

手機遊戲趨勢與開發設計



余執彰
萬能科技大學資訊工程系

教材大綱與目標



❧ 大綱

- ❧ 遊戲的分類與特性
- ❧ 手機遊戲的類型
- ❧ 手機遊戲的發展過程
- ❧ 手機遊戲的未來發展

❧ 目標

- ❧ 了解手機遊戲演化史以及未來可能的走向與開發目標

手機已成為日常生活的一部分

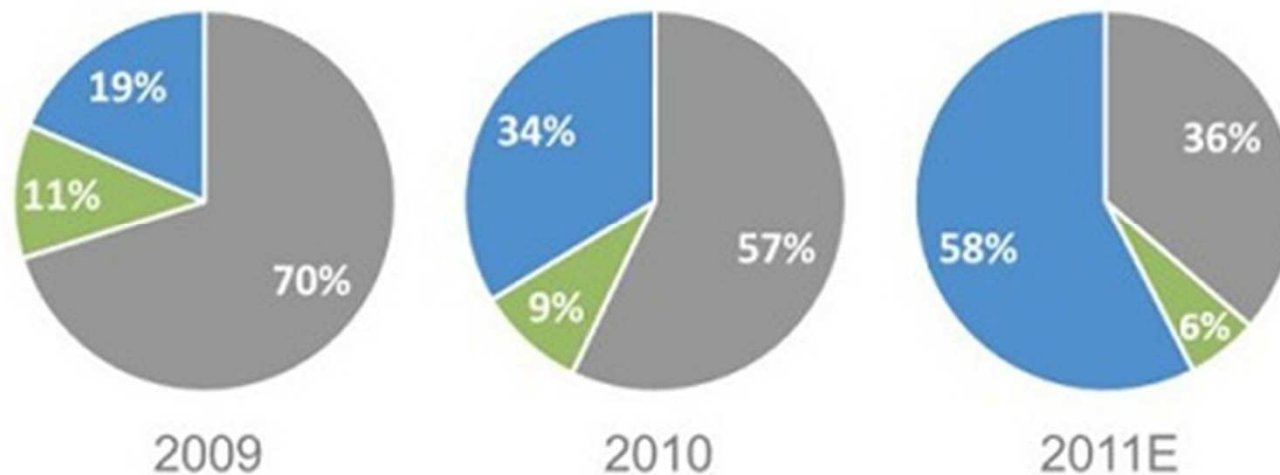


☞ 根據資策會FIND調查顯示，在台灣近九成的消費者擁有手機，其中持有智慧型手機的比例已高達15.1%。

做手機遊戲賺錢嗎？



U.S. Portable Game Software by Revenue



■ iOS & ANDROID ■ SONY PSP ■ NINTENDO DS



Source: Flurry Analytics, The NPD Group, Flurry analysis and estimates

2011 年 iPhone 付費 App 前十名



1. Angry Birds
2. Fruit Ninja
3. Angry Birds Seasons
4. Cut the Rope
5. Tiny Wings
6. Angry Birds Rio
7. Words With Friends
8. Camera+
9. Doodle Jump
10. Plants Vs. Zombies

2011 iPhone 最賺錢 App 前十名



1. Tap Zoo
2. Angry Birds
3. Zombie Farm
4. Tap Pet Hotel
5. Poker by Zynga
6. Smurfs' Village
7. Texas Poker
8. Haypi kingdom
9. Infinity Blade
10. Motion X GPS Drive

手機遊戲的獲利



- ❧ 根據IHS iSuppli的統計，2011年App市場的營業額為38億，其中apple的app store約佔了3/4
- ❧ 遊戲app在iOS平台佔了72%，在android平台上佔了60%（截至2011年Q3在美國當地的統計結果）



資料來源:IHS iSuppli

手機遊戲的獲利

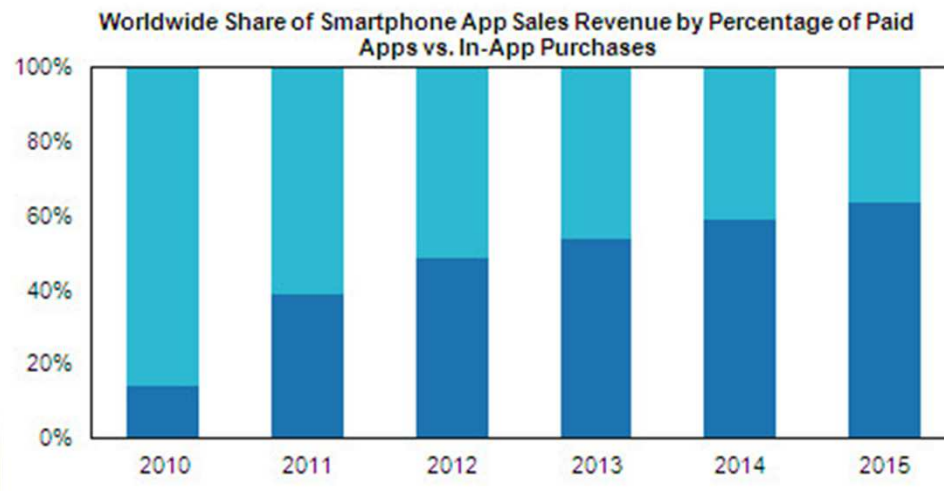


- ❧ 根據FADE LLC(Forecasting and Analyzing Digital Entertainment)的調查，2010年在iOS上的遊戲相關營業額約為75億美金
- ❧ 極度寡佔的市場
 - ❧ iPhone APP 中90%的遊戲營業額 被10%~15%的遊戲所囊括

手機遊戲的獲利



- ☞ 據2011年的統計，96%的智慧型手機使用者傾向免費下載的app
- ☞ 那手機遊戲要如何賺錢？
 - ☞ in-app purchases.
- ☞ iSuppli預估In-app purchase會在2015年佔app市場營業額的64%，大約是56億(2011年是9.7億)



資料來源:IHS iSuppli

遊戲的定義



- ❧ 遊戲，是人的一種活動或項目，就好像爬山跑步一般
- ❧ 遊戲本身具有以下四種特性
 - ❧ 行為模式
 - ❧ 規則條件
 - ❧ 娛樂/嚴肅
 - ❧ 輸贏勝負

遊戲的定義



- ❧ 行為模式：指的是整個遊戲在進行時的表現行為，任何參與遊戲的人員都必須按照這樣的流程來進行
- ❧ 任何遊戲必定會有固定的行為模式
- ❧ 例如：猜拳遊戲，玩家只能從剪刀石頭布三者取一作為輸出

遊戲的定義



- ❧ 規則條件：大家都必須遵守的遊戲行為規範，是為了遊戲的公平性而制定
- ❧ 例如：「21點」的遊戲，A是1點，J、Q、K是10點，其他的牌則是按照牌面數字決定點數，花色不論。這就是遊戲的規則

遊戲的定義



- ❧ 娛樂身心：大多的遊戲都具備有娛樂性，也就是說，玩家能藉由遊戲獲得快樂感。（也稱作casual game）
 - ❧ 一般講遊戲都是指休閒遊戲
- ❧ 學習功能：嚴肅遊戲是指設計中帶有特定學習目的的遊戲，在軍事訓練，醫療照護，企業訓練等領域廣泛地被運用。（也稱作serious game）

遊戲的定義



- ❧ 嚴肅遊戲可提供擬真的應用情境，讓學習者可在這樣的情境與相關人物互動，或應用工具完成學習任務。
- ❧ 以下是一個在應用嚴肅遊戲的例子
- ❧ http://www.youtube.com/watch?v=0UKhm3iM87Q&feature=player_embedded#!

遊戲的定義



- ❧ 輸贏勝負：任何遊戲都有勝負關係，缺少了輸贏勝負的遊戲，是沒有意義的
- ❧ 即便是單人遊戲，還是存在著勝負關係
- ❧ 例如：解謎類型的遊戲，勝負關係就是破關與否

遊戲的類型



❧ ACT/ATG(動作類)

❧ Platform game

❧ STG/FPS

❧ FTG

❧ MUG

❧ STA(策略類)

❧ RTS

❧ PZL(益智類)

❧ RPG(角色扮演)

❧ A-RPG

❧ S-RPG

❧ SLG(模擬類)

❧ SPT(運動類)

❧ RCG(競速類)

❧ AVG(冒險類)

❧ A-AVG

❧ 網頁遊戲

遊戲的類型



- ❧ ACT (Action game，動作類)
- ❧ 特色在於遊戲的流暢度與刺激感
- ❧ 故事劇情較為簡單(或是沒有劇情)，開發架構較為簡單
 - ❧ 平面動作
 - ❧ 射擊類
 - ❧ 格鬥類
 - ❧ 音樂類

動作類



- ❧ 平面動作(Platform game)
- ❧ 顧名思義，玩家的角色是在一個平面上進行一些動作(例如：蹲下、跳躍)來躲避障礙物過關的遊戲
 - ❧ 超級瑪莉兄弟(Super Mario Brothers)
 - ❧ 洛克人
 - ❧ 音速小子
- ❧ 體驗網址：
[super mario crossover](#)
- ❧ [Super mario 25週年紀念影片](#)



動作類



- ❧ 射擊類(STG, Shooting Game)
- ❧ 同樣的，射擊類遊戲就是指以射擊的方式遊玩的遊戲類型
 - ❧ 卷軸式射擊遊戲
 - ❧ 第一人稱射擊遊戲
 - ❧ 第三人稱射擊遊戲

動作類



- ❧ 卷軸式射擊遊戲
- ❧ 此類遊戲的背景，在遊玩的時候看起來像是捲軸在捲動，因此有這種名稱。
- ❧ 一般而言，玩家控制螢幕中的一架機器(例如：飛機)，對著不斷出現的敵人射出子彈，同時也必須閃避敵人射出的子彈
- ❧ 1942，雷電
(體驗網址：
<http://nintendo8.com/game/268/1942/>)



動作類



- ❧ 第一人稱射擊遊戲(First-Person Shooting)
- ❧ 以第一人稱視角進行的射擊遊戲，玩家非常容易有身歷其境的感覺，玩起來會更有刺激感
 - ❧ 德軍總部(Wolfenstein 3D)
 - ❧ 毀滅戰士(Doom)
 - ❧ 雷神之槌(Quake)
 - ❧ 戰慄時空：絕對武力(Counter-Strike)
- ❧ 體驗網址：[Doom 1](#)



動作類



- ❧ 格鬥類(FTG，Fighting)
- ❧ 玩家操縱某個角色與電腦或其他玩家所控制的角色進行打鬥
- ❧ 格鬥類遊戲通常需要較為複雜的按鍵組合(稱為招式)
 - ❧ 快打旋風(Street Fighter)
 - ❧ 格鬥天王(King of Fighters)
 - ❧ 鐵拳(Tekken)

動作類



- ❧ 音樂類(MUG, Music Game)
- ❧ 玩家須配合遊戲的音樂節奏按下畫面上對應的按鈕來獲得分數，累積到一定分數即可過關
- ❧ 這類型的遊戲常常會搭配時下最流行的音樂，以求不斷的吸引玩家繼續遊玩
- ❧ DDR
(Dance Dance Revolution)
- ❧ 太鼓達人



遊戲的類型



❧ PZL (Puzzle, 益智類)

- ❧ 不強調華麗的聲光效果，注重玩家的思考與邏輯判斷
- ❧ 不同於動作類遊戲，通常不需要繁複的按鍵組合
- ❧ 魔術方塊
- ❧ 踩地雷
- ❧ 大富翁



遊戲的類型



❧ STA (strategy game, 策略類)

- ❧ 策略類遊戲也是屬於讓玩家思考的類型
- ❧ 和益智類不同的是，策略類遊戲較為偏向對戰(玩家vs.電腦或玩家vs.玩家)，讓玩家運用個人的思緒來佈局，以擊敗對方玩家(電腦)為目的的遊戲
- ❧ 象棋、圍棋等棋藝型遊戲

遊戲的類型



- ❧ 策略類遊戲是由棋藝型遊戲開始發展，後來慢慢衍生出所謂的戰棋型遊戲。由於電腦硬體的運算能力限制，這些遊戲多半採取回合制的遊戲方式
- ❧ 回合制的特色在於雙方(或多方)玩家輪流進行遊戲，只有再輪到自己的回合時，才能進行操控。
- ❧ 然而，回合制的遊戲方式較缺乏遊戲的緊湊性。

遊戲的類型



- ❧ 後來在遊戲中加入了即時的概念，形成了所謂的即時戰略(real-time strategy, RTS)類型的遊戲
- ❧ 所謂的即時戰略，就是說遊戲的過程中，雙方的操控是「同時」在進行的，包含資源採集，建造建築，科技研發與兵種生成等等
- ❧ 由於雙方的佈陣、操作、策略都隨時隨地的在變化，大大的提高了遊戲的刺激性與緊張感
 - ❧ 星海爭霸(Starcraft)
 - ❧ 魔獸世界(Warcraft)
 - ❧ 世紀帝國(Age of Empire)

遊戲的類型



☞ 星海爭霸2遊戲畫面



遊戲的類型



∞ SLG (simulation game，模擬類)

- ∞ 電腦提供玩家一個模仿真實世界某種行為的遊戲系統，並運用電腦來模擬出各種可能發生的狀況與事件
- ∞ 無論是畫面或是遊戲性，模擬類遊戲都非常強調所謂的「擬真性」，要讓玩家有身歷其境的感覺

模擬類遊戲



- ❧ 早期模擬類遊戲比較偏向嚴肅遊戲(serious game)，例如對飛行員或是太空人的訓練。
- ❧ 這些都屬於較為昂貴與危險的工作，如果用實際工具進行訓練的話，所付出的成本會過高，因此用模擬類遊戲進行訓練人員的臨場反應是非常實用的做法。
- ❧ 後來模擬類遊戲慢慢發展為「藉由讓玩家在遊戲中達成現實生活無法做到的事，來達到娛樂的目的」

模擬類遊戲



☞ 模擬類遊戲列表

☞ 載具型

- ☞ 急速快感(Need for Speed)
- ☞ 模擬飛行(Flight Simulator)

☞ 經營型

- ☞ 模擬城市(SimCity)
- ☞ 鐵路大亨(Railroad Tycoon)

☞ 養成型

- ☞ 模擬市民(The Sims)
- ☞ 美少女夢工廠

遊戲的類型



- ❧ SPT (Sports，運動類)
- ❧ 運動類遊戲可歸類為模擬類遊戲的一種，只是把重點擺在運動項目
- ❧ 模擬類遊戲較注重機具(飛機、車)的物理運動，運動類遊戲則注重人體模型運動的擬真性
- ❧ 許多玩家可能因為現實生活中喜愛某位運動員而喜歡上運動類遊戲，藉由操作該球員來達到某種的娛樂性
- ❧ NBA 2K/NBA Live系列、FIFA/WINNING ELELVEN系列

遊戲的類型



- ❧ RPG(Role-Playing Game，角色扮演類)
- ❧ 玩家扮演一個以上的角色，在一個模擬的世界中進行，這個角色就如同現實生活一般，會隨著玩家遊玩的歷練(俗稱「經驗值」)而成長
 - ❧ 太空戰士(Final Fantasy)
 - ❧ 軒轅劍
 - ❧ 仙劍奇俠傳
 - ❧ 魔法門(Might & Magic)



角色扮演類



- ❧ RPG遊戲通常具備較長的劇情，且對遊戲的世界觀有完整的說明
- ❧ 例如：「艾澤拉斯」是眾多種族的誕生地，如精靈、人類、矮人、地精等。有一天，邪惡的大軍「燃燒軍團」打破這個世界的和平，帶來了獸人、德萊尼等種族。艾澤拉斯的人民奮力的和這群惡魔對抗，至今尚未停歇...

