

COMP7506 Smart Phone Apps Development

Assignment 1

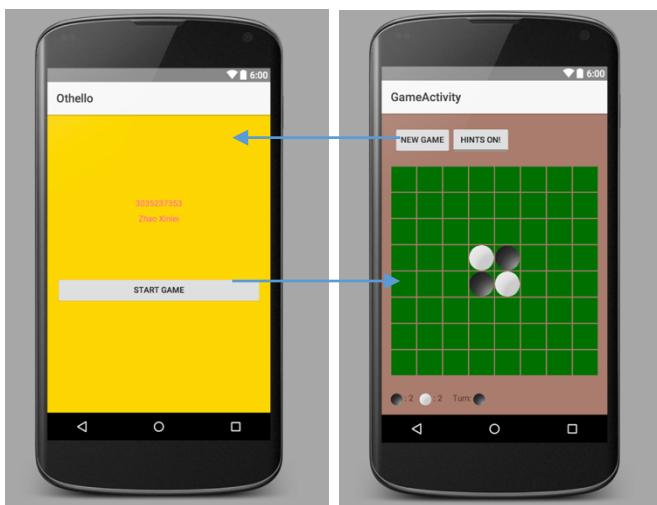
Zhao Xinlei
UID: 3035237353

1. Introduction

This project is an Android game called Othello that includes basic game, and “show hint” feature. In this game, only human vs human play mode is developed. There’s no human vs AI included. This app is based on Android API 23. On Android Studio, the code can be compiled and game can run successfully on AVD Nexus S API 23, with resolution 480x800: hdpi.

2. Game Flow and Design

This game UI includes two activities.



Main activity: This screen has name and student number, with a button “START GAME” that navigates player to the next screen. On this screen there are two textView and a button.

Game activity: This activity is a game board. When game starts, player can choose to start new game, or play with current game. This activity also shows game status and results. Chessboard is made of ImageButton filled in tableLayout, while status part is also a tableLayout with imageView and textView inside.

This game has also been tested on a higher resolution simulator (Nexus 6 API 23, 1440*2560: 560 dpi), which looks exactly the same with what we see on Nexus S, too. No strange distortion on UI is seen.

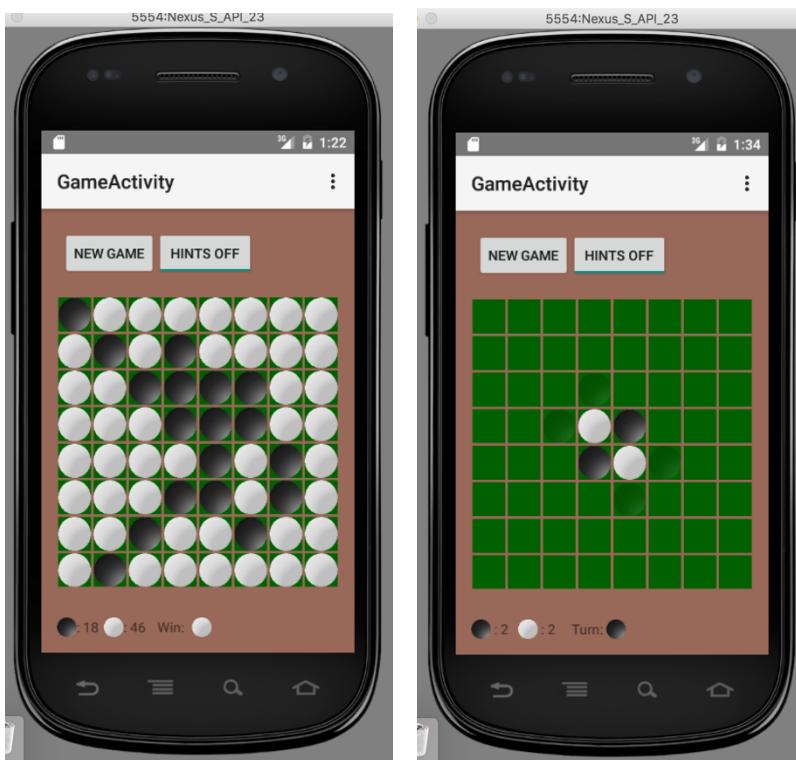
3. Implementation

Algorithm of the game Othello:

Firstly, a board is made and player can put black/white pieces alternatively without flipping pieces at any place as long as it's empty on board. Black_count and white_count addition calculation can be added to this step, too.

Secondly, before every step, the app need to check every empty space on board: whether this space is occupied [IsPlaceable(Position pos)], what direction can this piece flip the opponent [isMovePositionValid(Position pos)] and most importantly, get how many pieces can be flipped if we place this piece to that position [getNrOfSwitches(Position pos, boolean v)].

Finally, [checkState()] is added to every successful click to see whether it is possible for black/white to make moves on board. If it's white piece's turn but white don't have possible moves, it should be switched to black's turn. However, when there's no space available for both sides, game should end and results is placed at the end of status bar.



