

Android 魔術方塊玩家 App

An android application for Rubik's Cube player

曾理 Li Tseng¹

指導教授：張欽智

Advisor: Chin-Chih Chang²

中華民國 101 年 6 月

June, 2012

¹ 中華大學資訊工程學系大學部學生。

² 中華大學資訊工程學系助理教授(聯絡地址：30012 新竹市五福路二段 707 號，聯絡電話：03-5186091，E-mail: changc@chu.edu.tw)。

摘要

近年來，由於智慧型手機的流行與開發門檻的降低，越來越多的開發者選擇在智慧型手機上實現卓越的創意，使得現代人手一支智慧型手機，這便利了我們的生活也更成為了行銷的利器。

已經有越來越多店家利用智慧型手機打卡的機制做出種種羨煞人的優惠，使用者更可以便利地利用 QR code，方便而且快速地連接網路瀏覽各種資訊。

本論文提出一套在 android 作業系統上專屬於魔術方塊玩家的 android application，運用矩陣模擬魔術方塊的轉動讓玩家在沒有攜帶魔術方塊的時候也可以方便練習魔術方塊，使用 Handler 建置的計時器可自定畫面更新頻率降低系統負擔，更方塊玩家出門在外不再需要攜帶硬體計時器，並附有亂轉公式產生器讓玩家們可以使用統一的亂轉公式不再為了亂轉公式的公平性煩惱。

關鍵字: 魔術方塊、Android、Application

Abstract

In recent years, because of the popularity of smart phones and decrease of the development threshold, more and more developers realize their innovations on the smart phones. Nowadays, almost everyone has a smart phone on hand. Smart phones are not only convenient our life but also become a marketing tool.

More and more stores have already used the "check in" function of the smart phone to provide different sorts of discounts. Smart phone users can also easily use "QR code" to search more discount information.

Our project brings up an android application which is targeted to the player of Rubik's Cube. We use the matrix to simulate the move of the mini cube. It is easy for us to practice the mini cube even if you don't bring our own mini cube physically. The load of the system can be reduced though the hour meter of Handler. Mini cube users don't need to bring a timer with them when they are outside. Our application is also attached with a scramble maker which will produce a unifiable formula. The players don't need to worry about the equality of the results of the scramble formula.

Key words: Rubik's Cube, Mini Cube, Android, Application

壹、緒論

這些年來，由於智慧型手機的流行與開發成本的降低，越來越多的開發者開始在 android 平台上一展長才，漸漸的智慧已經深入了我們的生活，不論是開發者或者使用者都開始慢慢思考，智慧型手機如何應用在我們的生活，讓身邊的一切變得更智慧。

本論文旨在建構一個整合性的程式，將平常魔術方塊玩家需要使用的功能，全部整合進手上的智慧型手機裡，例如：計時器、亂轉公式產生器、二三階魔術方塊，期望減少平時魔術方塊玩家所攜帶的硬體，以往玩家需要額外攜帶硬體計時器或者攜帶筆記型電腦充當計時器跟亂轉公式產生器更或者使用碼表計時，但這些對在外行動的魔術方塊玩家而言都是額外的負擔，而且也與官方比賽的使用器材不符合。

貳、相關技術探討

一、Andorid

Andorid 是以 Linux 為基礎的半開放原始碼的作業系統，由 google 和開放手持設備聯盟持續開發，並主要在移動設備上運作，例如：智慧型手機、平板電腦。

Google 透過官方網路上的商店平台 Google Play，讓開發者上傳自己研製的應用程式也讓使用者可以透過這個平台下載需要的程式，而這樣便形成了良性的循環，越來越多的開發者與使用者投入了 android 的擁抱。

根據 2010 年尾的數據顯示，推出兩年的 Android 佔有率上已經超越了過去十年來的霸主諾基亞 Symbian，並成為世界第一大智慧型手機作業系統。

二、Eclipse

是著名跨平台的開源碼 IDE，最初主要是用作 Java 語言的程式開發，目前也有人透過外掛的方法，使其擴充為 C++、Python 語言的發開工具。

其實 Eclipse 本身只是一個平台，因為眾多外掛的支持讓 Eclipse 具有廣大的支援性，其中也支援本論文的 android 開發研究，許多 android 開發者與大部分市面上的書籍皆選擇 Eclipse 為主要開發工具。