角色扮演類

- ❧ RPG遊戲通常都存在所謂的「升級系統」，角色會根據遊戲的過程獲得經驗值(experience)而升級
- ❧ 為了突顯出不同玩家角色間的差異，遊戲人物往往會有所謂的「屬性」，例如力量、敏捷、體力等等



暗黑破壞神遊戲畫面

角色扮演類



除了屬性之外，許多RPG遊戲還加入了「技能」特性，增加了遊戲的耐玩度與變化性



暗黑破壞神遊戲畫面

遊戲的類型



- ❧ AVG (Adventure, 冒險類)
- ❧ 可定義成「沒有升級(經驗值)系統的RPG遊戲」
- ❧ 注重的是玩家的解謎與探索過程，讓玩家不停的思考去解決問題
- ❧ 在遊戲中，如果遇到某些情況無法繼續進行遊戲，稱為「卡關」
- ❧ RPG遊戲的卡關，也許可利用角色的成長來突破，但冒險類的遊戲常常需要靠邏輯上的思考來解決
- ❧ 鬼屋魔影(Alone in the Dark)
- ❧ 惡靈古堡(Biohazard)
- ❧ 大航海時代



遊戲的類型



☞ 複合型

- ☞ 遊戲類型發展至今，已經很難用單一特色去進行分類
- ☞ 目前市面上非常多的遊戲，幾乎都是結合了各種的遊戲類型所發展出的複合型遊戲

遊戲的類型



- ❧ 例如，以古墓奇兵(Tomb Raider)而言，遊戲中除了冒險的成分以外，玩家所扮演的主角還必須做出翻滾、攀岩、奔跑等動作，使得遊戲中附加了動作遊戲的因子
- ❧ 這樣的遊戲，通常稱做動作冒險類(A-AVG)遊戲
 - ❧ 薩爾達傳說(Legend of Zelda)、俠盜獵車手(Grand Theft Auto)、波斯王子(Prince of Persia)等
- ❧ 這類遊戲的動作成分不會佔太重，避免模糊原本是冒險類遊戲的精神

遊戲的類型



- ❧ 同樣的道理，如果一個RPG遊戲中包含了動作遊戲的元素的話，可稱做動作角色扮演(A-RPG)遊戲
- ❧ 遊戲角色除了具備RPG遊戲所具有的升級系統外，在戰鬥場景的處理方面較為偏向動作遊戲(表示並非傳統RPG的指令型態)
- ❧ 永遠的伊蘇
- ❧ 暗黑破壞神

遊戲的類型



❧ 如果一個RPG遊戲中在戰鬥場景的部分加入了戰略思考的特性的話，稱做戰略角色扮演(S-RPG)遊戲

❧ 這類遊戲在戰鬥場景的表現上偏向策略類的遊戲，玩家在戰鬥中必須考慮部队的配置、移动、攻击等，以达到胜利的目标

❧ 天使帝國

❧ 風色幻想



遊戲的類型



- ✧ 近年來由於網際網路的蓬勃發展，遊戲的類型也多了一種新的類型：「網頁遊戲」
- ✧ 所謂的網頁遊戲，便是玩家不需要安裝任何遊戲主程式在自己的機器上，只需要安裝瀏覽器便可以利用瀏覽器來遊玩遊戲
- ✧ 傳統的遊戲常常需要安裝對應的程式在主機上，對於一些上班族或是需要東奔西跑通勤的人來說，要隨時進行遊戲並不是非常的方便

網頁遊戲範例



遊戲的類型



- ❧ 網頁遊戲不需要「安裝」的動作，用瀏覽器就可以遊玩，這是它有別於傳統遊戲的優點
- ❧ 這類型的遊戲的開發方式有兩種
 - ❧ 網頁語言，如PHP/ASP/JSP/JavaScript/CGI等
 - ❧ 高階語言，如Java或是Flash

網頁遊戲的特色



❧ 安裝容易

❧ 基本上可說是不需安裝，相當方便

❧ 硬體需求低

❧ 現在的單機遊戲為了追求炫麗的聲光效果，常常需要高規格的硬體，但筆記型電腦已漸漸成為主流的現代，硬體上的升級難度要比桌機來的高，因此低硬體需求的網頁遊戲反而成為筆記型電腦使用者來說門檻較低

❧ 網路社群功能

❧ 這類型的遊戲重點是擺在網路上各種玩家的互動，因此一個良好的互動平台是很需要的

遊戲的類型



- ❧ 網頁遊戲的蓬勃發展，應該可以歸因於一個遊戲：Trivial
- ❧ Trivial被歸類為MMORTS，主要原因是它具有RTS的特性，但又是MMO的玩法
- ❧ Trivial有一個和傳統遊戲不同的特點，就是遊戲內時間步調跟現實世界是一樣的
- ❧ 遊戲中的指令大部分是無法透過遊戲中的特殊指令來回復的，必須要等待一段時間來回復，而這個回復的時間也是跟現實生活中的時間是一樣的。

遊戲開發平台



- ❧ 傳統上遊戲的開發平台為以下四種
 - ❧ 街機：指的是放在公共場所的遊樂機器，又稱作「大型機台」
 - ❧ 家機：又稱作「電視遊樂器」，如PS3、XBOX、Wii等
 - ❧ 桌機：指的就是桌上型電腦的遊戲
 - ❧ 掌機：又稱掌上型遊樂器，如Game Boy、Game Gear、Nintendo DS、PSP等等
- ❧ 隨著科技的進步，現在又多了一種平台：手機

手機遊戲 VS. 掌上型遊樂



- ❧ 2005年之前，如果你問：「出門時想要玩遊戲，該買什麼？」大部分的人會回答你：「掌上型遊樂器！」
- ❧ 2010年之後，如果你問：「出門時想要玩遊戲，該買什麼？」大部分的人會回答你：「iPhone！」
- ❧ WHY?

手機遊戲 VS. 掌上型遊 樂器

- ❧ 過去幾年，掌上型遊樂器的市場一直是任天堂和SONY 兩家大廠的天下，尤其是任天堂公司，從gameboy到NDS，任天堂一直是處在龍頭的位置上。
- ❧ 掌上型遊戲機的定位就是玩遊戲，功能單一因此可以針對遊戲特性進行開發。手機遊戲雖然在手機問世沒多久就開始發展，但一直無法和掌上型遊樂器相提並論。
- ❧ 原因分析如下：
 - ❧ 運算能力太低
 - ❧ 記憶體容量過小
 - ❧ 廠牌型號太多，造成硬體設備琳琅滿目，若為單一規格設計遊戲，根本無法普及化

手機遊戲 VS. 掌上型遊樂器

- ❧ 用手機平台搶佔掌上型遊樂器市場，並不是由Apple或Google所發起的
- ❧ 早在2003年，Nokia就推出了創新的手機產品 N-Gage
- ❧ N-Gage的概念從外型上一看就可以知道，直接就是手機+掌上型遊樂器的合體



手機遊戲 VS. 掌上型遊 樂器

- ❧ N-Gage一共有兩個系列，N-Gage(2003)與N-Gage QD(2004)，但其銷售大大不如預期，NOKIA 終於宣布N-Gage遊戲服務將於2010年終止。
- ❧ N-Gage失敗的原因可歸咎於不方便的遊戲操作以及過於昂貴的遊戲售價



手機遊戲 VS. 掌上型遊 樂器

- ❧ 但 NOKIA 在N-Gage上的失敗並不代表整個手機遊戲產業失敗了，Apple iPhone 以優異的硬體性能及軟體平台重新顛覆了手機遊戲產業，
- ❧ 而且目前智慧型手機的演化速度越來越快，硬體功能越來越強，當然也就能開發出畫面越來越華麗的遊戲。

手機遊戲 vs. 掌上型遊樂器



- ❧ 2011年遊戲版本的金式世界紀錄也頒給了iPhone與App store，包括
- ❧ 銷售速度最快的遊戲平台——iPhone 4
 - ❧ iPhone 4首日銷售量為150萬台。PSP在2005年時首日銷量為20萬台，而NDS在2004年11月首週銷量為50萬台。
- ❧ 最受歡迎的軟體商店——Apple App Store
 - ❧ 到2010年九月為止，總共有259,470個軟體可供購買或免費下載。
- ❧ 最大的遊戲下載商店——Apple App Store
 - ❧ 到2010年九月為止，總共有37,362個遊戲可供購買或免費下載。而PC遊戲下載商店Steam提供大約1,110個遊戲，Xbox Live Arcade則大約為1,300個，Wii的虛擬商店則提供576個遊戲。

手機遊戲 vs. 掌上型遊樂器



❧ 首發陣容最大的遊戲平台——Apple App Store

❧ Apple App Store在2008年7月10日推出時，有超過500個軟體，其中有145款遊戲，讓App Store成為史上首發陣容最大的遊戲平台。

❧ 在最多國家排行第一的付費遊戲——Angry Birds

❧ 到2010年10月為止，Angry Birds已在67個國家達成付費遊戲排行第一的紀錄。而根據出版商的資料，到2009年12月10日為止，Angry Birds已經售出超過650萬份。

❧ 銷售速度最快的一iPhone/iPod 戰略遊戲——Plants vs. Zombies

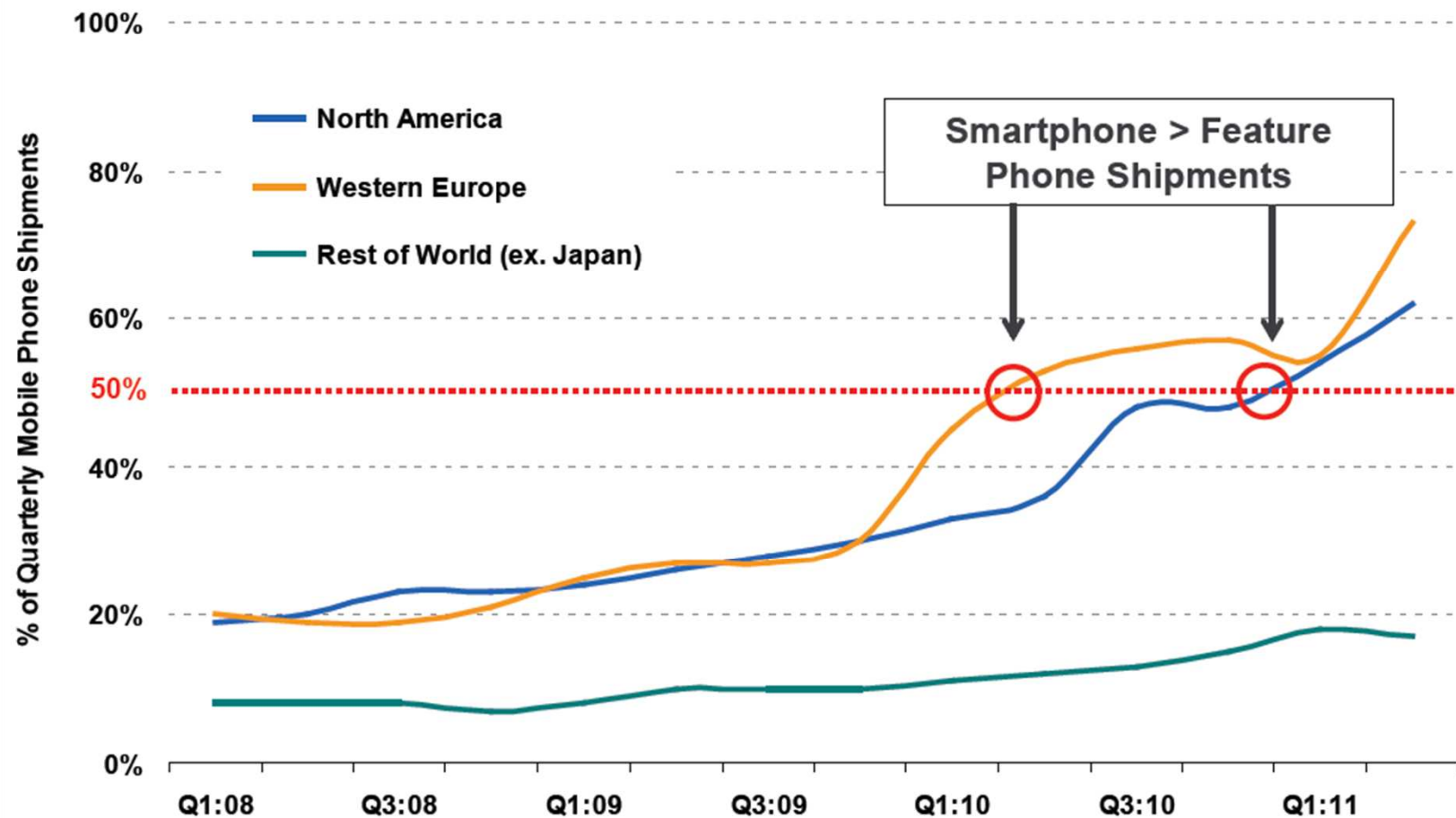
手機遊戲 VS. 掌上型遊 樂器

- ❧ 手機遊戲龐大的商機在哪？
 - ❧ 潛在的市場。
- ❧ 現在的人要出門有三個必備物品：錢包、鑰匙與手機。
可見手機已經是現代人的生活中不可或缺的一個工具。
- ❧ 此外，現代人對於電子產品的要求也從「功能化」逐漸轉為「全能化」，這一點可以從手機的演化方式看出一些端倪

手機遊戲 VS. 掌上型遊 樂器

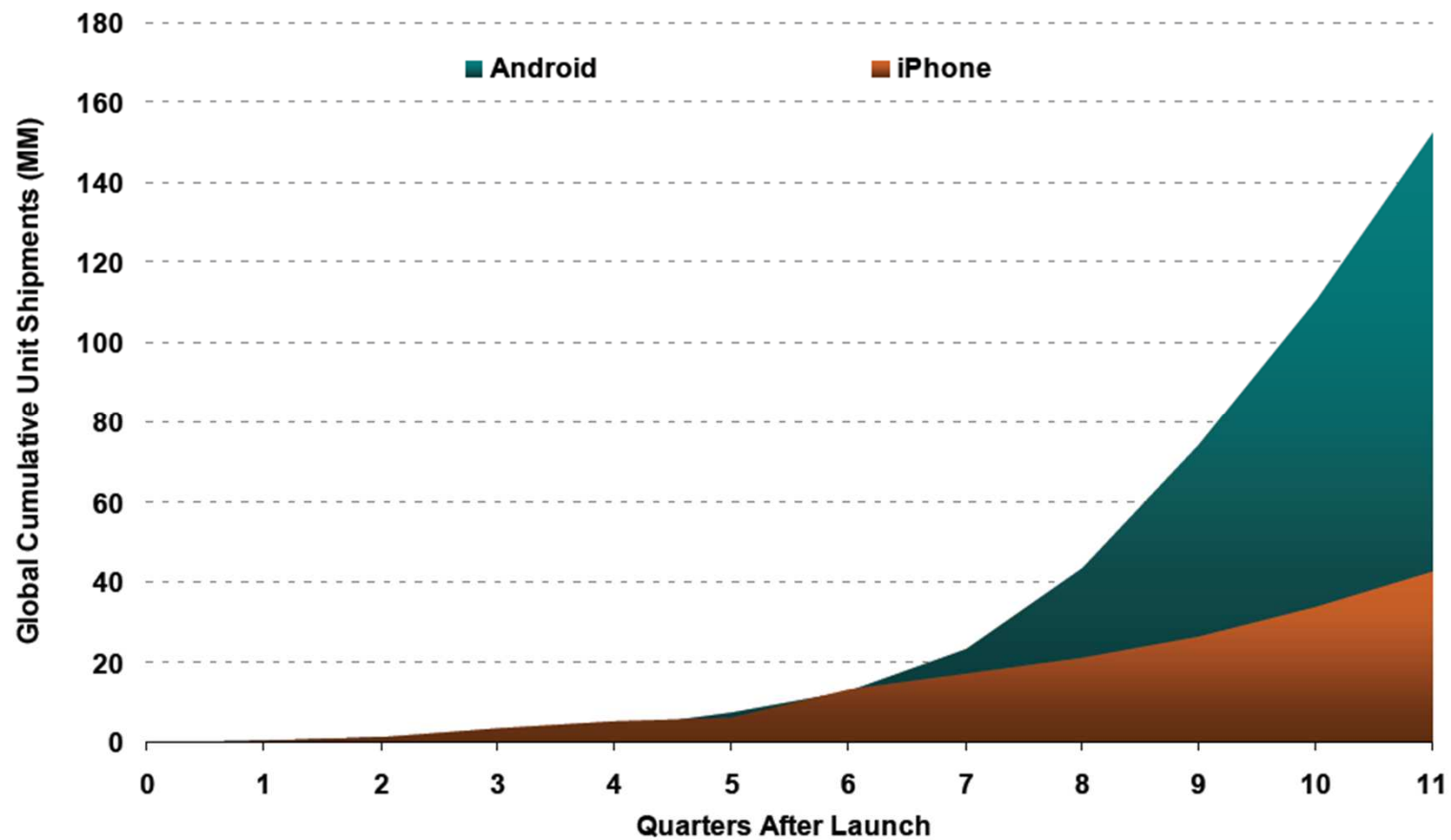
- 對每個人來說，如果能夠有一台同時間可以照相、通話、上網、遊戲、選擇產品。這等一台而足的非攜帶型機器，想必功能強大。
- 此外，和手機比起來，掌上型遊戲樂器完全屬於在市場上的小眾，銷量機任破天。突破在2010年，NDS主是年中，但第一季在2012年就賣了超過一億隻，而iPhone的銷量也預估在2012年可賣超過八千萬隻，潛力從無窮。

全球手機出貨量分析



資料來源：Morgan Stanley Research. 註：日本在Q4:08時智慧型手機銷量就已超越一般手機

Android vs. iPhone



資料來源：Morgan Stanley Research

手機遊戲 VS. 掌上型遊 樂器

- ❧ 而任天堂公司的總裁岩田聰先生也指出，假如任天堂不能證明未來任天堂的掌上型遊戲機與產品有別於手機或 iPhone 上的遊戲有何不同，那麼任天堂的掌上型遊戲未來是黑暗的。
- ❧ 連任天堂的總裁都有如此看法，可見手機遊戲的確在未來有挑戰掌上型遊戲市場的可能

手機遊戲發展



手機遊戲的型態



- ❧ 手機遊戲可根據其畫面的表現型態分為以下兩種
- ❧ 文字型
 - ❧ 遊戲中的各種場景、情境、事件等，都以文字或數值的方式來呈現的遊戲型態
- ❧ 動畫型
 - ❧ 利用一些動畫的展現來表現遊戲內涵
- ❧ 通常圖像型的遊戲較受到一般玩家的歡迎

文字型手機遊戲



- ❧ 以虛擬寵物而言，手機畫面可能顯示如下：寵物飢餓度：70，口渴度：20，疲勞度：20，下面有數個選項：1. 進食2. 喝水3. 休息等等……」
- ❧ 虛擬資料(寵物狀態)儲存在伺服器上，玩家僅透過按鍵操作的方式，將指令送至伺服器端，由伺服器端更新資料後再回傳至手機畫面上顯示

