Lab 2 – Input

1. Fork the lab project from GitHub https://github.com/Abertay-University-SDI/CMP105_W2, use the Lab0 Git Guide from last week if you need a reminder on how to do this. This project builds on last week's work and adds the Input class to our framework as discussed in the lecture. Make sure the application compiles and runs. Give the input class a look over, but **do not** edit it. The input class is in the "Framework" folder in Visual Studio, this is done to make the project easy to manage and separate the code, with your code in "header/source files" folders.
2. In the ***handleInput()*** function in Level.cpp write code to test the input class. Have the program output text to the console when:
   a. The 'W' key is pressed.
   b. When the three keys 'J', 'K' and 'L' are pressed together (but not individually).
3. Add code to the level class so that the application exits/closes when the Escape key is pressed.
4. The input class stores the mouse's current position and this is updated when the mouse moves. Add code to level class that renders text containing the mouse's current position, in the top left corner of the window. This should update as the mouse moves around the window.
   Example of building a string variable (http://www.cplusplus.com/reference/string/string/string/)
   ```
   std::string output ("this output string")
   ```
5. The input class has functions for storing the left mouse click event. Write an application that measures the distance of a mouse drag (while holding the left mouse button down). Output the distance of the drag to the console on completion of the drag. You can either calculate the distance the mouse moved by separately calculating the difference in X and Y positions, or use Pythagoras to calculate the distance. It is recommended you work out the problem on pen and paper first, consider what variables will be needed and what events/conditions the application will need to capture/track.
6. Add to the application so that on mouse right-click the application renders a circle at the mouse's location. On subsequent clicks the circle should be re-rendered at the mouse's new/current position.
7. Complete the Week 2 quiz on My Learning Space