Android 開發者可以透過 install new software 功能，連接 <http://dl-ssl.google.com/android/eclipse/> 來取得 android 的開發工具。

參、系統架構

1. 計時器

這是一個從魔術方塊玩家為出發點的智慧型手機應用程式，以往玩

家需要攜帶計時器(如圖 1)或者攜帶筆記型電腦執行計時器程式，而本應用程式將計時程式實作在本研究程式中(如圖 2)。



圖 1 魔術方塊比賽現場

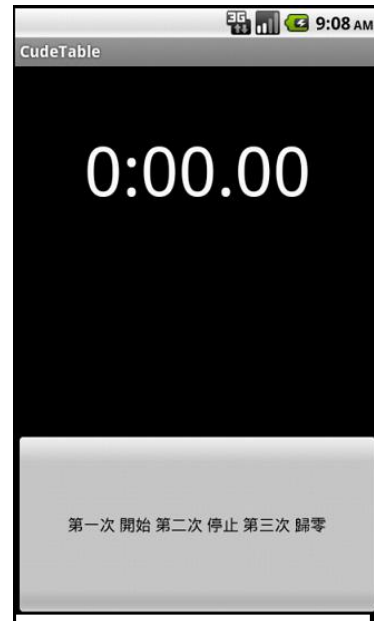


圖 2 本研究的計時器

2. 亂轉公式產生器

魔術方塊玩家一起聚會的時候，常會遇到需要一起比拚還原速度的時候，以往玩家需要攜帶筆記型電腦執行亂轉公式程式才可以有公平一致的亂轉公式，本研究將亂轉公式產生器實作在這次研究的應用程式裡(如圖 3)。

F Front

B Back

U Up

D Down

R Right

L Left

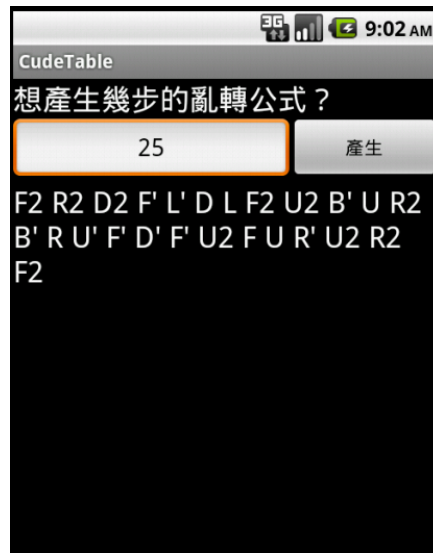


圖 3 本研究的亂轉公式產生器

3. 二三階魔術方塊

本研究也將二三階的魔術方塊實作了，考慮到魔術方塊玩家在閒暇無事的時候，也可以沉浸在魔術方塊的樂趣裡(如圖 4)。

如何使用-三階魔術方塊(如圖 5)

1. 英文按鈕表示方塊六個面的轉動，玩家可以自行轉亂方塊，或者使用步驟 2 自動轉亂。
2. 依據官方規則隨機產生 25 步的亂轉公式。
3. 重新開始(reset)
4. 開始計時，直到方塊被還原。
5. 計時器
6. 以平面展開的方式，將方塊六個面展開