手機遊戲發展



- ❧ 手機遊戲的發展過程，與手機的硬體與介面有密不可分的關係
- ❧ 硬體：由於大多數手機的機能所限，手機遊戲對於畫面的要求不能太高

手機遊戲發展



- ❧ 介面：以電腦遊戲而言，有三個主要操作方式：鍵盤、滑鼠、搖桿
- ❧ 早期的手機訴求仍以「通話」為主軸，並無滑鼠與搖桿的設計概念
- ❧ 因此，初期手機遊戲的開發是以鍵盤操作為主

手機遊戲發展



- ❧ 早期的手機鍵盤通常也只有數字鍵盤與接通、掛斷等按鍵
- ❧ 由於手機上的按鍵不足, 再加上按鍵並不大, 對於需要要求按鍵組合的遊戲類型(例如: 格鬥類)來說幾乎不適用



圖片來源:<http://semiloose.blogspot.com/>

手機遊戲發展



❧ 早期的手機遊戲發展方向

❧ 聲光效果低

❧ 操作簡單

❧ 由於這兩項要求，益智類型(PZL)的遊戲變早早就在手機上開發了

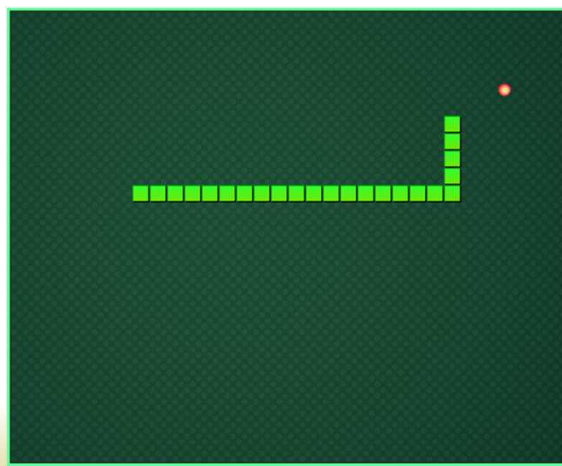
❧ 貪食蛇

❧ 俄羅斯方塊

手機遊戲發展

❧ 貪食蛇

- ❧ 玩家操控一條細長的直線（就是所謂的蛇），它會不停前進，玩家只能操控蛇的頭部往上下左右四個方向前進，一路吃掉觸碰到的食物（稱作豆），過程中必須避免觸碰到自身或者其他障礙物（如牆壁）。每次貪食蛇吃掉一個食物，它的身體便增長一些，速度更逐漸加快，讓遊戲的難度漸漸變大。



手機遊戲發展



- ❧ 貪食蛇在手機上發展的原因
 - ❧ 操控簡單(只需使用上下左右四鍵, 可剛好利用數字鍵盤的2468對應)
 - ❧ 貪食蛇考驗的是個人的敏捷與反應, 對玩家技巧的要求遠超過畫面的要求(硬體需求低)

手機遊戲發展



- ❧ 俄羅斯方塊
- ❧ 各種方塊從畫面上方開始緩慢落下，玩家可以做的操作有三種：
 - ❧ 以90度為單位旋轉方塊
 - ❧ 以格子為單位左右移動方塊
 - ❧ 讓方塊加速落下
- ❧ 方塊移到區域最下方或是著地到其他方塊上無法移動時，就會固定在該處，而新的方塊出現在區域上方開始落下。
- ❧ 當區域中某一列橫向格子全部由方塊填滿，則該列會消失並成為玩家的得分。同時刪除的列數越多，得分指數上升。
- ❧ 當固定的方塊堆到區域最上方而無法消除層數時，則遊戲結束。

手機遊戲發展



- ❧ 俄羅斯方塊在手機上發展的原因
 - ❧ 操控簡單
 - ❧ 旋轉:利用數字鍵盤的5
 - ❧ 左右移動:利用數字鍵盤的4, 6
 - ❧ 落下:利用數字鍵盤的8

手機遊戲發展



- ❧ 手機遊戲在早期的發展速度並不快，除了硬體與操作方式的原因之外，一個很大的因素是不易開發
- ❧ 不易開發的主要原因是由於每一家手機廠商都不易開發自己的作業平台，遊戲公司想要在手機上開發遊戲，便需要針對各家手機的平台特性來開發，非常的耗費時間與人力
- ❧ 再加上手機平台更新的速度非常的快，若是每次有新的手機出來，廠商都必須再為其重新開發一次應用程式，顯得相當沒有效率。

手機遊戲發展

- ❧ 手機遊戲的正式蓬勃發展，可以從SUN Microsystems發展出J2ME技術算起。
- ❧ J2ME(Java 2 Platform, Micro Edition)是為了機上盒、行動電話和PDA之類的嵌入式小型電子設備所提供的Java語言平台
- ❧ J2ME發展之後，在手機上開發遊戲問題就變小了，最主要的原因就是JAVA跨平台的特性。
- ❧ 手機廠商只要在其作業平台上提供可以執行J2ME的JAVA虛擬機器，便可以執行利用J2ME所開發出來的應用程式，這提高了軟體廠商在手機上開發應用程式的意願。
- ❧ 此外，開發J2ME的程式並不需要特別的開發工具，程式設計師只需要裝上Java SDK 及下載免費的 Sun Java Wireless Toolkit 就可以開始編寫J2ME程式，編譯及測試，非常的方便。

手機遊戲發展



- ❧ 由於嵌入式設備的種類繁多，並且之間的特性又大不相同，為了滿足不同設備上的特定需求，J2ME使用了三種架構，分別是Configuration、Profile以及Optional Packages
- ❧ Configuration包含了一個虛擬機器，針對消費性電子產品的網路連結能力、處理器速度，記憶體容量等特性為之分類。Configuration提供了開發人員JAVA平台最核心的部分，針對所使用的裝置，定義了一些較基本的核心類別函式庫。

手機遊戲發展



- ❧ Profile可以說是Configuration之上定義另一組額外的類別函式。針對特定領域細分特定用途的函式庫和API
- ❧ 以手機來說，J2ME便為其定義了MIDP (Mobile Information Device Profile)，MIDP針對手機定義了適用於手機的類別函式，例如簡訊的傳送，聲音的處理等。
- ❧ Optional Packages則獨立提供附加的、模組化和更多樣化的功能。例如多媒體、藍牙等等。

手機遊戲發展



- ❧ 也由於J2ME的誕生，手機的功能也慢慢從通話再附帶了行動遊戲的平台
- ❧ 在J2ME發展後，許多其他類型的遊戲便慢慢開始在手機上開發
- ❧ 例如，一些早期的RPG遊戲
- ❧ 相較之下，RPG遊戲的劇情比畫面上的要求要來的高
- ❧ 操作上也並不複雜，簡單的RPG遊戲只需要上下左右移動、確定、取消這些按鍵即可
- ❧ 一些簡單的平面動作遊戲也適合在手機上開發

手機遊戲發展



- ❧ 但是，J2ME當初的設計架構並沒有包含一些智慧型手機的硬體功能，如觸控式螢幕，重力加速度感應器等
- ❧ 然而，手機的演化速度實在是太快，硬體設備不斷的更新強化，而且硬體廠商與規格也非常多，為單一規格廠商設計特定的開發環境並不適合

手機遊戲發展



- ❧ 有鑑於此，Google的Android SDK，便是以JAVA語言為主的開發工具
- ❧ 和J2ME不同的是，Android SDK保留了J2ME與J2SE共用的核心函式，然後把J2ME獨有的部分加以重新改寫開發成適合Android平台手機硬體的相關功能
- ❧ 這樣做不但具有JAVA跨平台的好處，同時也能適應手機硬體的不斷演化(在作業系統版本相同的前提下)

手機遊戲發展



- ❧ 而iPhone的情況則較為特殊，由於Apple公司對於iPhone的硬體規格與作業系統都有固定的標準，因此跨平台或是跨規格的問題在iPhone上並不存在
- ❧ 因此iPhone程式的開發工具只需要針對iphone的規格定義即可。
- ❧ 也因為如此，iPhone SDK只能在 Intel 系統及Mac OS X 上執行

手機遊戲發展



而windows mobile系列手機，Microsoft也有推出 WM SDK，是以C++與.NET為主的開發環境

手機遊戲發展



- ❧ 隨著科技的進步，手機的硬體越來越好，連帶的手機遊戲的畫面也越來越華麗
- ❧ 由於硬體進步的關係，遊戲廠商可以利用OpenGL、DirectX這樣的API來發展3D遊戲
 - ❧ WindowsMobile.DirectX
 - ❧ OpenGL ES (opengl for embedded system)
- ❧ OpenGL ES 可以算是 OpenGL 精簡版，捨去了一些複雜圖形繪製函式。

手機遊戲發展



- ❧ 雖然J2ME讓手機遊戲的發展有了突破，但遊戲的「操作方式」仍趨於傳統(也就是用數字鍵的方式)
- ❧ 對於一些無法用數字鍵盤達到理想操作的遊戲，通常是採用「降低遊戲難度」或是「簡化操作方式」的方式處理
- ❧ 例如，賽車類(RCG)的遊戲，由於鍵盤無法提供像方向盤一般平滑的轉動，通常是採用如下的處理方式：
 - ❧ 加速：數字鍵的2
 - ❧ 煞車：數字鍵的8
 - ❧ 轉彎：數字鍵的4與6

手機遊戲發展



- ❧ 而運動類(SPT)的遊戲，例如棒球，可能就是只把投球與打擊的操作交給玩家處理，而守備可能就交由電腦自動幫忙處理
- ❧ 對於像運動類這種比較講求擬真性的遊戲而言，這樣的操作方式略為失去遊戲的樂趣
- ❧ 因此，在手機硬體已經足以應付遊戲需求時，大家開始思考「如何改變手機遊戲的操作方式？」

手機遊戲發展



- ✧ 近年來(智慧型)手機的兩大硬體設備的開發，讓手機遊戲的發展有了更寬闊的未來
- ✧ 觸控式螢幕
- ✧ 三軸重力感測器
- ✧ 這兩項裝置主要改變的是手機遊戲的「操作方式」

重力感測器



- ❧ 重力感測器(G sensor)，或稱作線性加速計 (Accelerometer)，是用來提供速度和位移資訊的產品。
- ❧ 這類感測器的運作方式是經由測量裝置的一些微小的物理量的變化，如電阻/電容值等，再以電壓信號來表示這些變化量。

重力感測器



- ❧ 重力感測器原本並不受到注目，直到Wii的問世以後才漸漸引起人們的注意。
- ❧ 目前的重力感測器是以微機電系統(Micro Electro Mechanical Systems, MEMS)製程為主，原因是這項製程技術非常成熟，因此成本非常低廉。
- ❧ 重力感測器可以用來測量目前裝置的傾斜角度(通常是pitch、yaw、roll三個值)，基本原理都是讓重力感測器在不同的傾斜角度輸出不同的電壓值，再經過程式計算出數值。

手機遊戲發展



- ❧ 觸控式螢幕在手機上的誕生，讓手機遊戲的操控方式有突破性的改變
- ❧ 由於目前手機的功能不再只是單單通話而已，還有記事、遊戲等需求
- ❧ 為了滿足這些需求，手機的螢幕便逐漸的增大。
- ❧ 為了便利性的考量，螢幕的增大就自然而然的壓縮到手機鍵盤的大小(甚至消失改用虛擬鍵盤或是滑開式鍵盤代替)

手機遊戲發展



- ❧ 對於遊戲來說，使用觸控式螢幕的好處就是，不必侷限於數字鍵盤的配置方式來進行操控的配對(甚至可以發展出連電腦上都沒有的新的操控方式!)
- ❧ 但是無論是何種類型的遊戲，在手機上的操作方式都不外乎一個原則：簡單與直覺
- ❧ 如果遊戲的操作按鍵過多，在畫面上的呈現反而會顯的擁擠，進而影響遊戲性

手機遊戲發展



ACT

- 可以直接用十字鍵盤圖示跟按鈕圖示來表示操作方式
- 這樣的做法，對於玩習慣掌上型遊樂器的玩家來說更為直覺



掌上型遊樂器按鍵(左)跟觸控螢幕上的虛擬按鍵(右)

手機遊戲發展

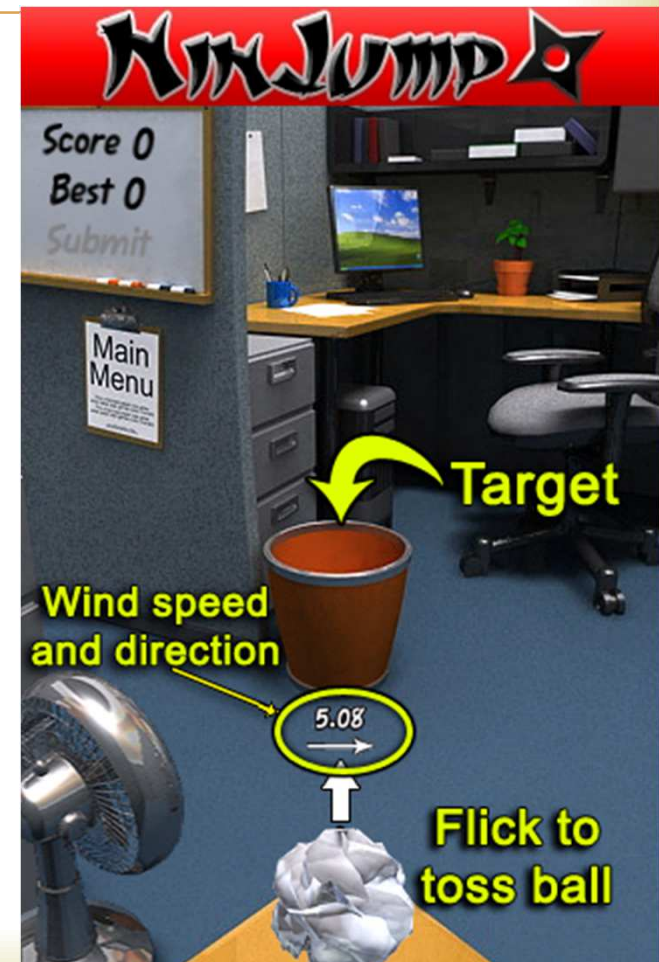


- ❧ 傳統的平面動作或是射擊遊戲都可以用這樣的操作方式
- ❧ 而衍伸出的新操作方式，則是利用觸控式螢幕的特性，玩家可以利用手指在螢幕上碰觸的方式來遊玩
- ❧ 此外，手機上還可以支援「多點觸控」的操作方式，這可是傳統遊戲所沒有的！

手機遊戲(動作類)



- ❧ 例如：Paper toss
- ❧ 玩家的目標是把紙團丟進垃圾桶內，
玩家必須根據畫面上顯示的風速及
風向，判斷紙團丟出的方向
- ❧ 紙團丟出的方向，可以利用偵測玩
家的手指在畫面上滑動的方向來決
定



手機遊戲(動作類)



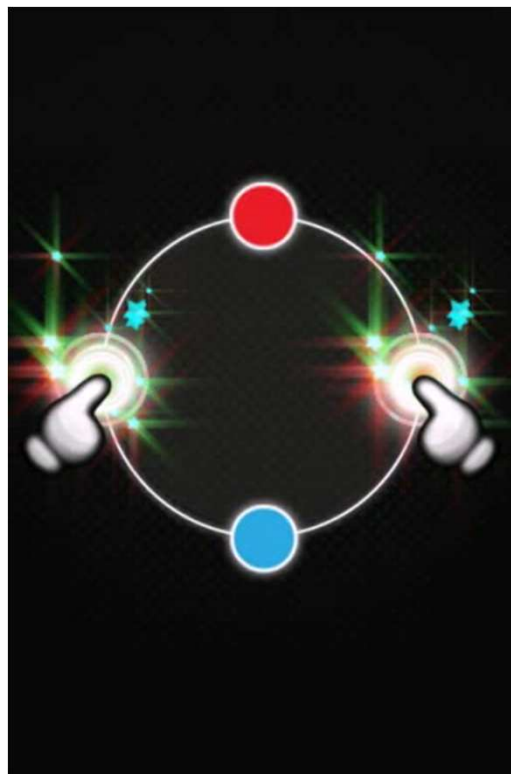
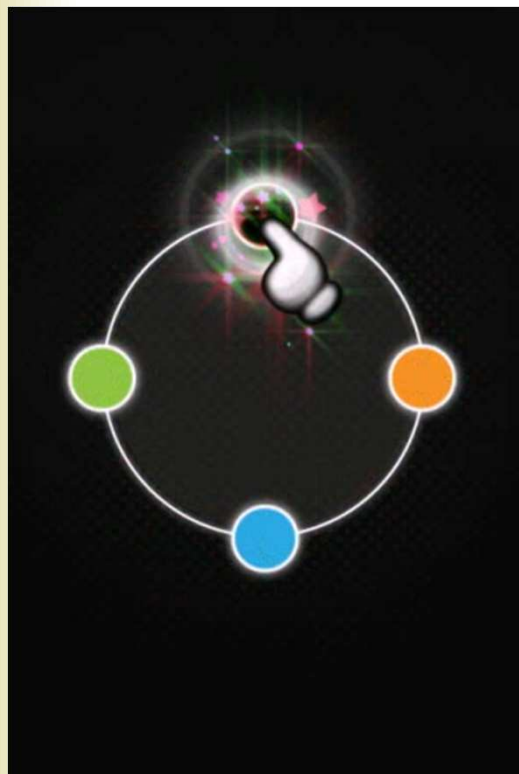
- 除了偵測手指移動方向外，更進一步的，我們可以偵測玩家的手指在畫面上移動的距離 d 與停留的時間 Δt
- 利用 $d / \Delta t$ 我們可以判斷玩家手指的移動速度，在適當的換算成物體丟出的速度
- 例如：丟硬幣遊戲

