

## Unit test project: Drop chip game

### *Part 1: Analyzing the Connect4 Class in main.py*

Student's name: **Darson Hak**

Date: **October 24, 2025**

**Learning outcome:** The objective of this project is to thoroughly analyze the Connect4 class defined in the main.py file. This process will involve examining the functionality of each method within the class, running them independently, and understanding how each behaves in isolation. The analysis will include modifying class values, observing the impact of these changes, and documenting findings.

#### **Steps to Complete the Analysis:**

1. **Access the main.py file:** Open and review the contents of the file to familiarize yourself with the class structure.
2. **Download the file unit\_test\_project\_part1.docx:** This document will be used to record your observations, changes made to the class, and results obtained from running the class.
3. **Analyze each class:**
  - Run the class independently.
  - For each method, understand how it works by testing its functionality.
  - Make modifications to some values in the class and observe how these changes affect the behavior of the class and the game logic.
4. **Document the analysis:** In the Word document (unit\_test\_project\_part1.docx):
  - Record how each method behaves when run.
  - Note the modifications made (e.g., changing player symbols, adjusting the board size, etc.).
  - Describe the outcome of these changes and any other observations regarding the class's behavior.
5. **Conclude the analysis:** Summarize the key takeaways from the modifications and their effects on the game's mechanics.

## Python class analysis:

### 1. `__init__(self)`

Description: The `__init__` method sets up the foundation of the game by allowing the board to be set up and played on, and declaring the starting player letter.

Changes and observations: When I changed “`self.current_player`” to “O”, I expected the game to bug and both players are “O”, but it only changes the starting player to “O” and the second player to “X”, reversing the order of the letters.

### 2. `switch_player(self)`

Description: The `switch_player` method simply switches the letter to Player 2’s letter so they may take their turn.

Changes and observations: When I change the value after “else” from “X” to “O”, Player 2 always gets to go.

### 3. `print_board(self)`

Description: The `print_board` method creates the foundation of the board and updates it after each player drops their chip.

Changes and observations: When I change the “end=” value to anything else, it changes the character (symbol) next to that player’s chip whenever the chip is dropped.

### 4. `drop_chip(self, column)`

Description: The `drop_chip` method allows the players to drop their respective chips whenever it’s their turn.

Changes and observations: If I change the rows values from  $-1$  to  $-2$ , a new row is created at the bottom, but all the chips dropped are dropped above that.

## 5. `check_win(self, player)`

Description: The `check_win` method constantly checks each row and column after each turn, checking if there are four in a row.

Changes and observations: If I change all the values that are 3 to 4, then the win conditions changes from four in a row to five in a row.

## 6. `play_game(self)`

Description: The `play_game` method is where the inputs are, allowing the player to select which column they want to drop their chip in. The method also stops the game when a player wins or the board is full, and it lets the player know when a move is invalid.

Changes and observations: If I remove all the “continue” values, the player that gets to go changes whenever an invalid move is made. Originally, it keeps the player’s turn.

## Conclude the analysis:

Each method plays a crucial role in ensuring that the Connect 4 game runs properly. However, if you change a couple of things, the game won’t break but more so modify the game which allows for fun variations of the game to be created. For instance, the game can be turned into Connect 5 or the board can be increased in size so that the game becomes more complex or longer.