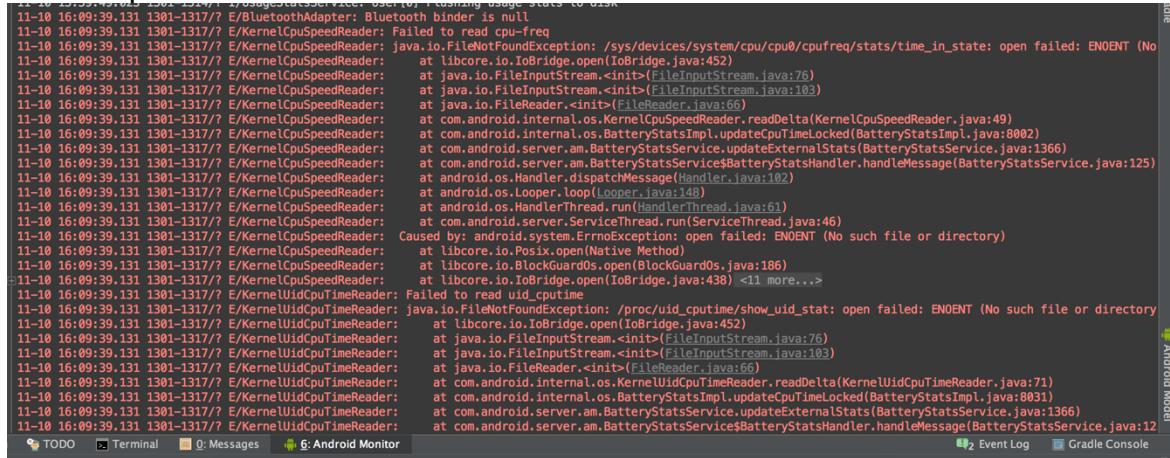
4. Features

In this game, “Hints On/Hints Off” toggle button is added for players to see which space is available for the current piece color, by adding black_trans and white_trans images to the corresponding imageButtons. When this toggle is on, it should also watch player's step, i.e. give out hint after every step is made. This is realized by adding a [Refresh()] function. Since we have board's real data stored on nboard[][][], after each time user put a piece on board, hints firstly clean up all spaces, then show hints with the current color, which will be in transparent pictures.

When this toggle is off, [Refresh()] function is used to clean up all non-existing pieces on board with nboard[][] data.

5. Limitation

- 1) The function of this app is working properly, but I have observed when this app is running on Nexus S simulator, logcat will show some error message. I assume that is caused by probably memory leak, or the simulator itself has some hardware/software conflicts since it happened while the app is standing by, but not quite sure about that.



```

11-10 16:09:39.131 1301-1317/? E/BluetoothAdapter: Bluetooth binder is null
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: Failed to read cpu-freq
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: java.io.FileNotFoundException: /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state: open failed: ENOENT (No
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at libcore.io.IoBridge.open(IoBridge.java:452)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at java.io.FileInputStream.<init>(FileInputStream.java:76)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at java.io.FileInputStream.<init>(FileInputStream.java:103)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at java.io.FileReader.<init>(FileReader.java:66)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at com.android.internal.os.KernelCpuSpeedReader.readDelta(KernelCpuSpeedReader.java:49)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at com.android.internal.os.BatteryStatsImpl.updateCpuTimeLocked(BatteryStatsImpl.java:8002)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at com.android.server.am.BatteryStatsService.updateExternalStats(BatteryStatsService.java:1366)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at com.android.server.am.BatteryStatsService$BatteryStatsHandler.handleMessage(BatteryStatsService.java:125)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at android.os.Handler.dispatchMessage(Handler.java:102)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at android.os.Looper.loop(Looper.java:148)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at android.os.HandlerThread.run(HandlerThread.java:61)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at com.android.server.ServiceThread.run(ServiceThread.java:46)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: Caused by: android.system.ErrnoException: open failed: ENOENT (No such file or directory)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at libcore.io.Posix.open(Native Method)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at libcore.io.BlockGuardOs.open(BlockGuardOs.java:186)
11-10 16:09:39.131 1301-1317/? E/KernelCpuSpeedReader: at libcore.io.IoBridge.open(IoBridge.java:438) <11 more...>
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: Failed to read uid_cputime
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: java.io.FileNotFoundException: /proc/uid_cputime/show_uid_stat: open failed: ENOENT (No such file or directory)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at libcore.io.IoBridge.open(IoBridge.java:452)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at java.io.FileInputStream.<init>(FileInputStream.java:76)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at java.io.FileInputStream.<init>(FileInputStream.java:103)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at java.io.FileReader.<init>(FileReader.java:66)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at com.android.internal.os.KernelUidCpuTimeReader.readDelta(KernelUidCpuTimeReader.java:71)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at com.android.internal.os.BatteryStatsImpl.updateCpuTimeLocked(BatteryStatsImpl.java:8031)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at com.android.server.am.BatteryStatsService.updateExternalStats(BatteryStatsService.java:1366)
11-10 16:09:39.131 1301-1317/? E/KernelUidCpuTimeReader: at com.android.server.am.BatteryStatsService$BatteryStatsHandler.handleMessage(BatteryStatsService.java:12

```

- 2) Also in the [Place(int[][] b, Position c)] function, most of its steps have been written twice, only because the place of black/white is different. There should be some easier ways to write this part, because on the example of reference (2) it has used fields. But if this board also use Field, it will probably cause a large area of code change, so function [Place(int[][] b, Position c)] looks complicated now. The code could look simpler if use player black/player white as Fields, and indicate change to the other Field while a piece is being turned.

6. Reference

Here is the list of resources I used for developing this app:

- 1) COMP7506 Course workshop materials
- 2) Game algorithm is rewritten from <http://martin-thoma.com/images/2012/04/minted-source-code.pdf>
- 3) When writing board imageButtons, this method is adopted:
<http://www.technotalkative.com/android-findviewbyid-in-a-loop/>
- 4) Stackoverflow which helped me solve a lot of UI layout issues.

7. Files included in this package

- 1) Workspace folder named Othello then compressed and renamed to 3035237353.zip.
- 2) Readme report (this file)