手機遊戲(音樂類)



- ❧ 對於DDR類的音樂遊戲來說，仍然可以用十字鍵盤圖示的方式進行操作上、下、左、右
- ❧ 然而，新的操控方式則改成玩家必須在適當的時間在特定的位置觸碰螢幕，甚至是在螢幕上滑行過特定的軌跡，這是以往音樂類遊戲所沒有的操作方式
- ❧ 例如：Swing Swing Touch

手機遊戲(音樂類)



手機遊戲(音樂類)



- ❧ 有些音樂遊戲除了觸控螢幕的操作方式之外，還加入了重力感應的功能
- ❧ 也就是說，玩家必須在畫面顯示出類似「shake」字樣時搖晃手機，以獲得更高的分數
- ❧ 例如：Tap Tap Revenge



手機遊戲(音樂類)

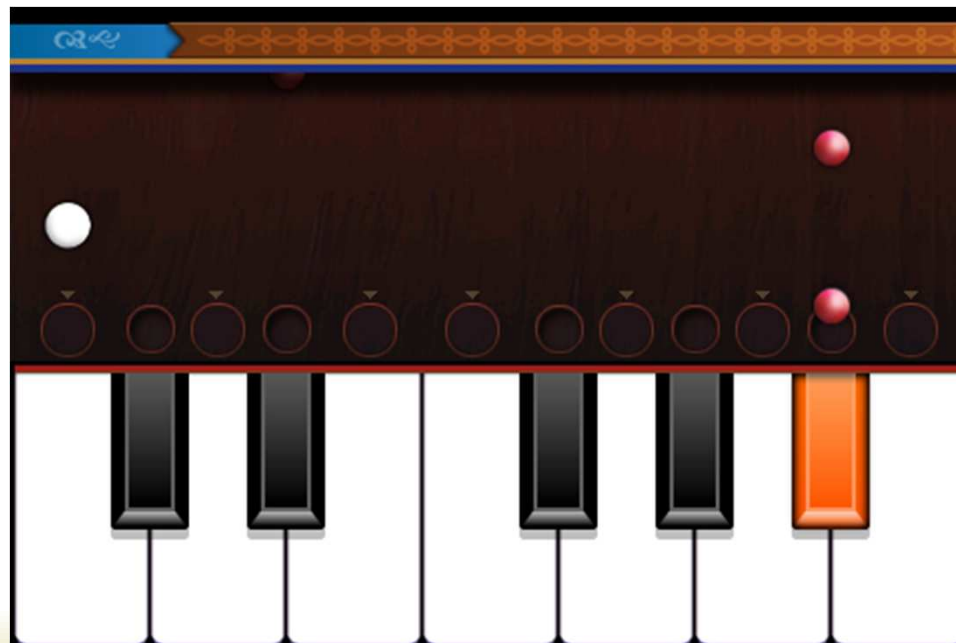


- ❧ 有些音樂類的遊戲則是加入了模擬類的元素，也就是模擬現實生活中真正的樂器演奏的方式
- ❧ 例如，iPhone上大受歡迎的陶笛遊戲(Ocarina)，螢幕上會有四個位置分別模擬陶笛的「孔」，玩家對著麥克風吹氣來模擬吹陶笛，依據真正的陶笛指法按對應的虛擬的「孔」，就可以吹奏出音樂。

手機遊戲(音樂類)



另外還有模擬鋼琴鍵盤的PianoMan，和陶笛遊戲類似，
玩家只要按下畫面上對應的黑白鍵，便可以彈奏出類
似的音樂



手機遊戲(音樂類)



- ❧ 音樂類的遊戲，在最近引起了一陣熱潮。原因無他，正是目前發展出的音樂類的遊戲已經融合了模擬的元素在裡面，使得玩家不必使用真正的樂器，也可以進行表演
- ❧ 例如澳洲有一個團體iBand，他們是一個樂團，只是他們使用的樂器是—iPhone!
- ❧ <http://www.youtube.com/watch?v=TwbnDDA7gyY>

手機遊戲(格鬥類)



- ❧ 對於格鬥類的遊戲而言，傳統的搖桿操作方式目前仍不適用於手機上(因為按鍵繁多)
- ❧ 如果要在手機上開發格鬥類遊戲，勢必要降低對操控的需求
- ❧ 開發快打旋風(Street Fighter)的CAPCOM公司，有試著將快打旋風移植到手機上
- ❧ 搖桿的操作方式，則是使用「虛擬搖桿」的方式

手機遊戲(格鬥類)



- ❧ 所謂的虛擬搖桿，其功能和虛擬十字鍵盤類似，不同的地方在於，虛擬十字鍵盤只需要點擊上下左右四個位置，而虛擬搖桿則須要手指在指定位置進行「滑動」的動作來模擬搖桿操作的方式
- ❧ 儘管搖桿的操作可用虛擬搖桿代替，但其操作觸感仍與真正的搖桿有差距(基本是沒有觸感)，要上手仍需一段適應期
- ❧ 所以甚至有公司推出了實體搖桿for iPad

Fling



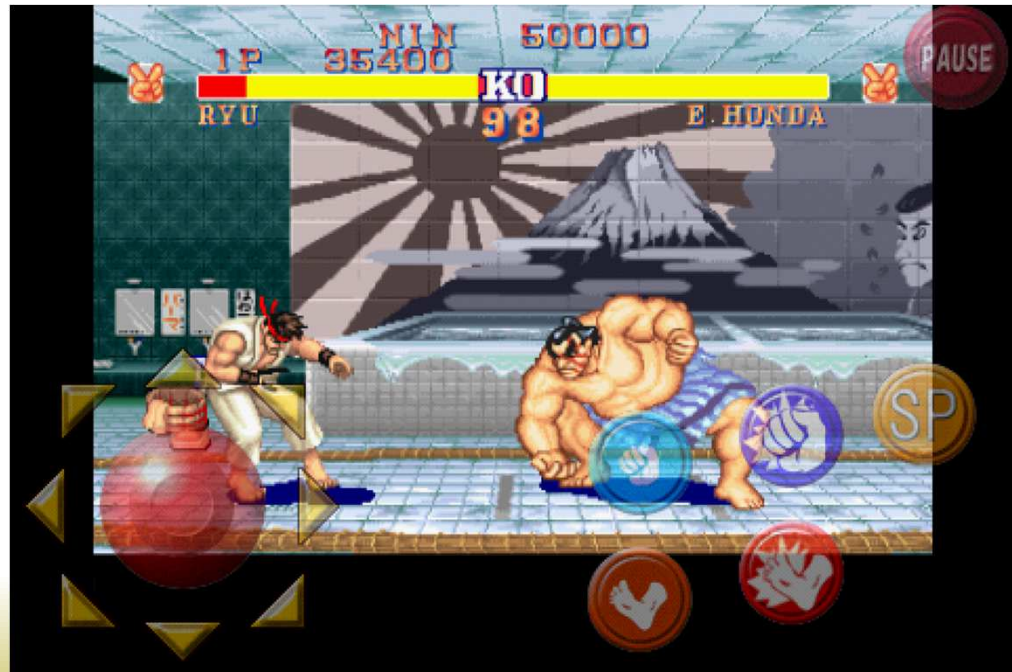
⌘ http://www.youtube.com/watch?v=1JuUG_DP00A&feature=player_embedded



手機遊戲(格鬥類)



- ✧ 受限於手機的畫面，手機版本的快打旋風跟傳統的稍有差別
- ✧ 除了簡化按鍵操作外，在遊戲的難度上也稍有調降



手機遊戲(FPS類)



- ❧ FPS類的遊戲，其操作方法較複雜，除了前後左右移動之外，還必須要能模擬人上下左右擺動的視角以及瞄準敵人發射子彈等功能
- ❧ 如此多的功能，若需全部以鍵盤按鍵來對應，勢必會需要相當多的按鍵，因此FPS類遊戲在早期手機上要發展相當困難

手機遊戲(FPS類)



- ❧ 但有了觸控式螢幕後，這樣的問題便減少了許多，一般手機上的FPS遊戲操作方式如下
 - ❧ 上下左右移動：利用虛擬十字鍵盤/搖桿
 - ❧ 上下左右轉動：用手指在手機的上下左右四邊滑移
 - ❧ 瞄準射擊：有兩種方式
 - ❧ 直接點擊畫面上的敵人
 - ❧ 使用虛擬搖桿

手機遊戲(FPS類)



和格鬥類遊戲相同的問題，由於FPS類的遊戲通常都比較緊湊，對於不熟悉操作的玩家難免會有手忙腳亂，難以上手的問題



手機遊戲(FPS類)



☞ 有些則是把遊玩方式簡化，例如移動方向只剩下往左或往右，虛擬搖桿負責瞄準與射擊

☞ Frontline Commando

<http://www.youtube.com/watch?v=C6lFxVaLlvg>

手機遊戲(益智類)



- ❧ 解謎類的遊戲，隨著手機硬體的發展，在操作的改變上並不大，最大的改變應該是「操作的便利性」
- ❧ 許多電腦上的解謎類的遊戲，通常都是以滑鼠進行操作的方式(例如: Bejeweled Blitz)
- ❧ 但在早期手機上發展時，由於無法使用滑鼠作為操作的方式，以Bejeweled Blitz來說，是改用「上下左右移動到指定位置」的方式操控

手機遊戲(益智類)



- ❧ 但利用觸控式螢幕進行遊戲的時候，這個問題便被解決了，玩家只需要利用手指觸碰螢幕的位置即可取代滑鼠「點擊」的動作
- ❧ 利用手指在螢幕上滑動即可取代滑鼠「拖曳」的動作，例如Unblock me或是Toobs這類型遊戲，很容易就可以從PC端移植到手機上

手機遊戲(益智類)



手機遊戲(益智類)



- 利用觸控螢幕的操作方式，讓這類遊戲的操作性變的相當便利，遊戲的難易度也稍微降低
- 因此，為了增加遊戲難度，這類型的遊戲可加入「限時」或是「限次」的概念，玩家必須在一定時間或是移動次數內儘可能的得到高分
- 這樣的遊戲方式，可增加遊玩時的刺激感與挑戰性

手機遊戲(益智類)



舊版與新版本Bejeweled Blitz的比較



手機遊戲(策略類)



- ❧ 傳統的策略類遊戲，大多是用滑鼠進行操作
- ❧ 因此，再手機上開發時，可同樣利用上述解謎類遊戲的方式來進行操作上的設定
- ❧ 同樣的，礙於手機畫面大小的限制，按鈕或是操作的介面仍以簡化且不妨礙到遊戲進行為原則

手機遊戲(策略類)



- 例如，Popcap公司著名的遊戲Plants vs. Zombies(植物大戰僵尸)，遊戲的內容在電腦版和手機版幾乎是完全相同的，主要改變的地方就是操作介面的佈置方式



PC版本



手機版本

手機遊戲(RTS類)



- ❧ RTS類的遊戲目前在手機上開發不若其他類型遊戲來的多，主要原因如下：
- ❧ 連線需求
 - ❧ 雖然RTS類遊戲仍有單人遊玩模式，但這類型遊戲的主要娛樂來源仍然是線上玩家對戰的方式
 - ❧ 雖然目前手機有提供網路連線，但伺服器等資源仍需開發廠商提供，若是玩家人數不足，就沒有維護的價值
- ❧ 操作介面
 - ❧ 雖然RTS類遊戲可以用滑鼠控制大部分的動作，但是RTS可算是一個以速度為主的遊戲，
 - ❧ 為了增加單位時間執行指令的次數，RTS遊戲往往會搭配大量的熱鍵，光是這一點就讓RTS類遊戲難以在手機上開發(因為手機無法使用大量的熱鍵)

手機遊戲(運動類)



- ❧ 由於運動類強調的是擬真性，擬真性分為「畫面上的擬真」跟「物理運算上的擬真」
- ❧ 「畫面上的擬真」對於硬體上的要求較高，因此早期手機比較注重「物理運算上的擬真」
- ❧ 或是有些遊戲直接反其道而行，以誇張的方式呈現運動類遊戲，來製造另一種的娛樂性

手機遊戲(運動類)



- ✧ 近年由於手機硬體的進步，運動類遊戲也慢慢的開始強調畫面性的擬真。
- ✧ 然而，一些較為擬真的運動類遊戲，往往都需要相當多的按鍵設計。
- ✧ 因此，在操作介面的設計上，仍需以簡單化為原則。

手機遊戲(運動類)



- ❧ 例如，移植至手機平台上的籃球遊戲NBA Live 10，便把遊戲操作簡化為：虛擬十字鍵盤(搖桿) + 兩個虛擬按鍵
- ❧ 雖然按鍵數目較少，但使用觸控式螢幕的好處在於可以利用「按壓的時間」來產生更多的按鍵組合，例如：
- ❧ 進攻
 - ❧ A鍵：投籃 / 長按A鍵：灌籃
 - ❧ B鍵：傳球/長按B鍵：指定傳球的人
- ❧ 防守
 - ❧ A鍵：跳躍
 - ❧ B鍵：抄球



手機遊戲(運動類)



- ❧ 有些以誇張的動作為主的運動遊戲(例如NBA JAM) , 其實也是相當受歡迎的。
- ❧ 這類遊戲因為操作簡單的關係, 所需求的按鍵並不多, 反而更適合在手機上開發。

手機遊戲(運動類)



手機遊戲(運動類)



- ☞ 有些運動類的遊戲則是捨棄虛擬搖桿的操作方式，改用「三軸重力感應器」作為操作的方式
- ☞ 例如：Homerun Battle這款遊戲便是利用重力感應器來移動球棒的位置，觸碰螢幕的任何位置則是「揮棒」的動作



手機遊戲(競速類)



- 而競速類的遊戲也可利用三軸感應器裝置，玩家可藉由左右旋轉手機來模擬方向盤的旋轉動作，製造出較為擬真的操作方式



手機遊戲(角色扮演類)



- ❧ 角色扮演類的遊戲，其操作方式也是較為簡單的類型，因此不管是移植或是新開發，其作品也是相當的多
- ❧ 操作的方式也是用虛擬十字鍵盤的方式搭配一到兩個虛擬按鍵即可
- ❧ 例如：太空戰士系列，便是由家用主機移植到iphone上的例子
- ❧ 而Zenonia則是在iphone上新開發的角色扮演遊戲
- ❧ Dungeon Hunter更是在iphone上大紅大紫後轉回家用主機上發行的遊戲

手機遊戲(角色扮演類)



Final Fantasy II 與 Zenonia 2 的遊戲畫面



手機遊戲(模擬類)



- ❧ 模擬類的遊戲，近年來也是大量的在手機上開發，原因無他，正是因為操控的方式比傳統手機更為直覺與方便
- ❧ 例如載具模擬類的遊戲(如競速類)，常常是利用「三軸重力感應器」來模擬方向盤的左轉或右轉，且用兩個虛擬按鍵來代表油門與煞車

手機遊戲(模擬類)



- ❧ 經營型或養成型的模擬類遊戲，在手機上開發更是不勝枚舉
- ❧ 這兩類的遊戲原本在電腦上的操作就有很大一部分是以滑鼠為主，因此無論是單純的移植或是開發新的構想，都可以原本滑鼠的操作介面為基礎進行開發

手機遊戲發展



- ❧ 目前的手機遊戲開發方向有三大分類
- ❧ 全新開發
- ❧ 平台移植
- ❧ 舊款HD

手機遊戲發展



- ❧ PC/PS3/XBOX360端耳熟能詳的遊戲巨作
- ❧ FIFA 12
- ❧ NBA 2k12
- ❧ Need for speed
- ❧ Resident evil
- ❧ Final Fantasy II & III

手機遊戲發展



❧ 波斯王子

❧ PC版

http://www.youtube.com/watch?v=_zyQ_OVyhNE

❧ 手機版

<http://www.youtube.com/watch?v=DiuSpXmhfKo>

手機遊戲發展



- ❧ 但是，移植的遊戲許多是屬於「叫好不叫座」，其中一個原因是「操作介面的改變」
- ❧ 由於移植遊戲的開發基礎是建構於原本其他平台的遊戲之上，因此在遊戲介面的設計上很難跳脫原本遊戲介面的框架

手機遊戲發展



- ❧ 以惡靈古堡遊戲為例，原本PC上的操作介面是以滑鼠為主，在iOS上就變成了虛擬搖桿的方式
- ❧ 這樣的操作方式在FPS類型的遊戲操控上並不是非常的方便



手機遊戲發展



- ❧ 雖然手機的硬體仍然不斷的在進步，但比起家用主機硬體發展的速度來說，仍然是存在著不小的差距
- ❧ 一些目前在PC上大為風行的遊戲，要直接移植到手機上有著一定的難度。這一點，可由目前手機遊戲市場上大多是五六年前的移植產品窺知一二
- ❧ 因此，針對手機的特性與需求開發新類型的遊戲是勢在必行

手機遊戲發展



- ✧ 近年來由於3G技術的崛起，透過手機連接網路並搭配各式各樣的服務，也因應而生
- ✧ 順應這樣的潮流，一些結合網路功能或特色的遊戲也開始流行於手機上
- ✧ 這些搭配網路功能的遊戲的設計方式，則是類似於PC上的「網頁遊戲」的設計概念

