

# WRPV302 Assignment 1

## Due Date:

Check the Learn site. This is a **two week** assignment.

## Objectives:

- Introduction to Android

## Note:

Make use of Android Studio when creating your solutions for assignments, **but** make sure to use the same versions as in the departmental labs **and** have the Android SDK and Java SDK installed in the same locations. If you do not, when people mark your assignments, it might take a while for them to get it working... and this may affect how easily marks are awarded...

You may find the install files on <\\cs2\courses\WRAP302\installs>

## Task 1: Android “Hello World”

Complete and submit the “Build your first app” tutorial on the Android Developer site (<https://developer.android.com/training/basics/firstapp/>).

## Task 2: Android Matching Numbers Game

In the June 2019 exam for WRPV301, you were required to write a JavaFX version of a *matching tiles* game. Below is an excerpt from the exam paper:

*Many parents buy their children memory card games to encourage pattern matching and memory skills.*

*The game uses a deck of cards, where each card usually has pictures, words or numbers on them. There are pairs of matching cards in the deck, e.g. two apple cards, two orange cards, two lemon cards, etc. The deck will therefore always have an even number of cards in it.*

*The game is played by shuffling the deck of cards and dealing them face down in a grid pattern. Once the cards have been placed, the player selects two cards that are face down and flips them over. If the cards match (both have the same number or picture) then they both remain face up. If the cards don't match, then they are both turned face down again. This continues until all pairs of cards have been match and all the cards are face up. In this case, the player wins.*

*To make the game more challenging, there is sometimes a time limit placed on winning, or a fixed number of times the player may flip a pair of cards. If the player does not win in the given time or flipped more pairs of cards than allowed without winning, then the player loses.*



You are now required to create a *simple* version of the game that allows a player to find matching numbered tiles, instead of pictures. The game is played over a *fixed number of turns* (not a limited time). And finally, the board has a fixed size of 4 by 6 tiles.

You need to create an Android app, that does the following:

- Starts with a splash screen that displays the name of the game “Memory Buster”. The size of the font must be large, and the text centered vertically and horizontally on the screen. The text colour and background colour must be changed to look “pretty”. Tapping on the screen anywhere, takes the player to the main game play Activity.
- The game player Activity has a label that displays the number of remaining turns and a grid of “tiles” underneath. The game is played on a 4 x 6 grid, where each “tile” is a Button that may be clicked.
- Clicking a tile “turns it over” and displays the number under each tile. After two “tiles” have been flipped, they either match or don’t. The “tiles” are either *both* flipped face down again if they don’t match, or left face up if they do match. This takes place *after* two “tiles” are flipped and any tile is clicked on, i.e. the tiles remain as they are *until* the user clicks on something else. Flipping two tiles is considered *one* turn.
- After each turn, the game is either over (a win or loss) or it continues for another turn.
- If the game is over, a win or lose message (on a different Activity) needs to be shown. On this Activity, the player can choose to play another game. If they choose to play another game, another random placement of tiles is to be generated and play resumes as before.
- The state of the “tiles” needs to be reflected on the Buttons. Since we haven’t dealt with changing styles from code yet, use the following convention. If a “tile” is face down, the text of the button will be an empty string. When a “tile” is flipped up, the number is displayed. If the “tiles” match, then the number is displayed with square brackets around it.