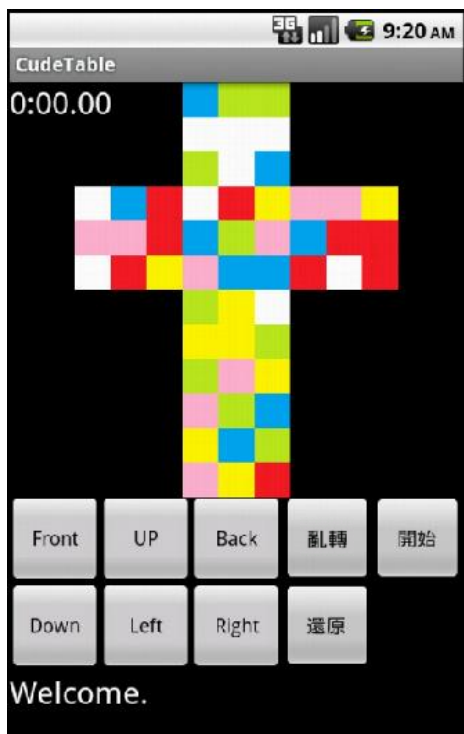


圖 4 本研究的三階魔術方塊

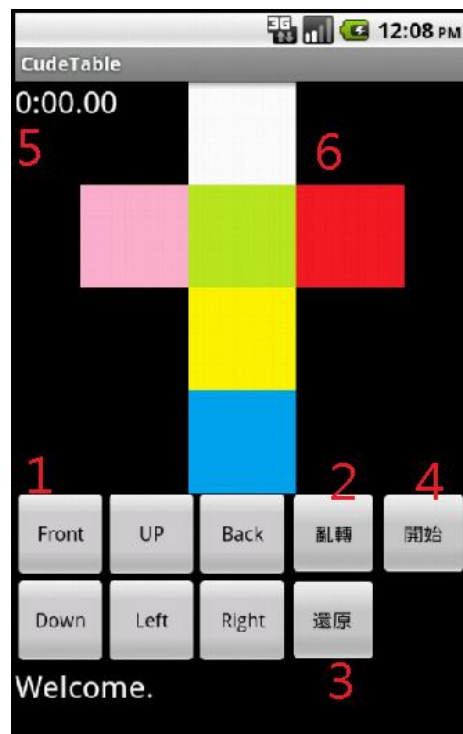


圖 5 三階魔術方塊使用展示

如何使用-計時器(圖 6)

1. 按第一次開始計時，第二次停止計時，第三次歸零。
2. 計時顯示

如何使用-亂轉公式產生器(圖 7)

1. 輸入想要多少步驟的亂轉公式
2. 點選產生按鈕
3. 亂轉公式產生結果顯示



圖 6 本研究之計時器



圖 7 亂轉公式產生器

演算法流程：

1. 此程式運作並不藉由矩陣上的操作再將資料對應到方塊圖形上。(提昇 IO 效能)
2. 見圖 8，以 Front 的轉動為例，使用 `getDrawable()` 來提取 1 號的圖形資料，並使用一個 Drawable 宣告的變數來暫存。
3. 接著提取 4 號的資料放進 1 號，3 號放進 4 號，2 號放進 3 號，暫存變數的資料放進 2 號。
4. 見圖 9，將四個角落的資料也做同樣的搬動。
5. 使用 `getDrawable()` 來提取 1 號的圖形資料，並使用一個 Drawable 宣告的變數來暫存。
6. 接著提取 4 號的資料放進 1 號，3 號放進 4 號，2 號放進 3 號，暫存變數的資料放進 2 號。

虛擬碼：

```
Drawable temp1=x1.getDrawable();
x1.setImageDrawable(x4.getDrawable());
x4.setImageDrawable(x3.getDrawable());
x3.setImageDrawable(x2.getDrawable());
x2.setImageDrawable(temp1);
```

7. 見圖 10 四個邊的資料也會因為轉動而移動所以也做同樣的搬動。
 8. 使用 `getDrawable()` 提取 A 邊的 1.2.3 號的圖形資料，並使用 3 個 Drawable 宣告的變數來暫存。
 9. 接著提取 D 組的資料放進 A 組，C 組放進 D 組，B 組放進 C 組，暫存變數的資料放進 B 組。
- 這樣便完成了整個 Front 的轉動。

虛擬碼：

```
Drawable temp1=x1.getDrawable();  
Drawable temp2=x2.getDrawable();  
Drawable temp3=x3.getDrawable();  
x1.setImageDrawable(x10.getDrawable());  
x2.setImageDrawable(x11.getDrawable());  
x3.setImageDrawable(x12.getDrawable());  
x10.setImageDrawable(x7.getDrawable());  
x11.setImageDrawable(x8.getDrawable());  
x12.setImageDrawable(x9.getDrawable());  
x7.setImageDrawable(x4.getDrawable());  
x8.setImageDrawable(x5.getDrawable());  
x9.setImageDrawable(x6.getDrawable());  
x4.setImageDrawable(temp1);  
x5.setImageDrawable(temp2);  
x6.setImageDrawable(temp3);
```