手機遊戲發展



- ❧ 所謂類似網頁遊戲的設計概念，並不是指在手機上直接用瀏覽器打開網頁進行遊戲，而是參考了這類遊戲有別於傳統遊戲的優勢。
- ❧ 這個優勢就是所謂的「移動性(Mobility)」。
- ❧ 網頁遊戲最大的好處就在於玩家可以在任何時間與地點，只要有一台能上網的電腦就可以進行遊戲，不再需要帶著主機跑。

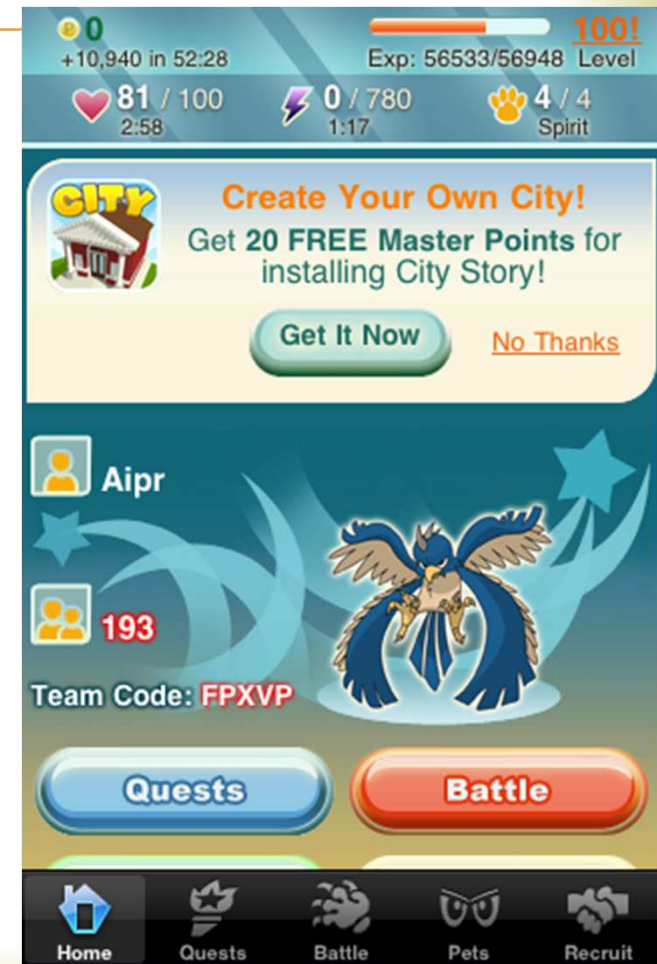
手機遊戲發展



- 而網頁遊戲的特色：「硬體需求較低」以及「高移動性」也正是手機遊戲的特色
- 因此，這類型的遊戲在手機上開發是一個必然的趨勢。
- 和網頁遊戲不同的是，手機本身就是一個高移動性裝置，因此，在開發這類型的遊戲時，不需要透過瀏覽器的安裝，即可滿足高移動性的需求。
- 也就是說，在開發上，我們直接讓手機的作業系統執行JAVA或Flash的程式，而非透過瀏覽器的掛載方式。

結合網路功能的手機遊戲

- 例如，Pets Live這是一個養寵物的文字型遊戲，玩家可以讓此虛擬寵物進行一些活動，這些活動會耗費寵物的精力，當寵物精力降成0時就無法繼續行動，必須等待精力的回覆
- 傳統的遊戲可能就有「睡眠」這項指令讓寵物回復精力，而在結合網路功能的手機遊戲上，在這個回復的過程就必須等待現實生活中的一段時間才行。



結合視訊鏡頭的手機遊戲



- ❧ 除了多點觸控的功能之外，手機還有另一項優勢，就是視訊鏡頭
- ❧ 雖然現在的筆電與螢幕大多已整合了視訊鏡頭，但幾乎沒有廠商利用此一設備開發遊戲

結合視訊鏡頭的手機遊戲



- ❧ 擴增實境(augmented reality)的一種延伸應用
 - ❧ 把虛擬世界與現實世界的畫面進行整合與互動
- ❧ 擴增實境的概念大概在1990年左右被提出，但由於影像處理技術需要耗費大量的運算，因此在硬體技術提升後的今日才慢慢普及

結合視訊鏡頭的手機遊戲



❧ Gunman

❧ AR basketball

❧ ARf

手機遊戲內容定位



- ❧ 早期在PC上暢銷的遊戲，很多都移植到手機平台上
 - ❧ NBA Live、Need for Speed、Red Alert
- ❧ 這些遊戲無論是在聲光效果或是遊戲性上都是一等一的好
- ❧ 但這些遊戲是手機上賣得最好的遊戲嗎？

手機遊戲內容定位



- ❧ 那些手機遊戲是大熱門?
 - ❧ Angry Birds
<http://www.youtube.com/watch?v=bNNzRyd1xz0>
 - ❧ Where's my water
<http://www.youtube.com/watch?v=UnFFWnw0ohk>
 - ❧ Tiny Wings
<http://www.youtube.com/watch?v=VUn1E4cGgz0>
- ❧ 全球下載次數超過兩億次、平均全世界每個人要花兩個小時玩它
- ❧ Angry birds沒有華麗的畫面、震撼的音效，為什麼會讓全世界的人為之瘋狂？
- ❧ 理由很簡單：因為它夠簡單！

手機遊戲內容定位



- ❧ 遊戲玩家可概略分成兩種
 - ❧ 輕度玩家(Light users)：偶爾才接觸遊戲的人
 - ❧ 核心玩家(Core users)：玩遊戲的年齡較長、對遊戲的聲光效果較為追求者
- ❧ 輕度玩家才是手機遊戲的主戰場！
- ❧ 許多人玩手機遊戲只是為了打發時間(排隊、等車)等等，遊戲自然不適合做得過於複雜
- ❧ 界面要簡單，玩起來輕鬆

手機遊戲未來發展



- ❧ 隨著手機的功能越來越強大，手機遊戲的未來也是不可限量的
- ❧ 手機遊戲的未來開發方向可分為「單機型」與「網路型」兩種
- ❧ 單機型：以活用手機硬體的特色進行開發
- ❧ 網路型：結合網路連線或是社群功能，以增加人與人互動的娛樂方式

手機遊戲未來發展



- ❧ 手機遊戲朝向整合社群功能的方向發展，是一個必然的趨勢
- ❧ 例如：棋藝類的對戰遊戲，在設計上較為簡單，操作介面也很容易可以利用觸控面板達成，這類的遊戲就可以利用手機上網的方式和其他玩家進行對戰
- ❧ 像是PC上的明星3缺1這類的麻將遊戲，或是撲克牌遊戲，都可以轉移到手機上
- ❧ 這類遊戲每一局遊玩的時間很短暫，玩家不需要長時間進行遊戲，非常適合通勤搭車的時候遊玩

手機遊戲未來發展

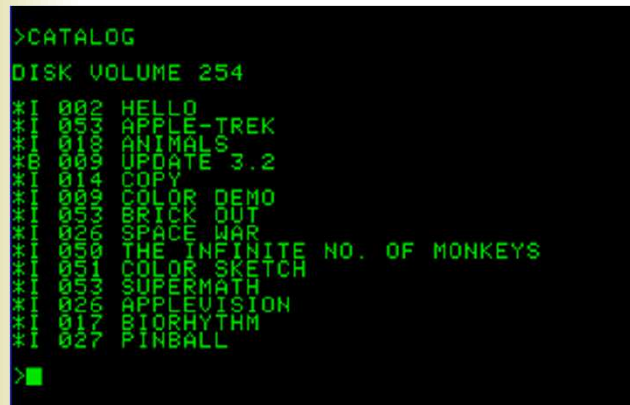


- ❧ 另一種最直接的方法，就是直接移植目前網路社群網站上的遊戲
- ❧ 這些社群網站上的遊戲幾乎都是以flash或是java的方式發展的，這也讓移植的過程方便些
- ❧ 目前許多facebook上的遊戲都有手機上的版本，有些甚至可以直接匯入facebook上的好友名單，直接跟facebook的網路社群整合

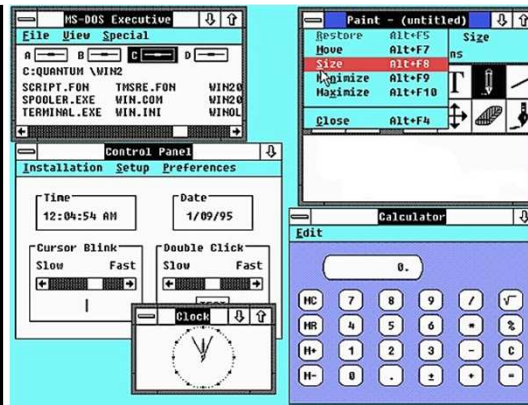
手機遊戲未來發展



觀看手機的發展趨勢，其實和PC是很相似的



文字介面



圖像介面



觸控介面

那麼，手機接下來的演變趨勢會是什麼呢？

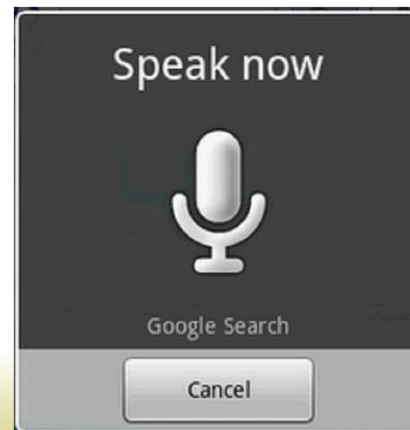
手機遊戲未來發展



Sound is going to be bigger than video...

'Record' is the new QWERTY.

- Alexander Ljung, Founder & CEO, SoundCloud



手機遊戲未來發展



- ❧ 單機型的手機遊戲，其開發目標可朝著手機硬體的特色來發展
- ❧ 除了既有的觸控螢幕、三軸感應器外，也可使用「聲控」的方式進行遊戲
- ❧ 目前語音辨認的技術已漸趨成熟，以後的遊戲走向，也可設計成利用「聲音」來取代畫面上的指令
- ❧ 例如虛擬寵物，也許可以利用講話的方式，與虛擬寵物進行互動，甚至可以設計成寵物會認主人的聲音，如此一來，是不是又更加的真實了呢？

手機遊戲未來發展



- ❧ 單機型的手機遊戲，未來可跟著手機硬體的發展潮流來走
- ❧ 例如，最近非常流行的「3D」成像技術，未來非常有機會在手機上普及化
- ❧ 由於人的雙眼本來就提供立體的視覺效果，3D的顯示方式自然會被認為較接近人眼的視覺，因此也被視為螢幕顯示技術的必然發展方向

3D成像技術



- ❧ 3D技術的應用，其實蠻早就開始研究與發展，而是電影「阿凡達」的正式來臨，更宣告了3D時代的正式來臨。
- ❧ 3D電影技術幾已成熟，遊戲邁入3D化也是已經是一個平面，要在平面上讓人眼產生立體視覺，勢必要用到一些技巧。
- ❧ 這些技巧可分為兩類
 - ❧ Pseudo-3D圖像技術(俗稱2.5D)
 - ❧ 實際3D圖像技術

模擬3D技術



- ❧ 模擬3D(2.5D)利用的是一些線條之間的相互關係來讓人眼產生錯覺，模擬出人類眼睛所看到真正3D景物的情形，而產生有遠近/立體的感覺



實際3D



- ❧ 實際3D圖像技術是在螢幕上顯示雙重影像，再讓使用者戴上偏光眼鏡，讓兩眼分別濾掉其中一個影像，來達成立體視覺的效果。
- ❧ 然而，這種偏光眼鏡對原本就有戴眼鏡的人來說並不好用。現在3D眼鏡的應用大多是在電影等短時間觀賞的需求上，因此還勉強可以接受
- ❧ 有越來越多的研究報告指出，長時間配戴這種偏光眼鏡會導致頭暈、嘔心等現象。
- ❧ 因此，「裸視3D」是一個必然發展的技術走向

手機上的3D技術



- ☞ 如果讓3D顯示技術發展至手機上，必須了解到使用者在使用手機上的一些特性，敘述如下：
1. 大部分時間為單人使用
 2. 觀看距離差異不大，一般而言約25-30cm
 3. 觀看角度變化不大
 4. 盡量不要要求配戴3D眼鏡(因為並非所有應用都會用到3D，可能會需要2D/3D顯示的轉換)

手機上的3D技術

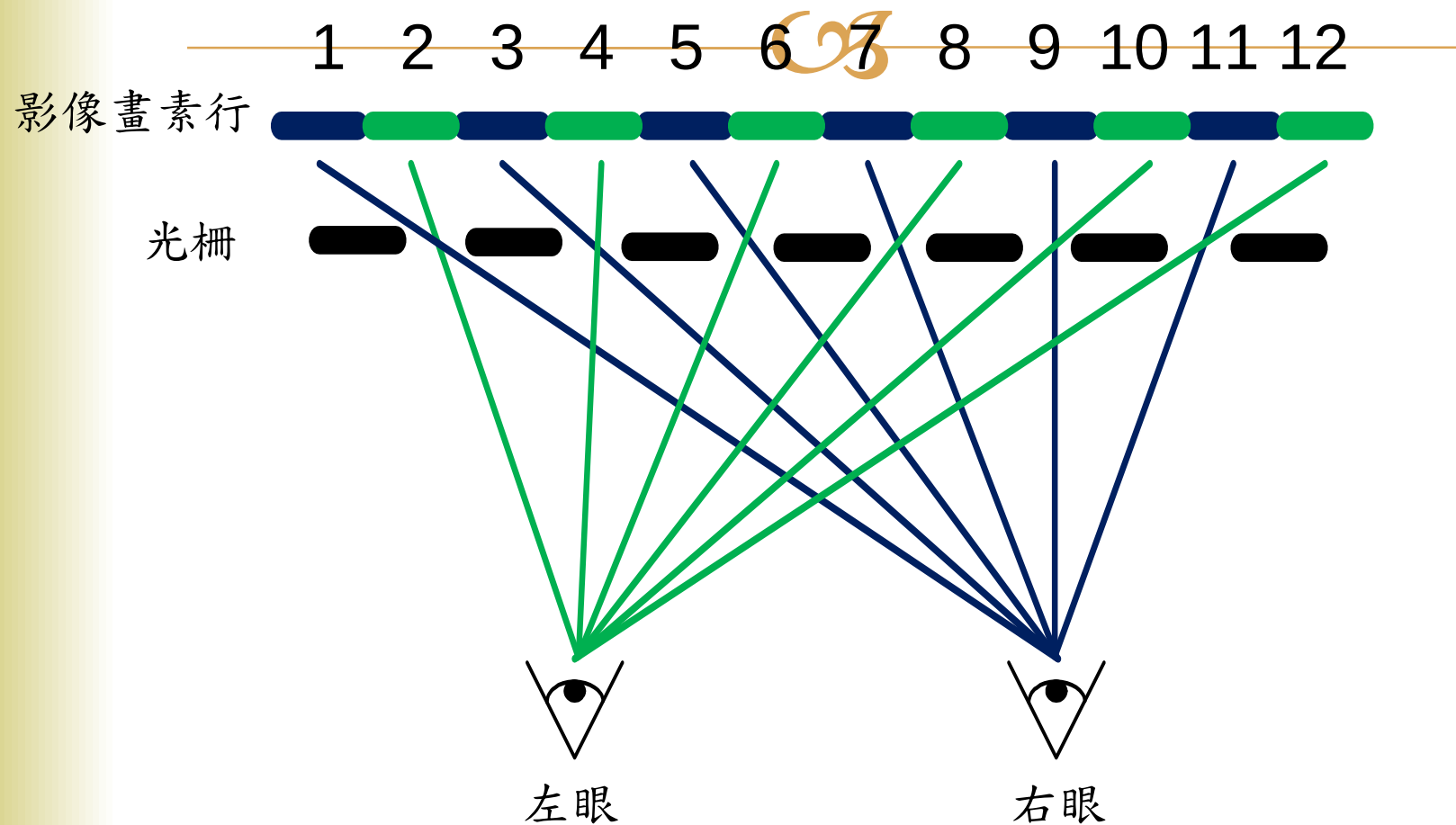


- ❧ 根據以上之需求，目前許多手機大廠例如Samsung，所開發之3D手機技術採用的是視差遮障法（ Parrallax Barrier）技術
- ❧ Parrallax Barrier技術是將取得的兩個不同角度的影像，分別切割成數條，並以交錯(interlace)的方式加以融合，再透過螢幕前方 一條一條的屏障，讓觀看者的左眼和右眼所獲得的影像不同，以落差的方式製造出3D效果。

Parallax Barrier

此一技術好像是讓螢幕戴上3D眼鏡一樣，其運作原理就是黑色(barrier)和透明(slits)垂直條紋間隔排列成的圖案，使得背光板的光透過該圖案之後，到達左眼的光線只經過奇數行的畫素，到達右眼的光線則只經過偶數行的畫素。藉由左右眼影像分別用奇數行和偶數行交錯顯示，就可以讓人不需佩帶額外眼鏡就看到立體影像了。

Parrallax Barrier



Parrallax Barrier



- ❧ 此屏障式技術的缺點在於使用者必需在特定的視角（正向）和視距才能產生最好的3D體驗效果
- ❧ 此外，屏障式技術會造成影像的解析度減半（實際上公式應為「解析度/可用視點」），若需求的視點越多，解析度會越低，這在大尺寸電視或大型電子看板上要普及變成是一個很難克服的問題

Parrallax Barrier



- ❧ 但是，這個問題對於小型的移動式電子裝置來說，並不是那麼的致命
- ❧ 因為手機並不需要很大的視角（一般人使用大多是正面），加上手機螢幕並不大（表示解析度並不需要太高）

