# COMP 2601 Winter 2021

# Assignment Proposal: Stock Viewer App

**Assignment Grading:**

**Background**

The purpose of this assignment is to build an IOS app that access a JSON RESTful-API server and parse the JSON data to display on the app. The assignment is based on class Tutorials 9A (touch events), 9B (multi-touch and gestures), 10A (Web Access Part A) , 10B (Web Access Part ) .

**Assignment Requirements**

In this assignment we want you to build an IOS-based stock-viewer app that a user can use to search for a stock and view the profile of it. The stock-viewer app must have a user interface that is built from using GUI objects and interacted by using simple gestures like tap, double tap etc. How you build the user interface is up to you whether that be doing it programmatically or SwiftUI.

# Design Requirements

**R1.1** The game should be built with Xcode 12.x, and Swift 5 or later. You can restrict the game to use gestures supported by the simulator if you want. The game should provide a two-player tic tac toe game played by two people on the same device taking turns to draw on the screen (the application does not play as one of the opponents).

**R1.2** When the game launches the screen should be completely empty. A user should then be able to draw the game board using a two-finger vertical drag and a two finger horizontal drag. Once the game board has been drawn the game should be ready to receive player inputs. (On the simulator it can take a while for the "blank" screen to load. You can put something on the screen so the user knows the app is loaded and ready.) Also on the simulator you can do a two-finger drag by holding down the option key when you drag your mouse.

**R1.3** When the game board has been constructed the game should acknowledge this by printing a caricature of the empty game board on the console output. (or at the bottom of the game screen if you prefer). Use a caricature made of mono-space font characters like courier.

**R1.4** After each player plays a new caricature should be printed which represents the state of the board. For example (again this can be on the console or a corner of the game screen if you prefer):

**R1.5** The first player should be able to start with either an X or an O. After that the players play in turn. The game should determine whether the correct symbol (X, or O) has been drawn by the user with their finger.

**R1.6** The player should enter their symbol by drawing it with a finger, or mouse:

**R1.7** The game should retain the shape they draw for their X or O symbol and use that shape in the game board display. That is, don't replace it with a font character.

**R1.8** The game should determine whether they drew an X, an O, or some other (non-legal) symbol. The game should use the touch points and determine whether what was drawn was sufficiently like an X or an O. Also when a player plays the game board caricature should update.

**R1.9** If the player plays out of turn or draws a shape that is not sufficiently like an X or an O the shape should be erased and the current player's turn should still be in effect (allowing them to draw another shape).

**R1.10** If the game ends with someone winning the "winning line" should be drawn by the game (the application should recognize this situation):