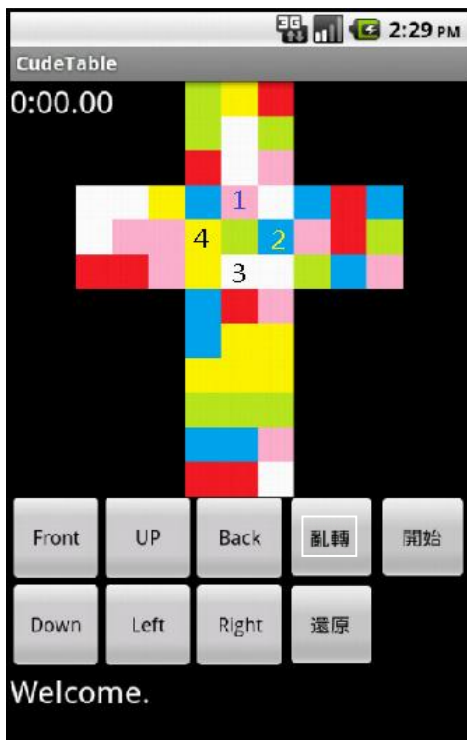


圖 8 三階魔術方塊演算法流程-1

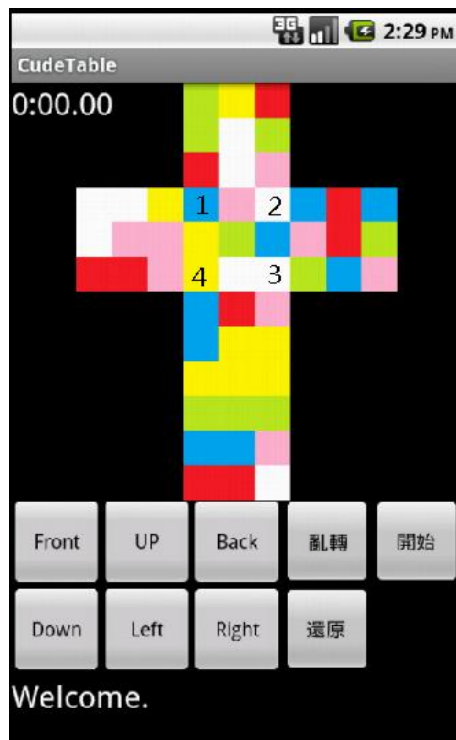


圖 9 三階魔術方塊演算法流程-2

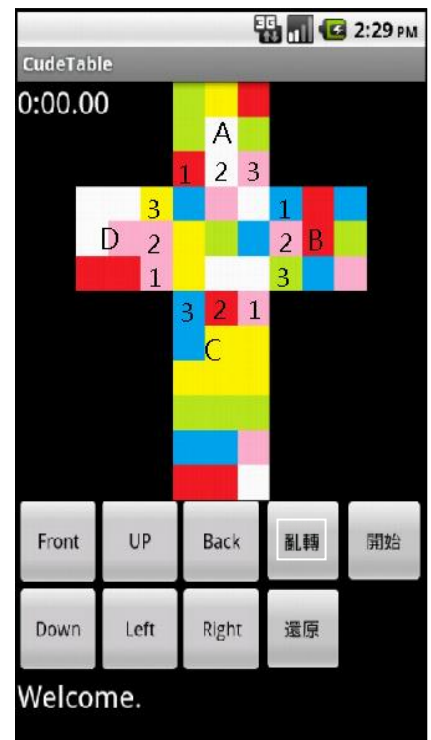


圖 10 三階魔術方塊演算法流程-3

1. 見圖 11 亂轉功能會依據官方規則產生沒有廢步的 25 個亂轉步驟。
2. 還沒有人證明出達到最大打亂狀態需要幾個步驟，但比賽會要求至少需要 20 步以上的亂轉公式，
因為 God's number is 20.
3. 亂轉公式產生規則
 - A. 第 i 步不能與第 $i-1$ 步相同面
 - B. 如果第 $i-2$ 步與第 $i-1$ 步互為對面，則第 i 步不能與第 $i-2$ 步相同
 此為簡單的產生規則，根據 WCA 官方比賽規則，亂轉公式的產生是由 cube explorer 所產生，而其中是採取較複雜的演算法。

