

Comprehensive Exercise Report

Pierre Abril 230ADB088 ; Valentin Cudey 230ADB087

NOTE: You will replace all placeholders that are given in <<>>

Requirements/Analysis	2
Journal	2
Software Requirements	3
Black-Box Testing	4
Journal	4
Black-box Test Cases	5
Design	6
Journal	6
Software Design	7
Implementation	8
Journal	8
Implementation Details	9
Testing	10
Journal	10
Testing Details	11
Presentation	12
Preparation	12
Grading Rubric	13

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.
 - The project was a connect4,
 - We must provide a graphical interface, and a 7x6 grid, also we must provide turn-based gameplay for 2 players, when one player plays, it will be up to the other to play on the same computer.
- 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.
- Does the project cover topics you are unfamiliar with? If so, look up the topics and list your references.
 - We already did a connect 4 in the past but it was for a first project so it was messy , and way too long, over hundreds of line of code. So we wanted to do one again, now that we have improved in programming.
- Describe the users of this software (e.g., small child, high school teacher who is taking attendance).
 - Power 4, is a game for any type of player aged 6 or over, but it is generally played with family or friends.
- Describe how each user would interact with the software
 - The game is turn-based when a player makes their move, the token will change color and it will be the next player's turn.
 - To put a token, the player will simply have to place their mouse on one of the columns and click.
- What features must the software have? What should the users be able to do?
 - Users can play the game on the same computer.
- Other notes:
 - <<Insert notes>>

Software Requirements

We need to make a simple graphical interface with a 7x6 grid. Above this we will see the player's token moving. When the user clicks on a column, the token is placed if possible, and the turn passes to the other player and therefore the token to be placed changes color. Once 4 tokens of the same color are aligned (vertical, horizontal, diagonal), then the player with that color wins.

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)?
 - The game is played by two players on a 6x7 grid, the players inputs are pieces that they place on the grid
- What equivalence classes can the input be broken into?
 - Valid input:
 - in a non-full column
 - Invalid input:
 - In a non-existent column
 - In a full column
 - Different state of the game:
 - 4 pieces of the same player are lined up
 - The grid is full
 - The game is in progress (no winner and the grid is not completely filled)
- What boundary values exist for the input?
 - Grid boundary:
 - The first and last column
 - The first and last line
 - Limit on the number of tokens
 - 2 color because 2 players
 - 21 tokens maximum per player
 - Game limit
 - 4 tokens lined up
 - The grid is full
- Are there other cases that must be tested to test all requirements?

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
1	Testing valid inputs with an empty column	The token should be at the bottom of the column	
2	Testing valid entries with a partially filled column	The token should be just above the highest token in that same column	
3	Test valid inputs with a full column	There will be an error message and the player will have to play in another location	
4	Test with a column that does not exist	There will be an error message saying that the column does not exist, and the player will have to play elsewhere.	
5	Test to place a token on a column	The token will be present on this same column according to the first 3 tests	
6	Victory test	If a player has 4 of these tokens lined up, he wins	
7	Equality test	If the grid is full, and there are not 4 tokens of the same color lined up, then there is a equality	