3D遊戲



- ✧ 任天堂公司在2010年宣佈旗下的產品 Nintendo 3DS 是一台不需要搭配3D眼鏡的3D掌上型遊樂器
- ✧ 既然這項技術已經在掌上型遊樂器上使用，相信在手機上發展也是指日可待的
- ✧ 但是，由於手機並非專為遊戲而製造，因此這項技術要發展到手機面板，還必須搭配2D/3D切換的技術

3D手機



- ❧ 韓國的大廠三星(SAMSUNG)之前也發表了裸視3D手機，解析度為240x400，但屬於封閉式的手機，並無智慧型手機的功能，整體來說應屬於測試3D手機市場水溫為主
- ❧ 而另一家廠商LG也在2011年發表了3D智慧型手機，搭配的是Android 2.3系統
- ❧ 宏達電(HTC)也有推出3D手機 EVO 3D，但僅限於拍照與錄影功能
- ❧ 日本的廠商SoftBank也推出了3D智慧型手機，甚至也有遊戲上市，例如惡靈古堡、實況野球、洛克人等3D版本。

3D手機遊戲發展？



- ❧ 3D顯示技術目前已經成型，但手機的螢幕顯示也能3D化，最大的關鍵應該不是在硬體，而是在軟體內容上的發展
- ❧ 好的硬體仍然需要好的軟體與其搭配，才能顯示出它的價值
- ❧ 目前投入3D手機遊戲發展的廠商仍然偏少，因為要把遊戲做成 3D，開發成本必然會增加，若將成本反映到價格上，遊戲銷量可能不增反減
- ❧ 只要投入廠商增加，3D手機遊戲必定可成為手機遊戲史上的一個重大突破

手機遊戲未來發展



- ❧ 觸控式螢幕讓手機遊戲的新鮮感與玩法大增，但是手的移動與操作的配置，都會受到螢幕的大小所侷限
- ❧ 此外，觸控式螢幕畢竟還是個平面的東西，手指一但離開了螢幕，就無法進行操作了
- ❧ 既然我們讓畫面顯示方式3D化了，我們能不能也讓「操作方式」3D化呢？

手機遊戲未來發展



- ☞ 要讓操作方式3D化，我們可以參考的方式就是微軟在Xbox上進行的體感控制器：Kinect
- ☞ Kinect有一個很大的特色就是「玩家本身並不需要任何的控制器」就可進行操作
- ☞ 這其中運作的大關鍵就是「鏡頭」

手機遊戲未來發展



☞ Kinect首先是Kinect機身上有3顆鏡頭，其中一個鏡頭是一般的彩色攝影機，另外兩個鏡頭則分別為紅外線發射器和紅外線CMOS攝影機所構成的3D深度感應器，可以擷取出具有3D深度的影像，Kinect就是靠這張具有3D深度的影像來偵測玩家的動作

手機遊戲未來發展



- ❧ 這個3D成像技術，微軟是與PrimSence這家公司合作所產生的
- ❧ 由於微軟並沒有買斷PrimeSense這家公司的技術，因此這種紅外線裝置未來也可能出現在手機上
- ❧ 只要在手機螢幕的周圍佈上這種紅外線裝置，再搭配軟體開發，要偵測出手離畫面的遠近是可能的

手機遊戲未來發展



- ❧ 當操作方式由2D轉成3D時，未來的手機遊戲，也許可加入「拉」跟「推」的動作，再搭配3D顯示的技術，遊戲性必定大幅提高
- ❧ 例如「拔蘿蔔」的遊戲，也許以後你就會看到蘿蔔就像真的被你給「拔」出畫面也說不定！

結論



- ❧ 手機遊戲獲利 < 傳統遊戲
- ❧ 手機遊戲開支 < 傳統遊戲
- ❧ 小蝦米不一定輸給大鯨魚
- ❧ 創意 > 技術

簡易手機遊戲設計



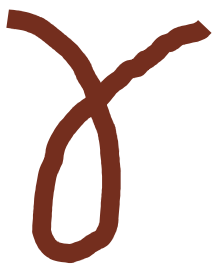
✎ 模擬情境：在RPG遊戲裡，你是一個魔法師，你會使用火，冰，電，毒四種魔法，你要如何使用不同的魔法打敗敵人？

基本設計方式

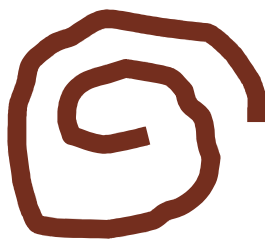


- ❧ 傳統設計方式：在畫面邊緣設計不同魔法的按鈕，按下按鈕進行攻擊
- ❧ 應用手機特性設計方式：藉由在螢幕上不同的滑動軌跡，來表示使用不同的魔法，例如

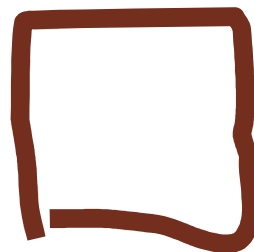
fire



poison



ice



thunder



流程架構



軌跡輸入



軌跡分類



顯示魔法

實作遊戲



- ❧ 在軌跡製作的部分，可利用Google提供的Gesture builder應用程式
- ❧ 安裝完ADT後的模擬器內即有內建此APP
- ❧ 如果想要研究程式內容的話，可利用以下兩種方式開啟專案：
- ❧ File->new->project->Android Project->Create a project from existing source->gesture builder
- ❧ File->new->project->Android sample project->選擇專案執行版本->gesture builder

Gesture Builder執行畫面



- ✎ 建立完成後可直接在project name的部分按下右鍵->run as->android application，執行後可建立各種自定義的Gesture



Gesture存放位置



- 建立完數個gesture後，會存成一個檔案(檔名可自訂)
並存放在模擬裝置的sdcard資料夾內(利用DDMS觀看)

DDMS - g_test/src/tw/vnu/g_test.java - Eclipse

File Edit Run Source Navigate Search Project Refactor Window Help

Devices

Name	Size	Date	Time	Permissions	Info
emulator-555 Online					
system_pr...	60	8600			
jp.co.omr...	119	8601			
com.andr...	123	8602			
com.andr...	126	8603			
com.andr...	129	8604			
android.p...	168	8655			
com.andr...	174	8663			
com.andr...	194	8667			
android.p...	202	8671			

Threads Heap Allocation Tracker File Explorer

Name	Size	Date	Time	Permissions	Info
data		2010-11-09	08:03	drwxrwx--x	
mnt		2010-11-09	12:40	drwxrwxr-x	
asec		2010-11-09	12:40	drwxr-xr-x	
sdcard		1970-01-01	00:00	d---rwxr-x	
LOST.DIR		2010-11-09	08:04	d---rwxr-x	
gestures	1984	2010-11-09	12:16	----rwxr-x	
secure		2010-11-09	12:40	drwx-----	
system		2010-06-30	21:06	drwxr-xr-x	

Emulator Control

Telephony Status

Voice: home Speed: Full

Data: home Latency:

實作遊戲



- ✧ 建好Gesture檔案後，先用DDMS把這個檔案抓出來放到電腦上，之後在開發的遊戲中加入這個檔案

實作遊戲



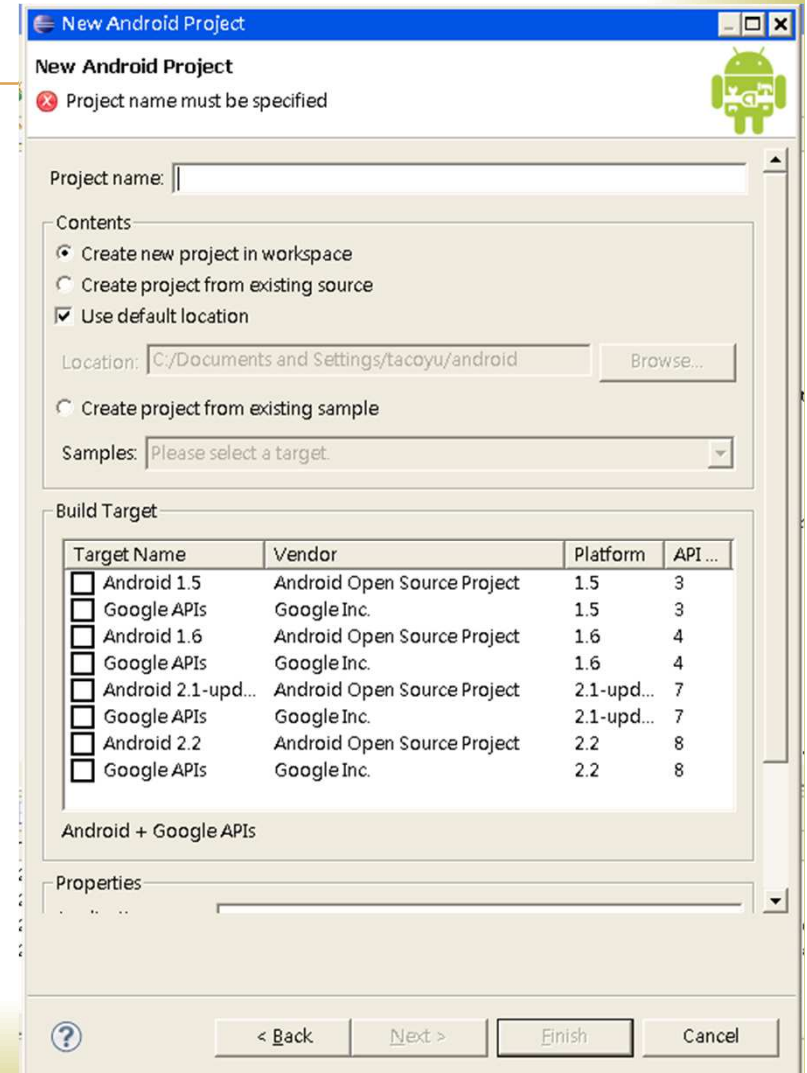
建立專案方式如下：

File->new->project->Android Project

Project name: 任意

Contents: 選擇 Create new project

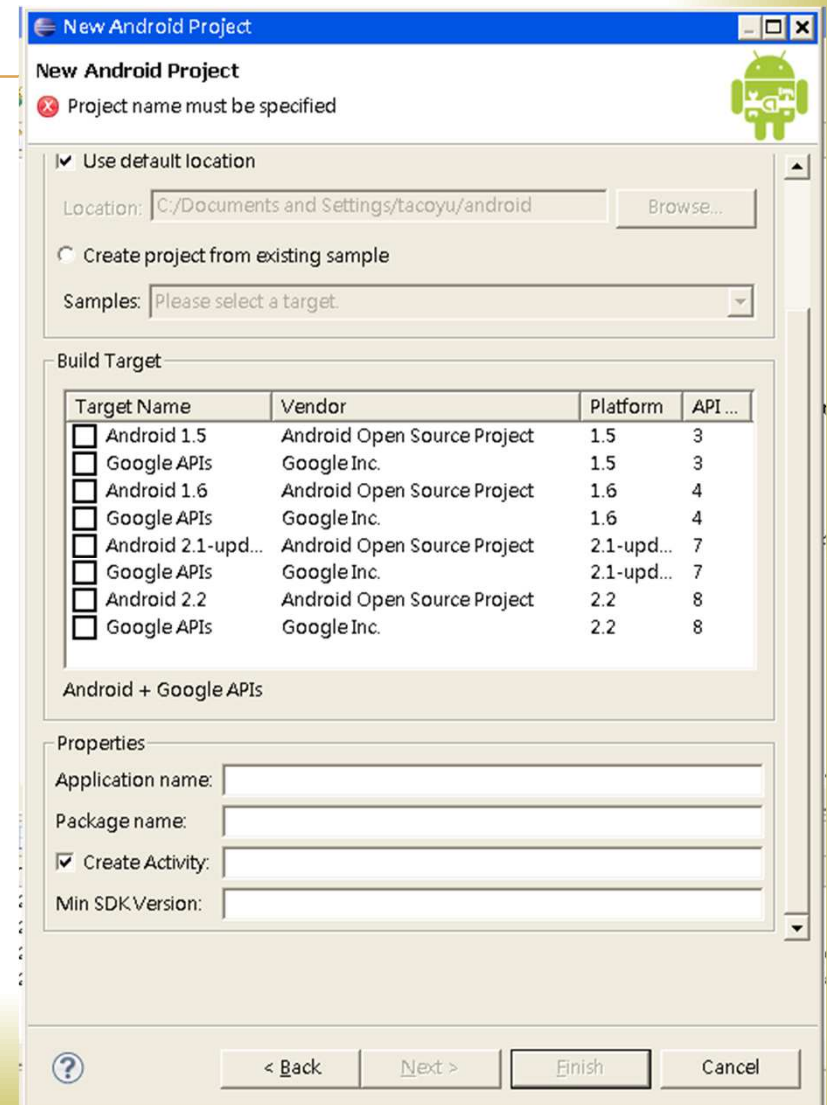
Build Target: 可看手機作業系統等級決定(純PC端模擬可直接選最高等級的)



設定專案相關屬性

Properties

- Application name: 隨意
- Package name: 必須是一個 java 的 namespace，這邊簡單使用 `tw.vnu` 作為名稱
- Activity: 隨意，主要是要撰寫的 java 檔案的檔名
- Min SDK version: 可空白



New Android Project

Project name must be specified

☒ Use default location

Location:

☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API ...
☐ Android 1.5	Android Open Source Project	1.5	3
☐ Google APIs	Google Inc.	1.5	3
☐ Android 1.6	Android Open Source Project	1.6	4
☐ Google APIs	Google Inc.	1.6	4
☐ Android 2.1-upd...	Android Open Source Project	2.1-upd...	7
☐ Google APIs	Google Inc.	2.1-upd...	7
☐ Android 2.2	Android Open Source Project	2.2	8
☐ Google APIs	Google Inc.	2.2	8

Android + Google APIs

Properties

Application name:

Package name:

☒ Create Activity:

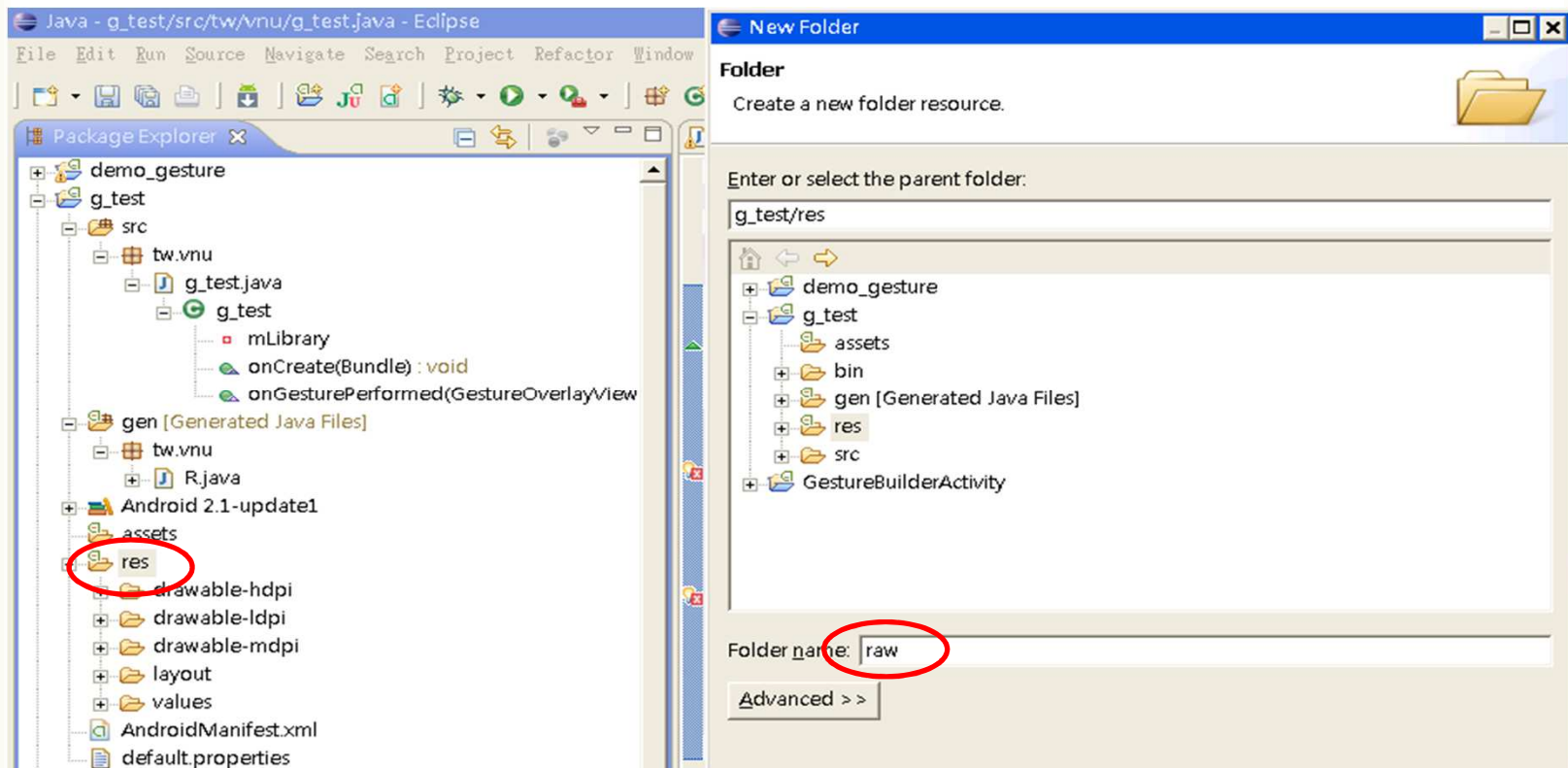
Min SDK Version:

讓程式直接讀取gesture檔案



- ☞ Res資料夾下建立 raw資料夾
- ☞ res圖示上按右鍵->new->folder
- ☞ 把建好的gesture檔案放入新的專案的res->raw資料夾下

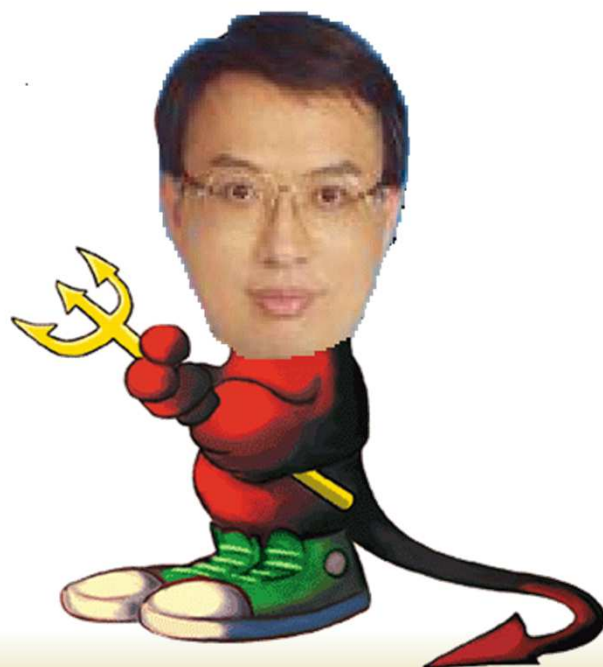
讓程式直接讀取gesture檔案



怪物圖片



☞ 把圖片放在專案的 `res->drawable->hdpi` 下，之後即可用xml或是程式直接讀取



設計Layout



- ☞ Android畫面的Layout是利用xml語法來設計
- ☞ 檔案存放在res->layout資料夾下

設計 xml 檔案



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <android.gesture.GestureOverlayView
        android:id="@+id/gestures"
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:foreground="@drawable/image1"
        android:foregroundGravity="center">

        </android.gesture.GestureOverlayView>
</LinearLayout>
```

讓程式讀取gesture檔案



☞ Android提供GesturePerformListener介面來記錄使用者描繪的軌跡

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    mLibrary = GestureLibraries.fromRawResource(this, R.raw.gestures);  
    if (!mLibrary.load()) {  
        finish();  
    }  
    GestureOverlayView gestures = (GestureOverlayView) findViewById(R.id.gestures);  
    gestures = (GestureOverlayView) findViewById(R.id.gestures);  
    gestures.setForeground(getResources().getDrawable(R.drawable.image1));  
    gestures.addOnGesturePerformedListener(this);  
}
```

程式運行邏輯



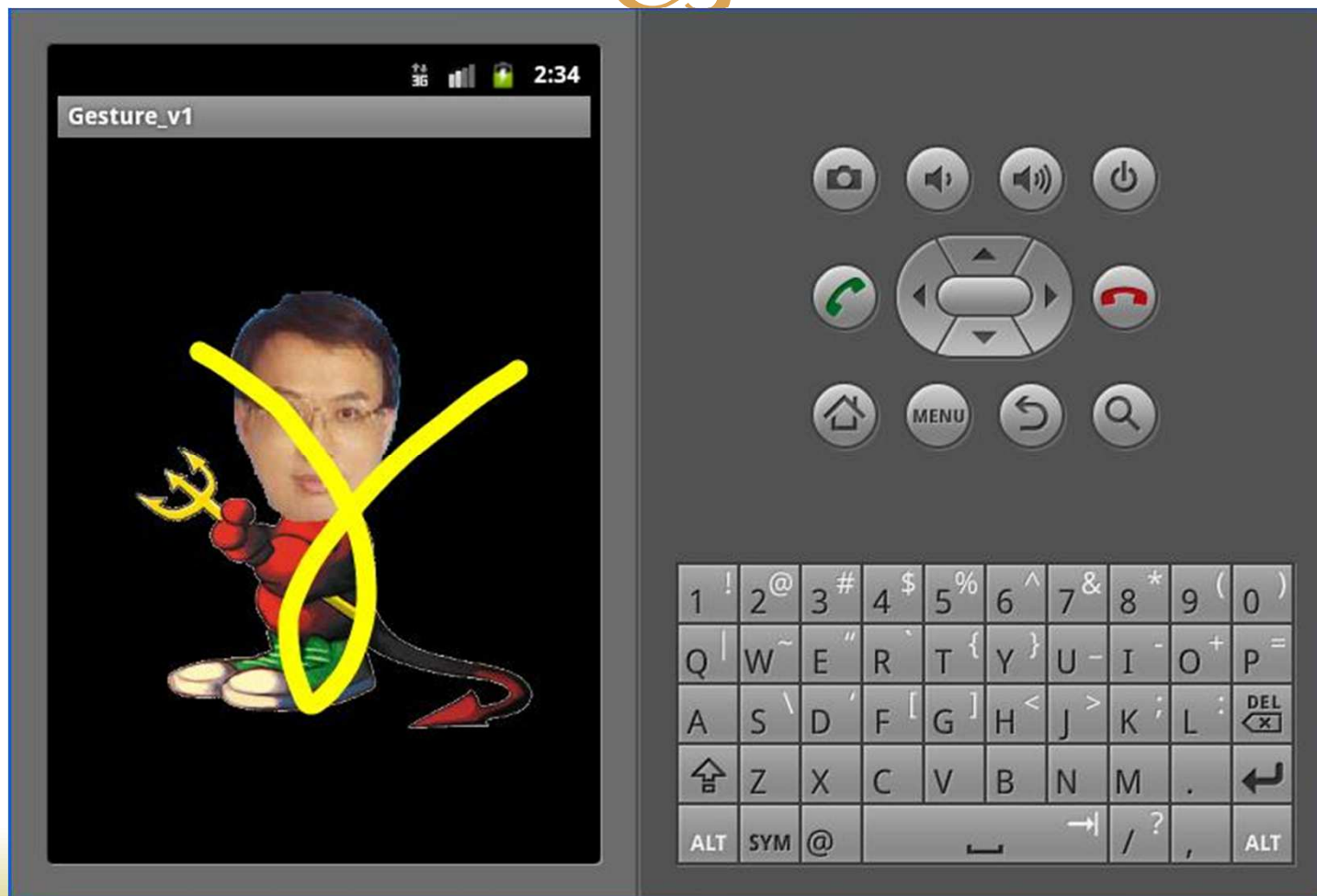
```
gesture = 偵測軌跡();  
spell = 判斷軌跡分類();  
if (spell=fire)  
    顯示("fire");  
else if (spell=poison)  
    顯示("poison");  
else if (spell=ice)  
    顯示("ice");  
else if (spell=thunder)  
    顯示("thunder");  
else  
    顯示("fail");
```

```

public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture) {
    @SuppressWarnings("rawtypes")
    ArrayList predictions = mLibrary.recognize(gesture);
    // We want at least one prediction
    if (predictions.size() > 0) {
        prediction = (Prediction) predictions.get(0);
        // We want at least some confidence in the result
        if (prediction.score > 1.5) {
            // Show the spell
            String s1 = "Fire", s2 = "Poison", s3 = "Ice", s4 = "Thunder";
            if (s1.equals(prediction.name)) {
                Toast.makeText(this, "Fire", Toast.LENGTH_SHORT).show();
            }
            else if (s2.equals(prediction.name)) {
                Toast.makeText(this, "Poison", Toast.LENGTH_SHORT).show();
            }
            else if (s3.equals(prediction.name)) {
                Toast.makeText(this, "Ice", Toast.LENGTH_SHORT).show();
            }
            else if (s4.equals(prediction.name)) {
                Toast.makeText(this, "Thunder", Toast.LENGTH_SHORT).show();
            }
        }
        else {
            Toast.makeText(this, "fail", Toast.LENGTH_SHORT).show();
        }
    }
    GestureOverlayView gestures = (GestureOverlayView) findViewById(R.id.gestures);
    gestures.addOnGesturePerformedListener(this);
}

```

模擬器執行情況



怪物圖片



- 為了凸顯不同魔法施放後的差異，可以稍加改變怪物的顏色來凸顯
- 既然是使用法術，就順便顯示出法術傷害吧！

fire



poison



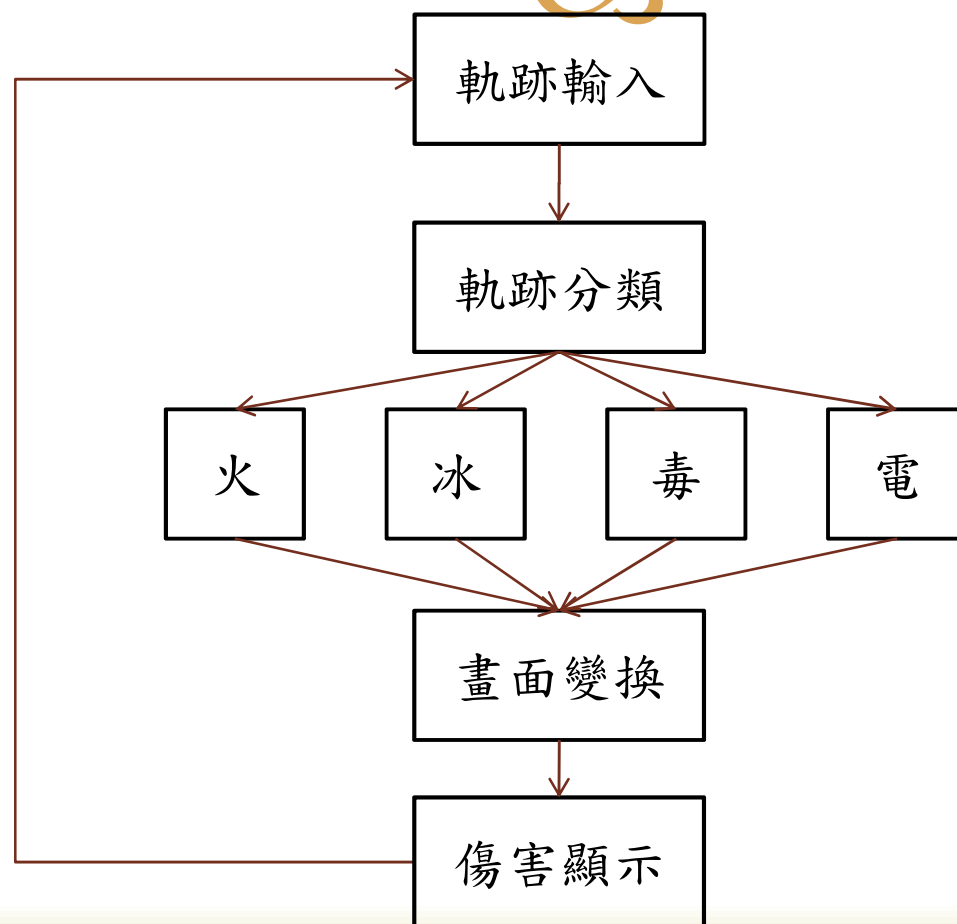
ice



lightning



流程圖



實作遊戲



- ❧ 先把上面四張圖片放進drawable-hdpi資料夾內
- ❧ 遊戲設計上，根據不同魔法的施展，對應到怪物畫面的不同，有兩種設計方式
 - ❧ 利用切換版面設定(Layout)的方式
 - ❧ 利用變更GestureOverlayView的前景圖片
- ❧ 在判斷完玩家所施展的法術(軌跡)後，再根據結果來切換對應的版面/圖片
- ❧ 法術傷害可簡單的使用Math.random()函式

實作遊戲



- 新增4個xml，更換GestureOverlayView的foreground圖片

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <android.gesture.GestureOverlayView
        android:id="@+id/gestures"
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:foreground="@drawable/image_fire"
        android:foregroundGravity="center">

        </android.gesture.GestureOverlayView>
    </LinearLayout>
```

替換成
image_poison
image_ice
image_thunder

```
int dmg = (int) (Math.random()*100);
```

隨機產生傷害值

```
// We want at least one prediction
```

```
if (predictions.size() > 0) {
```

```
    prediction = (Prediction) predictions.get(0);
```

```
    // We want at least some confidence in the result
```

```
    if (prediction.score > 1.5) {
```

```
        // Show the spell
```

```
        String s1 = "Fire", s2 = "Poison", s3 = "Ice", s4 = "Thunder";
```

```
        if (s1.equals(prediction.name)) {
```

```
            setContentView(R.layout.main1);
```

切換對應的Layout

```
            Toast.makeText(this, "Fire damage:" + dmg, Toast.LENGTH_SHORT).show();
```

```
        }
```

```
        else if (s2.equals(prediction.name)) {
```

```
            setContentView(R.layout.main2);
```

```
            Toast.makeText(this, "Poison damage:" + dmg, Toast.LENGTH_SHORT).show();
```

```
        }
```

```
        else if (s3.equals(prediction.name)) {
```

```
            setContentView(R.layout.main3);
```

```
            Toast.makeText(this, "Ice damage:" + dmg, Toast.LENGTH_SHORT).show();
```

```
        }
```

```
        else if (s4.equals(prediction.name)) {
```

```
            setContentView(R.layout.main4);
```

```
            Toast.makeText(this, "Thunder damage:" + dmg, Toast.LENGTH_SHORT).show();
```

```
        }
```

```
    }
```

```
    else {
```

```
        setContentView(R.layout.main);
```

```
        Toast.makeText(this, "fail", Toast.LENGTH_SHORT).show();
```

```
    }
```

```
}
```

實作遊戲



- ❧ 由於我們僅改變一張圖片，上述切換Layout的作法顯得比較多餘
- ❧ 實際上每個xml檔案的差別也只有`android:foreground`這一行而已
- ❧ 比較簡潔的做法就是用同一個layout更換foreground圖片即可(在程式中動態的改變GestureOverlayView的前景圖片)

程式內添加變更前景圖片



✎ 先在xml內刪除 android:foreground="..."的敘述

✎ 程式中加入以下語法

(GestureOverlayView *view_name*).

setForeground(getResources().getDrawable(R.drawable.*drawable_name*));

XML内删除foreground設定



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <android.gesture.GestureOverlayView
        android:id="@+id/gestures"
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:foreground="@drawable/image1"
        android:foregroundGravity="center">

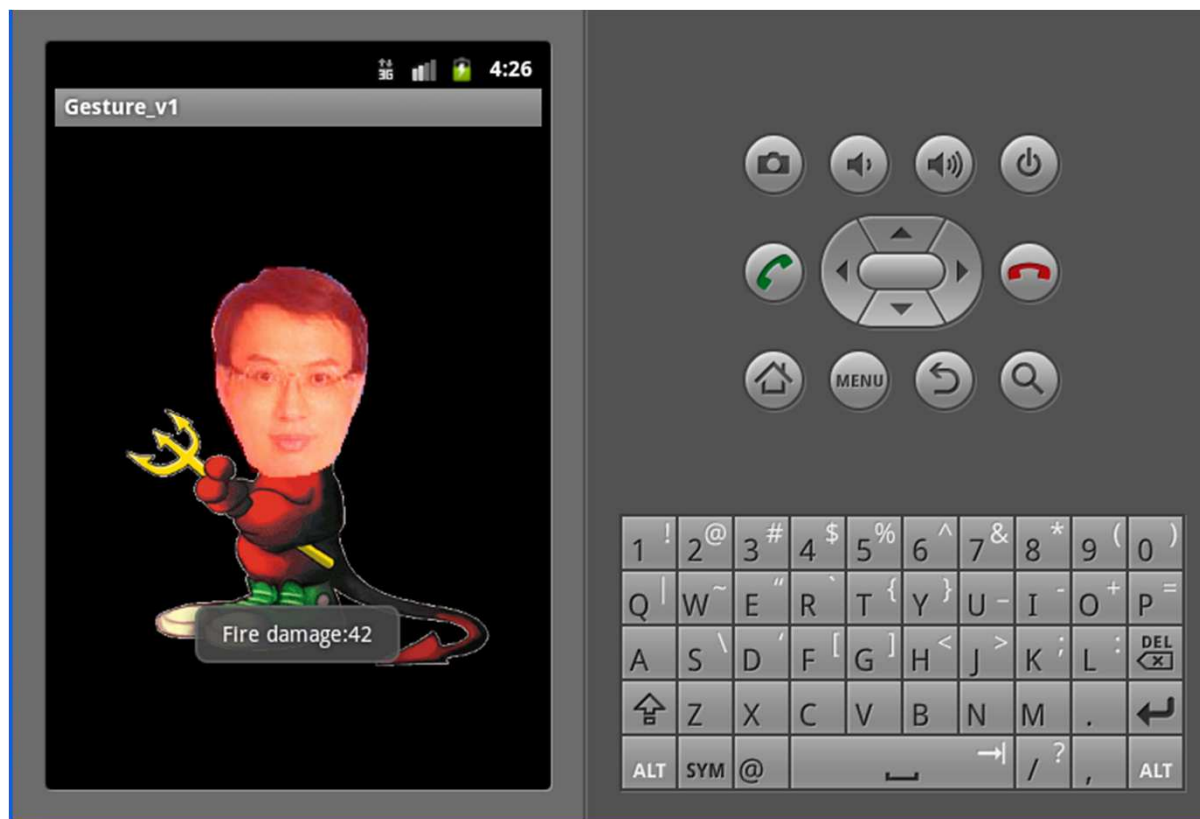
    </android.gesture.GestureOverlayView>
</LinearLayout>
```