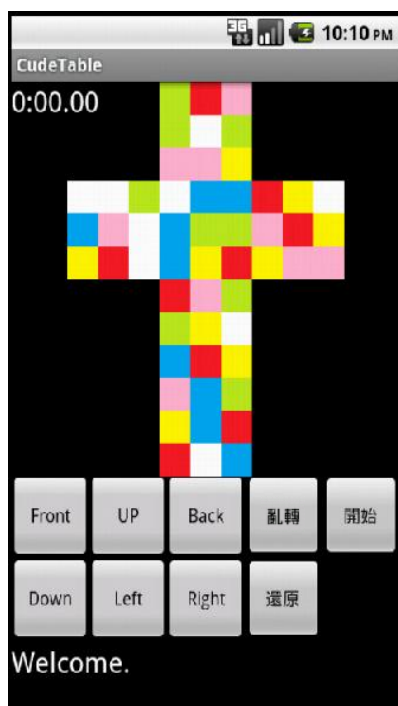


圖 11 亂轉公式演算法流程

1. 見圖 12，還原功能有三個步驟
 - A. 將方塊還原
 - B. 將計時器還原
 - C. 下方訊息列還原
 - A. 因為方塊在轉動中 6 個中心塊，相對位置永遠不會改變，所以還原時直接將 6 個中心塊的資料往四周八格複製。
 - B. 將計時器停止之後，把顯示資料還原成 0:00.00
 - C. 下方訊息列還原成 Welcome.

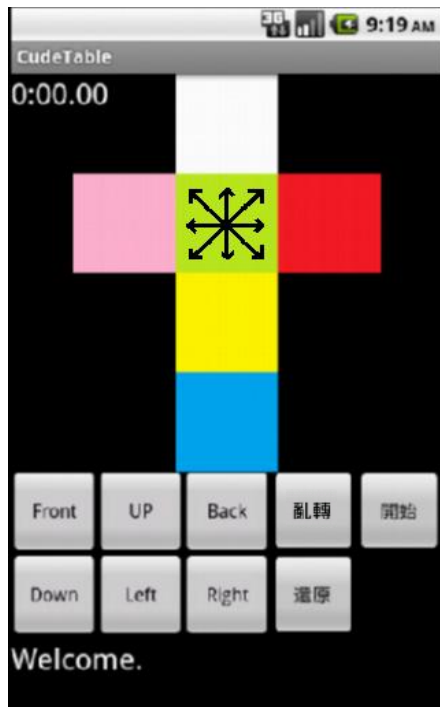


圖 12 還原功能演算法流程

1. 圖 13，此功能如果計時器是沒有被啟動的狀態，就會開始計時。

虛擬碼：

```

if(t.equals("p")){
    t="a";
    startTime = System.currentTimeMillis();
    handler.removeCallbacks(updateTimer);
    handler.postDelayed(updateTimer, delay);
}

```



圖 13 開始計時

在魔術方塊轉動的過程中，我們需要判斷魔術方塊完成與否，所以在每個方向的轉動中，我們皆需要在轉動過後加入魔術方塊完成與否的判斷。

程式碼：

```
getCudeState();
for (int x = 0 ; x < 6 ;x++ )
for(int y = 0; y<9 ; y++){
    if(cude[x][y] != cude[x][4]) return;
}
TextView time = (TextView) findViewById(R.id.TextView1);
time.setText("Finish!Time:"+t);
handler.removeCallbacks(updateTimer);
t="p";
```

1. getCudeState() 會取得目前方塊狀態存在 cude 矩陣中。
2. 根據編號 cude[x][4]剛好是魔術方塊的中心塊，因為在轉動過程中每個中心塊的相對位置永遠不會改變，所以其餘的方塊只需要跟中心塊做比對即可。