Design

Instructions: Week 6

Journal

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

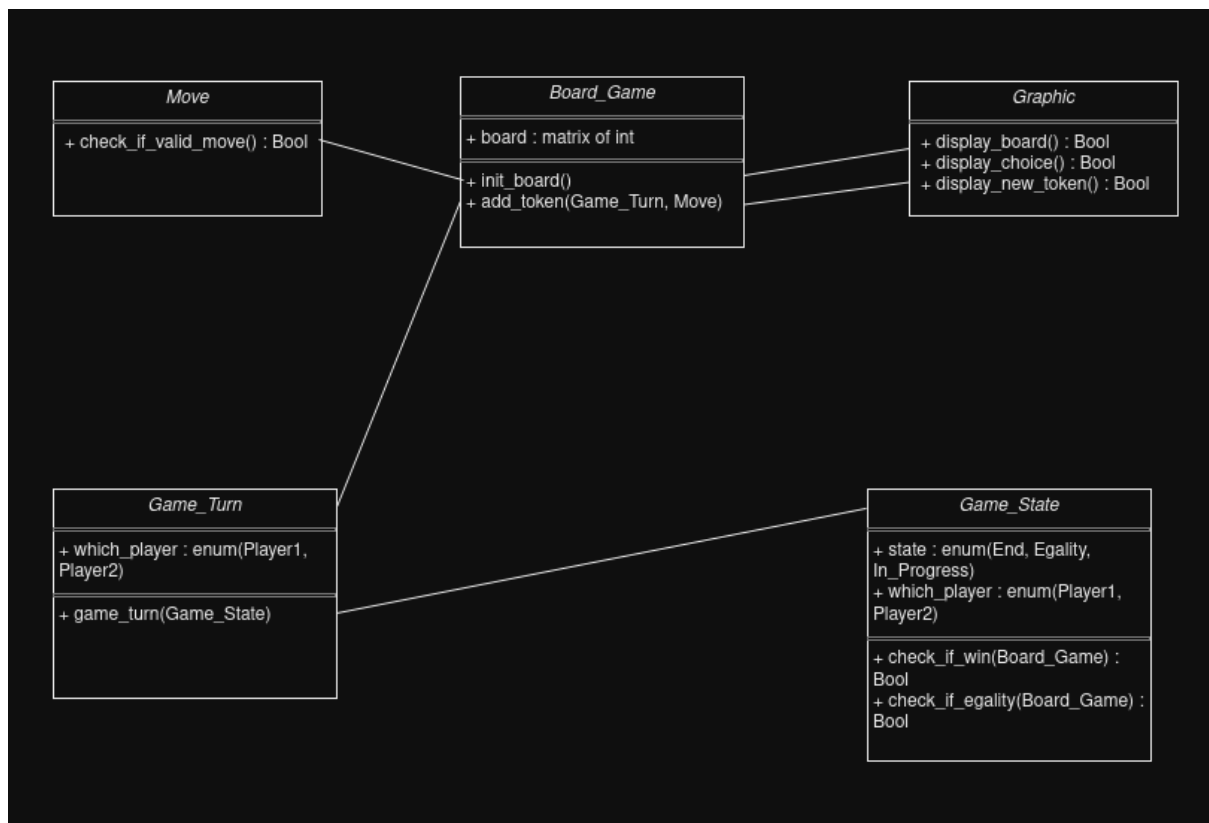
The following prompts are meant to aid your thought process as you complete the design portion of this exercise. Please respond to each of the prompts below and feel free to add additional notes.

- List the nouns from your requirements/analysis documentation.
 - Board Game
 - Token
 - Player
 - Cols
 - Rows
 - Game State
 - Game turn
 - Move
 - Graphic
- Which nouns potentially may represent a class in your design?
 - Board Game : it represents in a matrix, all the tokens placed by each player in the grid.
 - Game Turn : Represent which player it is to play, and make sure that his move is played with his token, and make the change of turn.
 - Game State : Check if there is a winner (4 tokens aligned with their color), loser or if there is a full grid.
 - Move : Check if a movement is valid.
 - Graphic : Graphical interface for the connect 4.
- Which nouns potentially may represent attributes/fields in your design? Also list the class each attribute/field would be a part of.
 - Board Game :
 - init_board
 - add_token
 - Game Turn :
 - new_turn
 - Game State :
 - check_if_win
 - check_if_equality
 - Move :
 - check_if_valid_move
 - Graphic :
 - display_board
 - display_choice
 - display_new_token

- 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.
 - To place
 - Check
 - Initialize
 - Choose
 - Display
- Which verbs potentially may represent a method in your design? Also list the class each method would be part of.
 - To place :
 - add_token
 - Check :
 - check_if_win
 - check_if_equality
 - check_if_valid_move
 - Initialize :
 - init_board
 - Display:
 - display_board
 - display_choice
 - display_new_token
- 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 be 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.
 - variable
 - function
 - loop
 - condition
 - classe
 - method
 - matrix
 - input / output
- Other notes:
 - We decided to implement our project in python, because it has very simple graphical interface management, for example with pygame which we will use.

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.
 - At the start of the project we forgot to look at the equalities and that's what we did next.
- List the classes of your implementation. For each class, list equivalence classes, boundary values, and paths through code that you should test.
 - Board Game :
 - empty board
 - inserting pawns into an empty column
 - inserting pawns into a half-filled column
 - insertion of pawn in a full column
 - insertion of pawn, while the game is over (win, equality)
 - Game Turn :
 - change of turn
 - know who's playing
 - Game State :
 - victory test
 - equality test
 - game state
 - find out who won
 - Move :
 - know if a move is possible
 - Graphic :
 - empty board
 - board with pawns
 - place a pawn
 - check that the token changes color when the turn changes
- 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.>>