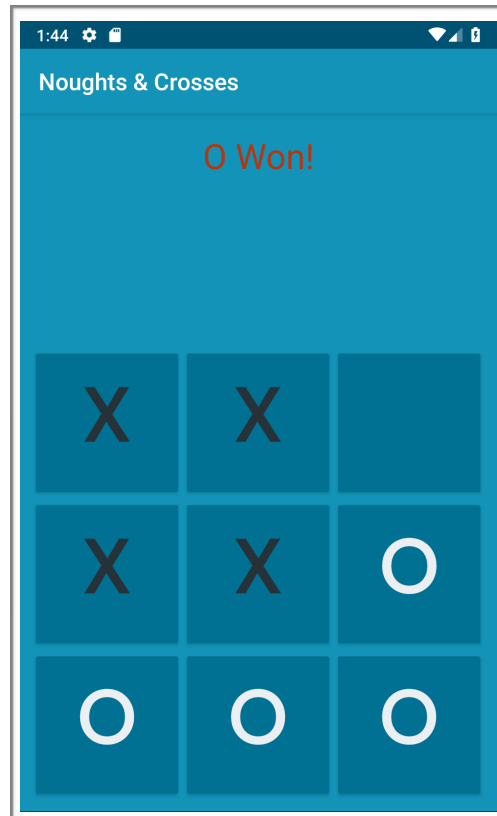


Noughts and Crosses

Android Development Skills - Reflective Report



Paul Johannes Aru

Autumn 2018

[\[github.coventry.ac.uk/A202SGI-1819T1A/7155312-Paul_J._Aru\]](https://github.coventry.ac.uk/A202SGI-1819T1A/7155312-Paul_J._Aru)

Android is the worlds most popular mobile Operating System (StatCounter 2018). In fact, for a brief moment last year, its market share passed that of the Windows', making it the worlds most dominant OS compared to any platform. In addition, as Java is the longtime second most popular programming language on Github (GitHub 2018), learning Android Development Skills did not just seem fascinating, but also necessary.

With no previous experience in Android Studio or Java, this project was going to be without a doubt, challenging. As such, the application that I would create for Android had to be simple enough to be accomplishable in the given time frame. However, at the same time, the idea would have to flexible enough that features could be added if the base application was finished early. After a long consideration, a Noughts and Crosses (Tic-Tac-Toe) game seemed like the ideal choice. It would provide more flexibility in the development phase than a Sudoku game and be more interesting to create than a generic notes or contacts app.

Noughts and Crosses has a long history in video game development. While many people consider Pong from 1972 as the first video game, then Bertie the Brain, the first computerised Tic-Tac-Toe, was exhibited all the way back in 1950 (Blitz 2016). Although it used lightbulbs to indicate X's and O's on a nine-square grid, then just two years later a University of Cambridge PhD student created OXO. This Noughts and Crosses game ran on the universities mainframe and displayed the grid on a CRT, making it the worlds first graphic computer game (Cohen 2018). Therefore, not only is Noughts and Crosses practical for students looking to quickly entertain themselves between lectures, but it is also a symbolic game in the Computing field.

An Android application construction starts at the activity design, where the layout of the app is created. Even though developers have a free hand in designing the look of the program to their heart's content, there are numerous guides available for smartphone and Android app design. Google has published a guide to Material Design, a design language that is not just used for Android, but throughout their products. Inspired by the interaction between ink and paper, Material Design guides developers towards a more minimal User Interface that matches the look and feel of the rest of the Operating System (Thornsby 2016).

After looking through many of the existing “Tic-Tac-Toe” and “Noughts and Crosses” games on the Google Play store, I noticed that most of them could be placed into either two categories. Some games looked like they were not ready for downloading yet with the default grey buttons and a white background. Others were heavily over designed with colourful images to the point that just made them aesthetically unpleasing. I chose to differ by creating a simple and visually pleasing game with the Material Design language in mind. I set the Noughts and Crosses game that is built-into Google search, as seen in Figure 1, as a benchmark. That is not to say that I set out to create an identical clone, but much rather, my own Noughts and Crosses game that was inspired and followed the spirit of Google’s version. As Figure 2 illustrates, I created a custom colour scheme with the help of the Material Palette Generator, available online.

The design of an easy-to-use application does not just end at visual design, but also extends to activity construction. The fewer components an app’s layout has, the quicker the end-user will be able to comprehend its functionality. Therefore, as seen in Figure 3, aside from a *TableLayout* that contains the 3x3 grid of buttons for the game, there is a single *TextView* used for various messages.

