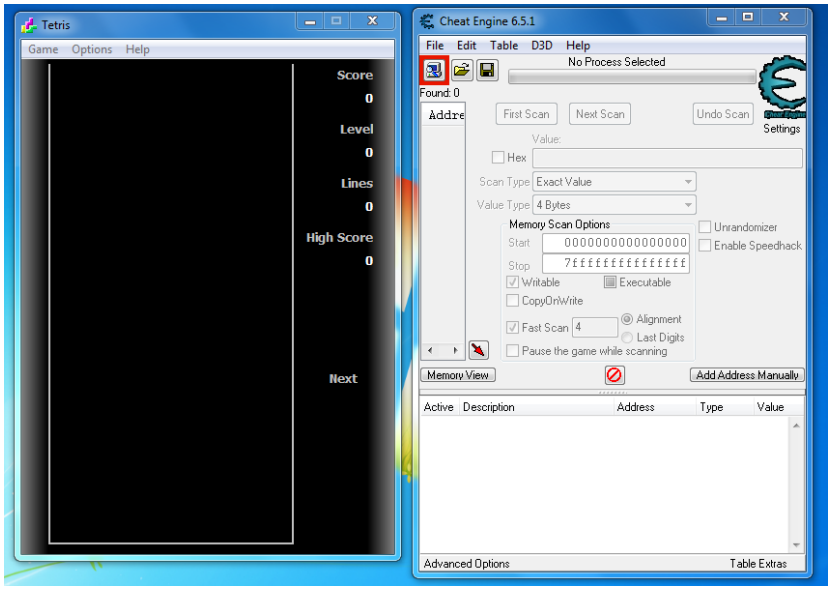
cs3235:10: - Laboratory #10 for November 6th, 2019

This laboratory could be done individually, or in a group of (at most) 2.

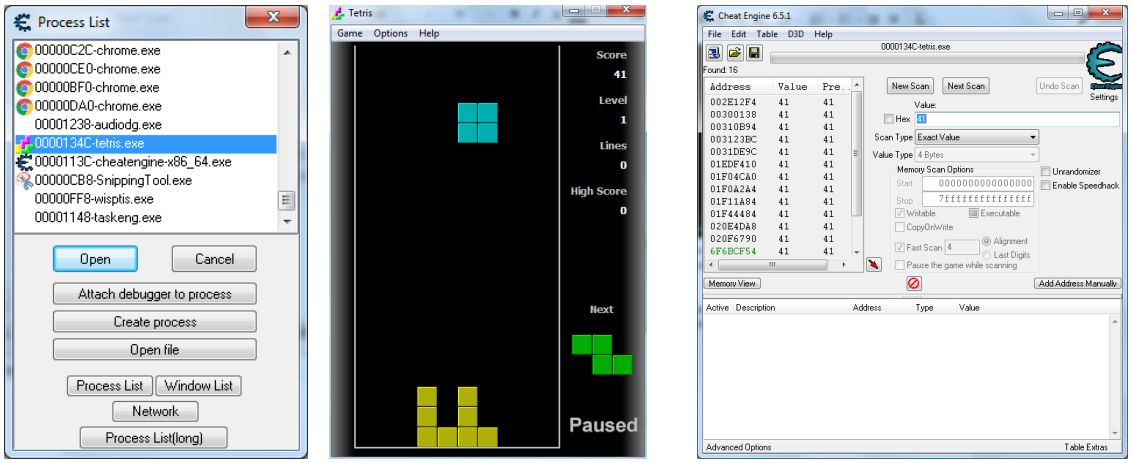
1 Lab 10, part 1: Getting started: Modifying running programs with CheatEngine

In this lab, you should begin by booting up from the Win7 partition, and logging in. Go to All programs->Cheat Engine 6.5.1->Cheat Engine tutorial, and run the tutorial for using the application. Follow the on-screen instructions to complete the tutorial. This is to get yourself familiar with the tool.

CheatEngine's primary users are probably game players trying to get better scores, but it is a generally useful memory inspection tool for studying a running program, not just games. Having said that - we will use it to cheat at a game!