在魔術方塊的領域裡，一個很重要的事情就是計時，計時的時候也必須務求精準，在實作計時器的時候，將更新畫面頻率設定為人類反應極限的一半，可以降低系統的負擔。

虛擬碼：

計時器的方法見圖 14：

```
private Runnable updateTimer = new Runnable() {
    public void run() {
        TextView time = (TextView) findViewById(R.id.textView1);
        Long spentTime =
            System.currentTimeMillis() - startTime;
        //計算目前已過分鐘數
        Long minius = (spentTime/1000)/60;
        //計算目前已過秒數
        Long seconds = (spentTime/1000) % 60;
        Long ms = (spentTime%100);
        t = getTime(minius, seconds, ms);
        time.setText(t);
        handler.postDelayed(this, delay);
    }
};
```

其按鈕的運作虛擬碼：

```
if(Bun==0){
    startTime = System.currentTimeMillis();
    handler.removeCallbacks(updateTimer);
    handler.postDelayed(updateTimer, delay);
}
else if(Bun==1){
    TextView time = (TextView) findViewById(R.id.textView1);
    time.setText(t);
    handler.removeCallbacks(updateTimer);
}
else if(Bun==2){
    TextView time = (TextView) findViewById(R.id.textView1);
    time.setText("0:00.00");
}
Bun++;
if(Bun==3) Bun = 0;
```

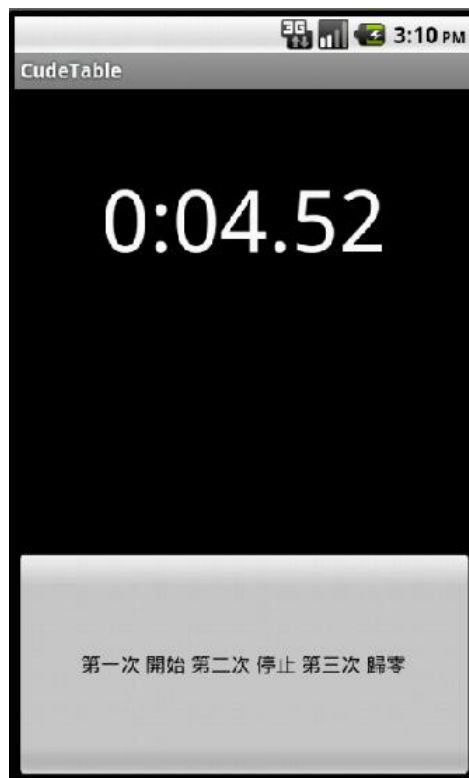


圖 14 計時器

肆、系統開發

本研究中，為可以讓更多的智慧型手機可以使用，系統選定在 android2.2 上實作，而主要開發工具為 Eclipse IDE。

表一

硬體	CPU	AMD Athlon64 X2-4800+
	記憶體	6GB DDRII
軟體	作業系統	Windows 7 Enterprise
	開發平台	Eclipse IDE for Java Developers
	開發程式語言	Java
	SDK	Android SDK

圖 15-實際運作畫面



圖 15 本研究實際運作畫面

伍、結論

本研究中，我們實作並驗證了一套可以在智慧型行動裝置的魔術方塊輔助程

式，可以幫助魔術方塊玩家在外奔波時，不再需要攜帶額外的器材，由於 android 開放的特性，除了本應用程式使用者還可以另外安裝需要的程式，讓每隻智慧型手機都具有獨特性。

誌謝

在這裡要特別感謝我的指導教授張欽智老師，不厭其煩的給予我幫助並教導我鼓勵我，也感謝 PTT 網友們熱心地回答我的問題，特別感謝這一切。
滿滿的感謝。

參考文獻

- [1] 林城，Google Android 應用程式開發實戰，碁峰資訊，2011 年 9 月
- [2] 陳會安，Java SE 7 與 Android 4.x 程式設計範例教本，碁峰，2011 年 12 月 29 日
- [3] 鄧文淵/總監製；文淵閣工作室/編著，Android 初學特訓班，碁峰，2011 年 12 月 29 日
- [4] 佘志龍、陳昱勛、鄭名傑、陳小鳳，Google Android SDK 開發範例大全、第 3 版，悅知文化，2011 年 4 月
- [5] Paul J. Deitel and Harvey M. Deitel, "Android: How to Program", Prentice Hall, February 18, 2012.
- [6] Charles Petzold, "Programming Windows Phone 7 - Microsoft Silverlight Edition", Microsoft Press, December 2010