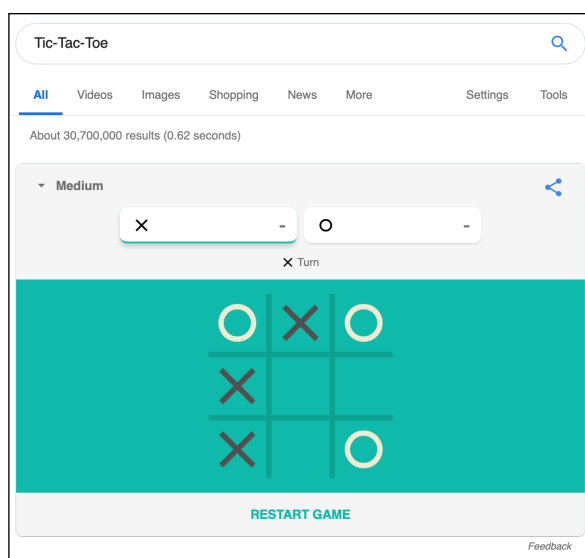


Figure 1 - A Noughts and Crosses game that is built into Google search.

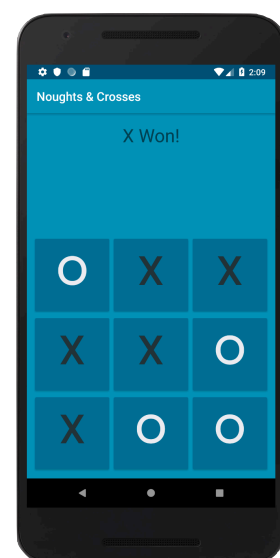


Figure 3 - The Activity Layout.

Colour Location	Primary	Primary Dark	Accent	Background	3x3 Squares	X - Symbol	O - Symbol
Colour Code	#1293b7	#005273	#b73612	#1293b7	#007093	#263238	#ECEFF1

Figure 2 - The Colour Scheme.

As mentioned earlier, I had no previous programming experience in Java. Because of that, a lot of the code had to be re-written time and time again to improve the program efficiency. For example, changing 9 individual objects that stored information about each square into an array that contained the same information made the file approximately 100 lines shorter. In addition, as I managed to finish the main game logic ahead of time, there was also time to add functions for saving & loading a game and restarting a game once it is over.

The main Java file contains seven fundamental functions. The *onButtonClick* function handles player input and the *computerGuess* function handles computer input. While not very smart, *computerGuess* based on a random number generator is simple and can beat the player if it gets lucky. *gameIsOver* is a function that checks if any of the conditions required for a victory or a tie are met. As the only function with a return, if a game is over, it returns “true” and displays the reason on *TextView*, otherwise, the function returns a “false”. When a game has ended, the *gameOverMessage* checks if the player wants to start another game with a pop-up message, as seen in Figure 4. The *gameReset* function takes care of returning everything to its original state before a new game begins. Finally, the *save* and *load* functions store and look-up what value each square has, in case the user quit the app in the middle of a game. As figure 5 illustrates, the data is stored square by square, from 1 to 9, separated with “|” symbols.

As each square of the 3 by 3 Noughts and Crosses grid needs to store and process similar information, it only seemed suitable to create a class for this. The *Square* class itself is very simple. It has three significant variables and one method. Variables *isEmpty* and *value* keep track of the objects state and *location* points to the respective button in the activity. The method *setValue* does exactly what it sounds like. It takes in a value and a colour and sets the object to that.

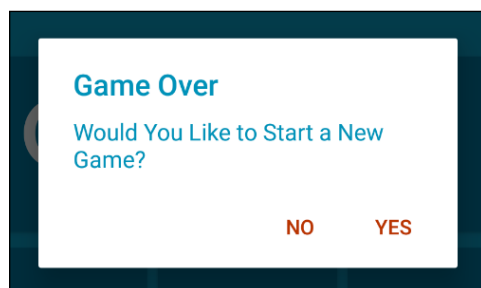


Figure 4 - Pop-up Question.

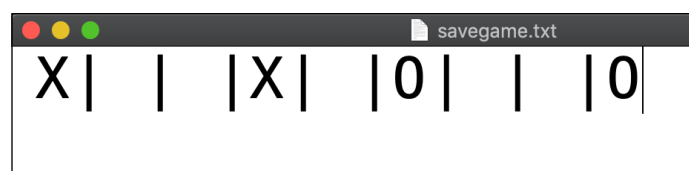


Figure 5 - Data Stored by the Game.

Unlike Apple's iOS, anyone can create an Android device. As Google and Open Handset Alliance emphasised in the press release revealing Android, it is a developer-friendly open-source OS, that provides manufacturers with a remarkable independence and flexibility in device design (Google and Open Handset Alliance 2007). While this has resulted in a large variety of devices being available for end-users, it has also created many compatibility and support issues for developers. Android app creators have to take into account different display sizes, screen densities, platform versions, etc.