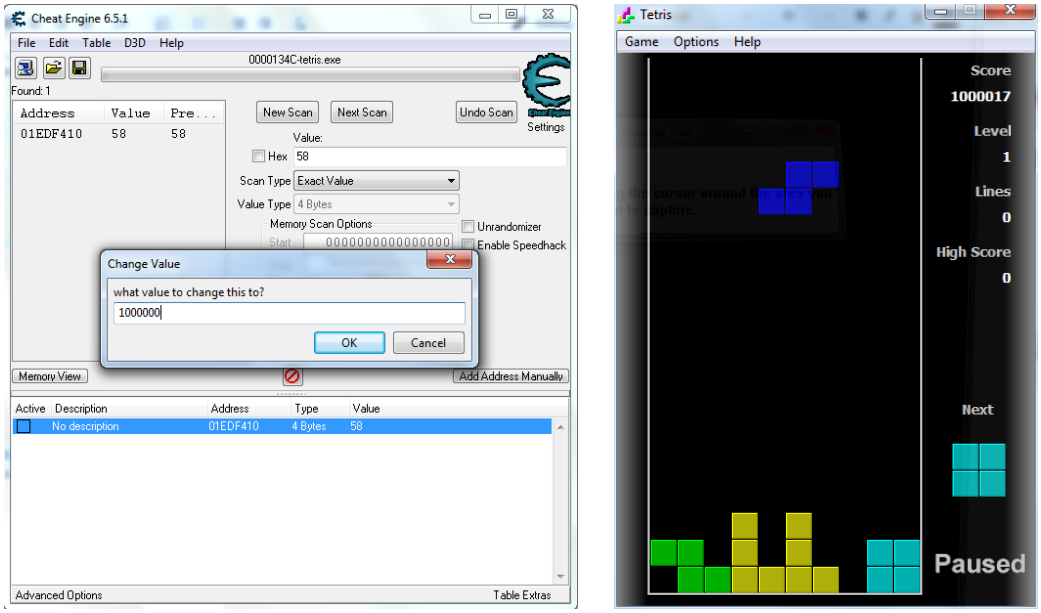


Find the tetris game, and CheatEngine, and run them both side-by-side. We will need to attach Cheat-Engine to the tetris program. We can do so by clicking on  (the first icon on the top left corner). Select "tetris.exe" from the list and click open:



In order to change the score, we will need to find the memory address which stores the score variable. Play the game for a while and then pause it. As you can see from the screenshot above, a score of 41 points is currently being displayed. We will now go to our CheatEngine and locate the memory address that has the value 41. You will probably see a different score, use the score that is currently being displayed on your screen. Type "41" into the value textbox, as seen at the top right.

Click on "first scan", a few memory addresses start to appear on the list on the left of CheatEngine. However, only one of them is what we are looking for. Go back to tetris and continue playing the game for 10 seconds, and pause the game. Now type the new score into the value textbox of the cheat engine. Instead of clicking on the "New Scan", we will click on the "Next Scan". This will force the cheat engine to scan for the value using the memory addresses in the list on the left. You should now be able to find the correct memory address for the SCORE variable and change the value to (say) 1000000:



Now go back to the game and resume it. You will soon see your score has been increased to 1000000. Access the cs3235 grading website, and enter in the likely maximum score displayable in tetris....

`https://hugh.comp.nus.edu.sg/cs3235/lab10/gradeslab10-1.php`

2 Lab 10, part 2: SEED Buffer-Overflow Vulnerability Lab

Have a look at the SEED Buffer-Overflow Vulnerability lab, found at

`http://www.cis.syr.edu/~wedu/seed/Labs_16.04/Software/Buffer_Overflow/`

There is a lot in this laboratory, but I would like you to just try task 1, a simple buffer overflow attack. Once you have finished the lab, you can upload your exploit.c code to the Luminus upload folder for Lab10-2. Please name your file e00XXXXXXX.c (i.e. your login id followed by .c).

Access the cs3235 grading website, and enter in a brief description of the steps someone could follow to recreate your attack, using your code.

`https://hugh.comp.nus.edu.sg/cs3235/lab10/gradeslab10-2.php`