Here’s a list of what has changed from the original design. The new class diagram will be shown on the next page.

* Added the string positions to the private fields in Board.h to make the constructor work.
* Added the method setBoard() in Board.h to set the board according to the player’s choices.
* Updated the EMPTY character in Position.h from ‘ ‘ to ‘-‘ to help visualize the empty spaces on the board.
* Added the integer pieceCount in Player.h to keep track of the number of pieces each player has.
* Added a method makeMove() in Game.h to make the player move and verify if it’s a correct move.
* Added a Position piece to the Player class so that every player has a piece char.
* Moved takeTurn() from Board.h to Game.h since it worked better with the attributes available in Game.h compared to Board.h. Also modified the function (it takes nothing as a parameter instead of Player current, it was easier to implement it this way).
* Added a verifyWinner() method to Game.h to check who collected the most pieces (who won).
* Added a method flipPieces() to Game.h to flip the pieces when a move is deemed correct.
* Changed what the method load() returns, it now returns a Game instead of a Board so I can create a new Game object in the main() when a saved game is loaded.
* In my save() method in Game.h, I also save the number of tokens each player has so that the load() function can keep track of how many pieces each player has.
* Changed the load() method from static to non-static so that the load() function in the main() works properly.
* Added the isSavedGame parameter in the play() function so the program can identify if a game is a New Game or a Saved Game (it is important to know because it changes how the play() method works. If it is a Saved Game, I do not need to initialize a new board.).

NOTE : Save files are stored in the cmake-build-debug folder.

NOTE : The game doesn’t check the winner automatically, the loser has to manually concede.

