

# The game Connect Four



The game Connect Four is played as follows. There are two players using the board shown in figure 1. Each player chooses a color (yellow or red for instance). Yellow starts. Players take turns by choosing a column and dropping a disc of their own color into that column. The disc takes the lowest empty space in that column. The first player who obtains a contiguous sequence of 4 of the same color, either horizontally, vertically or diagonally, wins. A more detailed explanation can be found in Wikipedia (Connect Four).

The standard board has 7 columns and 6 rows.

Figure 1: Connect Four board

## Assignment

The assignment consists of two parts, called A and B. You will get feedback on both parts, but for A the feedback is only formative (no impact on the final grade for the course). Part B is graded as Sufficient/Insufficient, and must be sufficient in order to pass the course (it is a necessary condition, not sufficient, because there is also an exam).

The assignment can be done alone or **in a team of at most three students**. The grade for the game holds for all team members. Teams are formed in class. Each team is assigned a number for identification.

Assignment A consists of drawing a usable board with Turtle graphics (or any other graphics library), such that one might play the game, using a computer instead of a physical board. It doesn't need to be playable, it's just the image of the board.

Assignment B consists of a playable game for 2 human players. The user interface can be console-based (entering column numbers), but the board must be the graphical board made in Assignment A, and the moves and current game status must be made visible in this board.

## Result and Assessment

For assignment A, the result is a piece of code that can run, error-free, in the Python 3.7 interpreter, using Turtle graphics (or any other graphics library). It is okay if it produces a reasonably tidy board, clearly usable to play the game.

For assignment B, the result is a piece of code that also can run error-free in Python 3.7, uses the board of assignment A to display game positions when playing, and that offers a console interface (text entry) to play the game.

Assignment B will be graded as **Sufficient** if:

1. two players can use it to play the game,
2. incorrect input is detected and properly handled without stopping the game (A move into a full column is also considered incorrect input and must be detected and properly handled.)
3. a winning position is detected immediately after the winning move.

## Extensions

The following features are optional extensions, for those who consider this assignment not challenging enough:

- Add an undo mechanism: at every moment in the game, you can undo moves till the beginning of the game.
- Make the board size variable. This also requires a board configuration dialog with the players.
- Make the winning sequence length (now 4) variable.
- Make it playable for more than 2 players.
- Make it playable by computer, that means, one of the players is the computer, which does look-ahead several moves deep.

## Handing in the assignment

The assignments should be handed in in Cum Laude (formerly N@tschool).

## Deadlines for handing in

Assignment A: No deadline. Strictly speaking, there is no obligation to hand this in separately. Handing in means that you will get earlier feedback on this part of the total assignment.

Assignment B: There are two attempts. **The first deadline is Friday November 9, 2018, 23:59**, i.e. the end of week 9 of period 1. Handing in on-time for this first attempt means that you will get feedback and your grade within three weeks after the deadline. **The deadline for the second attempt is Friday 25 January, 2019, 23:59**, i.e. the end of week 8 of period 2. It will again be graded within three weeks after this deadline. If you miss deadline 1 but hand in much earlier than deadline 2, you cannot count on earlier grading. You will have to wait till 3 weeks after deadline 2 at the worst.