程式內添加變更前景圖片



```
if (prediction.score > 1.5) {  
    // Show the spell  
    String s1 = "Fire", s2 = "Poison", s3 = "Ice", s4 = "Thunder";  
    if(s1.equals(prediction.name)){  
        gestures.setForeground(getResources().getDrawable(R.drawable.image_fire));  
        Toast.makeText(this, "Fire damage:"+dmg, Toast.LENGTH_SHORT).show();  
    }  
    else if(s2.equals(prediction.name)){  
        gestures.setForeground(getResources().getDrawable(R.drawable.image_poison));  
        Toast.makeText(this, "Poison damage:"+dmg, Toast.LENGTH_SHORT).show();  
    }  
    else if(s3.equals(prediction.name)){  
        gestures.setForeground(getResources().getDrawable(R.drawable.image_ice));  
        Toast.makeText(this, "Ice damage:"+dmg, Toast.LENGTH_SHORT).show();  
    }  
    else if(s4.equals(prediction.name)){  
        gestures.setForeground(getResources().getDrawable(R.drawable.image_thunder));  
        Toast.makeText(this, "Thunder damage:"+dmg, Toast.LENGTH_SHORT).show();  
    }  
}
```

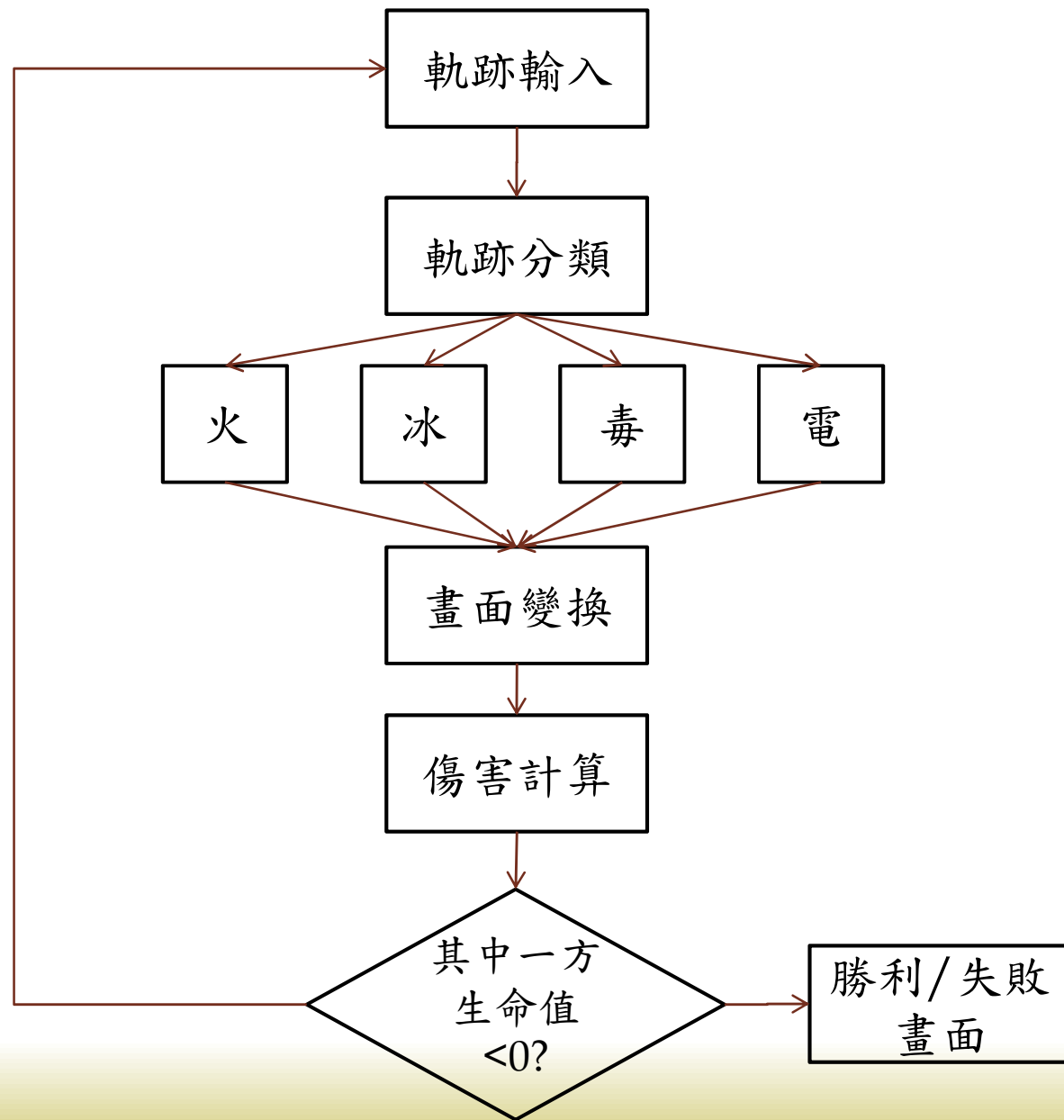

模擬器執行結果



實作遊戲



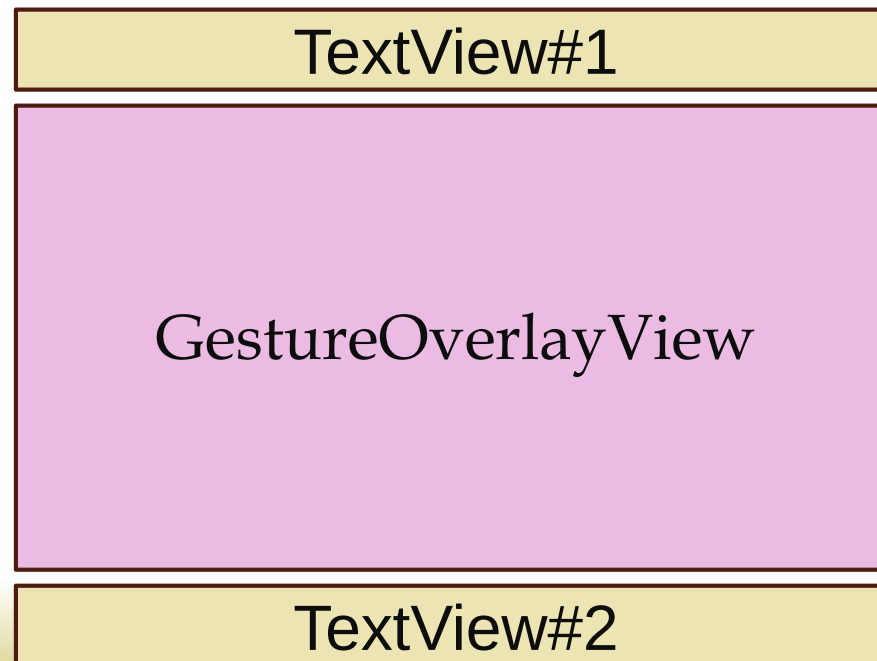
- ❧ 先前提到遊戲如果沒有勝負關係，就不能稱為遊戲了
- ❧ 因此我們加上玩家與怪物的生命值，與輪流的攻擊



設計 xml 檔案



- ☞ 我們把版面設計成三塊：怪物血量、怪物畫面與玩家血量。其中怪物畫面的部分則是採用 `GestureOverlayView` 的方式來記錄軌跡，血量則是用 `TextView` 顯示



設計 xml 檔案



☞ 這樣的編排方式，可簡單的直接用LinearLayout的格式完成

```
☞ <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView
    ...
    />
    <android.gesture.GestureOverlayView
    ...
    </android.gesture.GestureOverlayView>
    <TextView
    ...
    />
</LinearLayout>
```

設計 xml 檔案



```
<TextView  
    android:id="@+id/textview1"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center_horizontal"  
    android:textStyle="bold"  
>
```

設計 xml 檔案



```
<TextView  
    android:id="@+id/textview2"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center_horizontal"  
    android:textStyle="bold"  
>
```

執行結果



- 如果要畫面再美觀一點，可以加入一個背景圖片
`android:background="@drawable/xxx"`



加入動畫

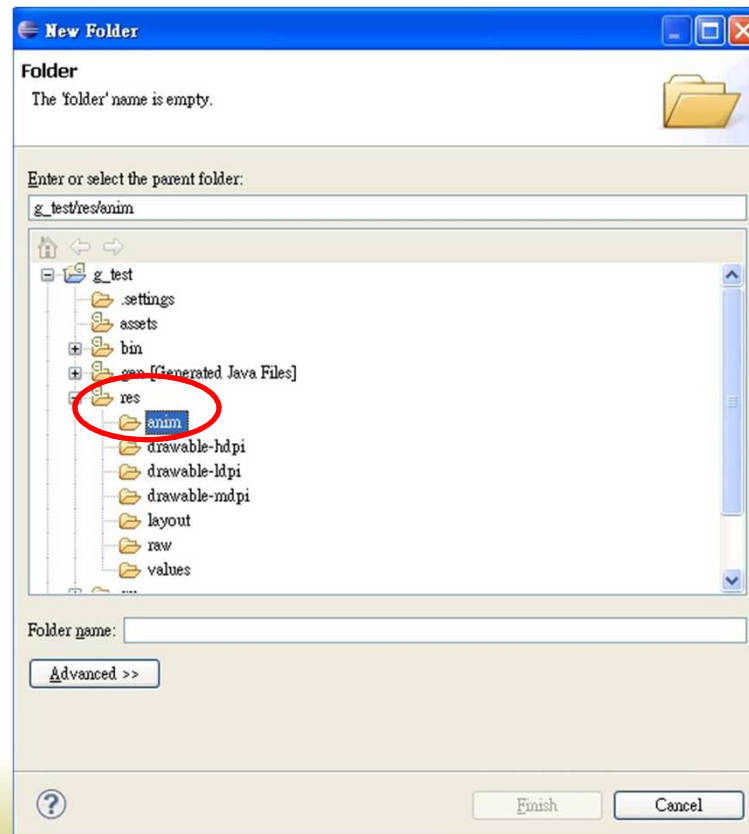


- ❧ 如果想要讓遊戲更生動一點，可以再加入動畫的部分
- ❧ Android中的動畫呈現有兩種：Tween animation與Frame animation，這裡利用的是Tween animation
- ❧ 要使用動畫需要import以下兩個類別
- ❧ `import android.view.animation.Animation;`
- ❧ `import android.view.animation.AnimationUtils;`

加入動畫



- 在res資料夾下建立一個anim的子資料夾，裡面放的是動畫的xml檔



加入動畫



- ☞ Tween animation有四種動畫：平移(translate)縮放(scale)旋轉(rotation)與透明(alpha)。這裡用平移與縮放作示範
- ☞ 我們定義：玩家發動攻擊後使用搖晃動畫(利用平移動畫完成)，怪物發動攻擊時使用縮放動畫

加入搖晃動畫



```
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:fromXDelta="0"
    android:interpolator="@anim/cycle_7"
    android:toXDelta="10"
/>
```

☞ 新增一個cycle_7.xml，內容為

```
<cycleInterpolator
xmlns:android="http://schemas.android.com/apk/res/android"
android:cycles="7" />
```

加入縮放動畫



```
<scale xmlns:android="http://schemas.android.com/apk/res/android"  
    android:duration="1250"  
    android:fromXScale="0.9"  
    android:fromYScale="0.9"  
    android:toXScale="1.2"  
    android:toYScale="1.2"  
    android:pivotX="50%"  
    android:pivotY="50%"  
/>
```

執行動畫



☞ 宣告兩個動畫變數

```
anim1 = AnimationUtils.loadAnimation(this,  
R.anim.anim1xmlfile);
```

```
anim2 = AnimationUtils.loadAnimation(this,  
R.anim.anim2xmlfile);
```

☞ 設定執行動畫的view id

```
findViewById(R.id.viewid).startAnimation(anim1);
```

```
findViewById(R.id.viewid).startAnimation(anim2);
```

執行動畫



- ❧ 如果需要播放連續動畫的話，可使用以下做法
 - ❧ 在xml中使用animation set(也就是把要做的animation依序寫好)
 - ❧ 但如果我們想在連續動畫的中間加入一些變化(例如：音效，換圖等)，就必須要使用第二種方法：在程式中override AnimationListener

執行連續的動畫



✧ 在class宣告時加入

`implements Animation.AnimationListener`

✧ override以下三個函式

```
public void onAnimationEnd(Animation animation) {
```

```
//動畫結束時要做的事情
```

```
}
```

```
public void onAnimationRepeat(Animation animation) {
```

```
}
```

```
public void onAnimationStart(Animation animation){
```

```
//動畫開始時要做的事情
```

```
}
```


執行連續的動畫



```
public void onAnimationEnd(Animation animation) {  
    if (animation==anim1){  
        gestures.setForeground  
            (getResources().getDrawable(R.drawable.drawables_id));  
        findViewById(R.id.view_id).startAnimation(anim2);  
    }  
}
```

如此當anim1執行完畢便會繼續執行anim2

播放音效



- ❧ 一個遊戲如果沒有音效的搭配，那遊戲性就會大打折扣了
- ❧ 適當的搭配音效可以讓遊戲更好玩

播放音效



- ❧ Android中使用media類別來播放音效
- ❧ 先把音效檔放在專案的raw資料夾下
- ❧ `import android.media.MediaPlayer;`
- ❧ 在要播放音效的地方輸入以下程式碼
`MediaPlayer mplayer=MediaPlayer.create(this, R.raw.media_file);`
`mplayer.start();`

播放音效



例如，要在怪物攻擊時播放攻擊音效，便可在怪物攻擊動畫播放開始時載入並播放音效

```
public void onAnimationStart(Animation animation) {  
    if(animation==anim1){  
        MediaPlayer mp;  
        mp = MediaPlayer.create(this, R.raw.attack);  
        mp.start();  
    }  
}
```

播放音效

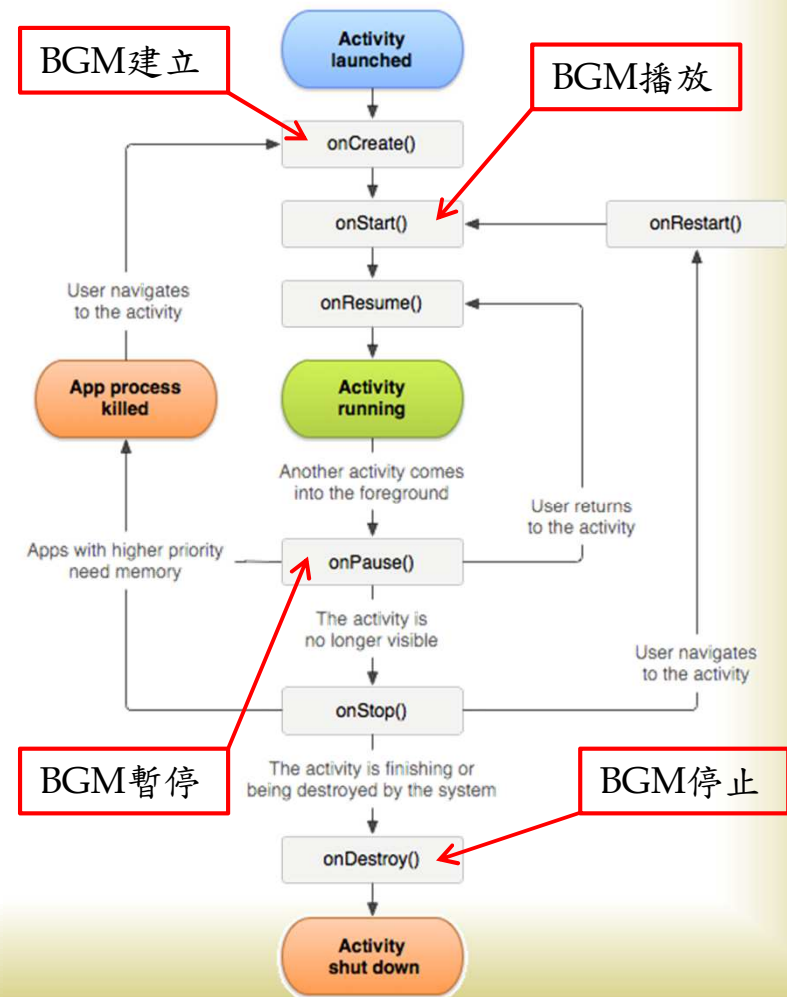


- ❧ 除了音效之外，遊戲通常都會有搭配BGM(背景音樂)
- ❧ BGM的做法和一般音效類似，不過需要將變數宣告為field，才能確保在整個activity中會維持執行
- ❧ 但使用Meidaplayer有一個要注意的地方，就是Medialplayer元件在android中是屬於Service
- ❧ 這樣會造成一個情況，就是當目前的activity被移至背景時(例如按下Home或是有來電)，Mediaplayer是可以繼續執行的
- ❧ 但一般來說，我們會希望當目前的activity被移至背景時，BGM會一併暫停。

播放音效



因此，BGM的播放必須
跟著activity的生命週
期一起運作



播放音效



```
❧ private MediaPlayer mplayer;  
    onCreate(){  
        mplayer=MediaPlayer.create(this, R.raw.media_file);  
        mplayer.setLooping(true);  
    }  
    onStart(){  
        mplayer.start();  
    }  
    onPause(){  
        mplayer.pause();  
    }  
    onDestroy(){  
        mplayer.stop();  
        mplayer.release();  
    }
```