In order for my app to be available for a large market, I chose API level 22 as the minimum platform version. As Figure 6 illustrates, this should mean that approximately 4 out of 5 Android devices can run the app. To verify that the game indeed runs as expected on different devices, 3 virtual devices were used for testing. As seen in Figure 7 and 8, aside from Lollipop not supporting the style that sets the *gameOverMessage* button colour, everything worked perfectly.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%

Figure 6 - Android Platform Adoption

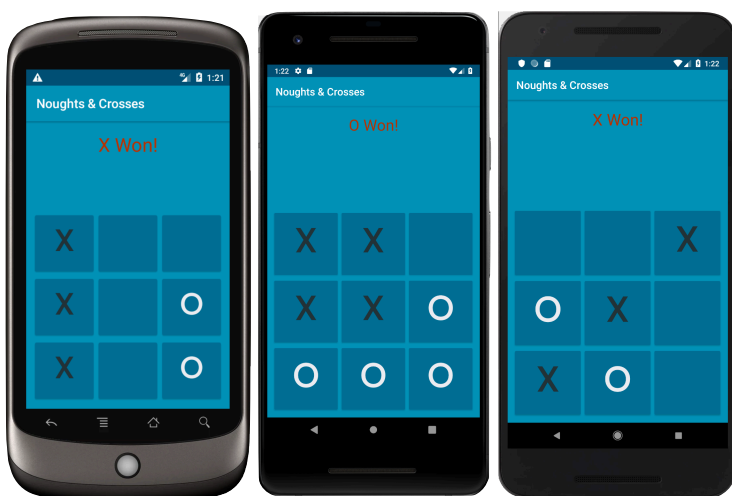


Figure 7 - Testing on Nexus One [API 22] (Left), Pixel 2 [API 28] (Middle) and Nexus 5X [API 27] (Right).

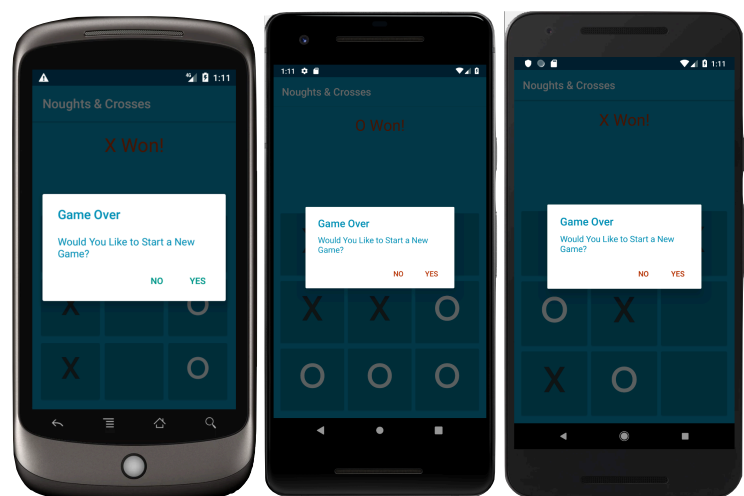


Figure 8 - Testing on Nexus One [API 22] (Left), Pixel 2 [API 28] (Middle) and Nexus 5X [API 27] (Right).

Starting with no previous knowledge, I found the material covered during the tutorial sessions very challenging. It did not help that the assignment required us to make an app that could compete with the top-tier applications on the Play store. Had I not had a strong passion for creating mobile applications, I might have given up by this point. Nevertheless, I covered a lot of the foreign concepts individually and came up with my first Android app. I am proud to say that I am very pleased with the outcome.

However, that is not to say that the app is perfect. Quite the contrary, I believe that there is always room for improvement in application development. If there had been more time to complete the assignment, there are a number of features that I would have done differently or included. Starting with small things like a visual indicator that highlights the squares that formed a winning combination, all the way to a separate multiplayer mode.

This module has not only given me a good insight into Android app development and programming in Java but also improved my understanding of classes in object-oriented programming. These skills will certainly improve my employability and can come in handy in any future projects I set out to accomplish. I look forward to creating many more Android applications.

References

- StatCounter (2018) *Operating System Market Share Worldwide* [online] available from <<http://gs.statcounter.com/os-market-share#monthly-200901-201812>> [5 December 2018]
- GitHub (2018) *Octoverse 2018 Projects* [online] available from <<https://octoverse.github.com/projects>> [5 December 2018]
- Blitz, M. (2016) 'Bertie the Brain Still Lives - The Story of the World's First Arcade Game.' *Popular Mechanics* [online] 2 November. available from <<https://www.popularmechanics.com/technology/gadgets/a23660/bertie-the-brain/>> [6 December 2018]
- Cohen, D.S. (2018) *OXO Aka Noughts and Crosses - The First Video Game* [online] available from <<https://www.lifewire.com/oxo-aka-noughts-and-crosses-729624>> [6 December 2018]
- Thornsby, J. (2016) *Android UI Design*. Birmingham: Packt Publishing
- Google and Open Handset Alliance (2007) *Industry Leaders Announce Open Platform for Mobile Devices*. 5 November. available from <<http://googlepress.blogspot.com/2007/11/industry-leaders-announce-open-platform.html>> [8 December 2018]