with descriptions of what they are testing. You will also complete the black-box test plan by running the program and filling in the Actual Results column.>>

# Presentation

Instructions:Week 12

## Preparation

The following prompts are meant to aid your thought process as you complete the presentation portion of this exercise. It is recommended that you examine the previous sections of the journal and your reflections as you work on the presentation as it is likely that you have already answered some of the following prompts elsewhere. Please respond to each of the prompts below and feel free to add additional notes.

* Give a brief description of your final project
  + <<Insert answer>>
* Describe your requirement assumptions/additions.
  + <<Insert answer>>
* Describe your design options and decision. How did you weigh the pros and cons of the different designs to make your decision?
  + <<Insert answer>>
* How did the extension affect your design?
  + <<Insert answer>>
* Describe your tests (e.g., what you tested, equivalence classes).
  + <<Insert answer>>
* What lessons did you learn from the comprehensive exercise (i.e., programming concepts, software process)?
  + <<Insert answer>>
* What functionalities are you going to demo?
  + <<Insert answer>>
* Who is going to speak about each portion of your presentation? (Recall: Each group will have ten minutes to present their work; minimum length of group presentation is seven minutes. Each student must present for at least two minutes of the presentation.)
  + <<Insert answer>>
* Other notes:
  + <<Insert notes>>

<<Use your notes from above to complete create your slides and plan your presentation and demo.>>